

## N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM  
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT  
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED  
IN THE INTEREST OF MAKING AVAILABLE AS MUCH  
INFORMATION AS POSSIBLE



National Aeronautics and  
Space Administration

"Made available under NASA sponsorship  
in the interest of early and wide dis-  
semination of Earth Resources Survey  
Program information and without liability  
for any use made thereof."

JSC-13821  
Revision A

Lyndon B. Johnson Space Center  
Houston, Texas 77058

JAN 1980  
80-10140  
NASA CR-  
160572

EARTH OBSERVATIONS DIVISION

SPACE AND LIFE SCIENCES DIRECTORATE

EARTH OBSERVATIONS DIVISION VERSION OF THE LABORATORY  
FOR APPLICATIONS OF REMOTE SENSING SYSTEM  
(EOD-LARSYS) USER GUIDE FOR THE  
IBM 370/148  
VOLUME IV - PROGRAM LISTINGS

Job Order 76-662

(E80-10140) EARTH OBSERVATIONS DIVISION  
VERSION OF THE LABORATORY FOR APPLICATIONS  
OF REMOTE SENSING SYSTEM (EOD-LARSYS) USER  
GUIDE FOR THE IBM 370/148. VOLUME 4:  
PROGRAM LISTINGS (Lockheed Electronics Co.) G3/43 N80-20727  
Unclas 00140

Prepared By

Lockheed Electronics Company, Inc.  
Systems and Services Division  
Houston, Texas

Contract NAS 9-15800

November 1979



LEC-12566  
Revision A

JSC-13821  
Revision A

EARTH OBSERVATIONS DIVISION VERSION OF THE LABORATORY  
FOR APPLICATIONS OF REMOTE SENSING SYSTEM  
(EOD-LARSYS) USER GUIDE FOR THE  
IBM 370/148  
VOLUME IV - PROGRAM LISTINGS

Job Order 76-662

PREPARED BY

M. L. Burnell  
Technical Publications Department


and


P. J. Aucoin  
Earth Observations Data Products Department

APPROVED BY

NASA

LEC

  
\_\_\_\_\_  
J. M. Sulester, Technical  
Monitor, Systems and  
Facilities Branch

  
\_\_\_\_\_  
J. I. Morrow, Supervisor  
Scientific Applications  
Software Section

Prepared By

Lockheed Electronics Company, Inc.

For

Earth Observations Division  
Space and Life Sciences Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

November 1979

LEC-12566  
Revision A

## CONTENTS

Section	Page
1. SCOPE . . . . .	1-1
2. APPLICABLE DOCUMENTS . . . . .	2-1
3. MONTOR . . . . .	3-1
4. MSCAN . . . . .	4-1
5. COMMON BLOCKS AND BLOCK DATA . . . . .	5-1
6. HIST PROCESSOR . . . . .	6-1
7. GRAYMAP PROCESSOR . . . . .	7-1
8. STAT PROCESSOR . . . . .	8-1
9. ISOCLS PROCESSOR . . . . .	9-1
10. SELECT PROCESSOR . . . . .	10-1
11. CLASSIFY PROCESSOR . . . . .	11-1
12. DISPLAY PROCESSOR . . . . .	12-1
13. DATA-TR PROCESSOR . . . . .	13-1
14. TRSTAT PROCESSOR . . . . .	14-1
15. NDHIST PROCESSOR . . . . .	15-1
16. SCTRPL PROCESSOR . . . . .	16-1
17. DOTDATA PROCESSOR . . . . .	17-1
18. LABEL PROCESSOR . . . . .	18-1
19. UTILITY SUBPROGRAMS . . . . .	19-1
20. DAMRG PROCESSOR . . . . .	20-1
21. GTDDM PROCESSOR . . . . .	21-1
22. GTTCN PROCESSOR . . . . .	22-1
23. TESTSP PROCESSOR . . . . .	23-1



## 1. SCOPE

This document is one of a four-volume series entitled "Earth Observations Division Version of the Laboratory for Applications of Remote Sensing System (EOD-LARSYS) User Guide for the IBM 370/148" (section 2). Originally, the EOD-LARSYS software was written for execution on the Univac 1108/1110 computer at the Laboratory for Applications of Remote Sensing (LARS). The original version of this document covers the conversion of the EOD-LARSYS software for execution on the IBM 370/148, which was acquired subsequently by the LARS. The LARS recently replaced the IBM 370/148 with the IBM 3031 computer, which is thoroughly compatible with software as altered for execution on the IBM 370/148. Thus, no conversion of software is required for this system to be operable on the IBM 3031.

This volume IV contains a listing for each subprogram within the existing EOD-LARSYS processors and the utility subroutines. It is modeled after the As-Built Documentation (volume III), inasmuch as the listings appear in the same order as the subprograms are documented in volume III. Table 1-1 of volume III lists the EOD-LARSYS subprograms in alphabetical order, along with the processor to which each belongs. The processors, by section, are as follows:

<u>Section</u>	<u>Processor</u>
6	One-Dimensional Histogram (HIST)
7	GRAYMAP
8	Statistics (STAT)
9	Iterative Self-Organizing Clustering System (ISOCLS)
10	Feature Selection (SELECT)
11	Classification (CLASSIFY)
12	Performance Display (DISPLAY)

<u>Section</u>	<u>Processor</u>
13	Data-Transformation (DATA-TR)
14	Statistics Transformation (TRSTAT)
15	N-Dimensional Histogram (NDHIST)
16	Scatter Plot (SCTRPL)
17	Dot Data (DOTDATA)
18	Automatic Cluster Labeling (LABEL)

Within each of the above sections, the processor driver routine is listed first, followed by the subprogram listings in alphabetical order (the same order as they are documented in revision A of volume III). Utility subprograms are listed in section 19. In addition, this document contains subprogram listings for the following new processors:

<u>Section</u>	<u>Processor</u>
20	Data Merge (DAMRG)
21	Ground Truth Data Tape Dump (GTDDM)
22	Ground Truth Tape Conversion (GTTCN)
23	Iterative Self-Organizing Clustering System Using Packed Pixel Storage (TESTSP)

The listing for the EOD-LARSYS monitor routine, MONTOR, is given in section 3, along with a listing for an optional monitor routine, MONPAC. Provisions have been made in MONTOR for the addition of the following processors to the system: CLASY, AMOEBA, Equi-Probable Blocks (EQUPRB), Multitemporal Bayes (MULBAY), and Principal Component Greenness (PCG). These, which will be a part of the EOD-LARSYS, will be documented separately.

~~1-2~~

The MONPAC routine was created for use with the TESTSP processor, which clusters pixel values and stores them in packed form on disk storage. It differs from MONTOR in that it stores pixels in packed form (one sample per byte) rather than in floating point (one sample value every four bytes), as is done by the ISOCLS processor. The MONPAC routine may be used with other processors.

The listing for MSCAN, MONTOR's supervisory routine, is given in section 4, and common block listings are given in section 5.

## 2. APPLICABLE DOCUMENTS

1. Stewart, J.; et al.: EOD-LARSYS User Guide for the IBM 370/148 - vol. I, System Overview. JSC-13821, LEC-12563, NASA/JSC (Houston), Aug. 1978.
2. Stewart, J.; et al.: EOD-LARSYS User Guide for the IBM 370/148 - vol. II, User's Reference Manual. JSC-13821, LEC-12564, NASA/JSC (Houston), Dec. 1978.
3. Burnell, M. L.; et al.: EOD-LARSYS User Guide for the IBM 370/148 - vol. III, As-Built Documentation. JSC-13821, LEC-12565, NASA/JSC (Houston), Mar. 1979. (Revision A to be published.)

### 3. MONTOR

#### FILE MONITOR:

```

//MONTOR
-----
CALL..  SYSTEM MONITOR (// EXEC LARSYSA )
PURPOSE.. MONITORS THE VARIOUS SYSTEM SUPERVISORS
ROUTINES MSCAN  CLSFY  DSPLAY  STAT
          SELECT HIST  GRAYMP  DATATR
          ISOCLS TRSTAT NDHIST  SCTRPL
          DOTDATA LABEL  EQUPRB  MULRAY
          GTTCN  DAMRG
          AMOEBA CLASY TESTSP GTDDM PCG EXIT
RETURNS.. NONE
-----

IMPLICIT INTEGER(A-H,O-Z)
COMMON ARRAY
DIMENSION ARRAY(10600)

*ARRAY* IS A BLOCK OF STORAGE PASSED TO EACH PROCESSOR FOR THE
VARIABLE DIMENSIONING OF OTHER ARRAYS. THE ARRAY IS NEVER USED
TO PASS INFORMATION FROM ONE PROCESSOR TO ANOTHER.

DATA TOP/10600/
INCLUDE COMRKG.LIST

INCLUDE COMNT6.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWD, PAGESZ,DATFIL,STAFIL,ASAV,ASAVL
* NHSTUN,NHSTFL,SCTRUN,MAPFIL
* DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PPTUNT,WANDIO

GLOBAL COMMON IS USED IN EVERY PROCESSOR. IT IS ALWAYS IN COPE.
ALL PARAMETERS ARE INITIALIZED IN THE MONITOR.ROUTINE OR BLKCOM
EXCEPT AS NOTED BELOW
DEFINITIONS
HEAD - STANDARD HEADING PRINTED ON MOST OUTPUT PAGES.
MAPTAP - FORTRAN UNIT NUMBER ON WHICH THE MAPTAP FILE IS
WRITTEN (=2)
DATAPE - UNIT NO. FOR THE IMAGE DATA TAPE (=3)
SAVTAP - UNIT NO. ON WHICH THE STATISTICS FILE IS WRITTEN (=1)
BMFILE - UNIT NO. ON WHICH THE B-MATRIX FILE IS WRITTEN (=10)
BMKEY - TRIGGER INDICATING THAT THE B-MATRIX FILE HAS BEEN
WRITTEN. CAN BE SET IN SELECT CLASSIFY OR DATA-TR.
HISFIL - UNIT NO. ON WHICH THE HISTOGRAM FILE IS WRITTEN (=13)
HISKEY - TRIGGER INDICATING THE HISTOGRAM FILE HAS BEEN
WRITTEN. SET IN HIST PROCESSOR.
TRFORM - UNIT NO. ON WHICH THE TRANSFORMED IMAGE IS WRITTEN BY
THE DATA-TRANSFORMATION PROCESSOR. (=14)
ERIPTP - UNIT NO. ON WHICH THE ISOCLS PROCESSOR WRITES
CLUSTER STATISTICS FOR THE ERIPS SYSTEM. (=15)
ERPKEY - TRIGGER INDICATING THAT THE ERIPS INTERFACE TAPE
HAS BEEN WRITTEN.
MAPUNT - UNIT NO. ON WHICH THE ISOCLS OR DISPLAY PROCESSOR
WRITES THE CLUSTERED OR CLASSIFIED DATA
TO BE DISPLAYED ON THE PMIS DAS
NOFILE - NO. OF FILES WRITTEN ON UNIT 16 (MAP OUTPUT TAPE)
BY DISPLAY AND/OR ISOCLS
SET EITHER IN ISOCLS OR DISPLAY.
DRUMAD - BEGINNING ADDRESS FOR THE RANDOM ACCESS HIGH SPEED
REAL TIME
DRUM FILE. THIS FILE IS USED AS A SCRATCH FILE IN
SEVERAL PROCESSORS. REFERENCES TO SYSTEM ROUTINES
'HEAD' AND 'WRITE' ACCESS THIS FILE.
DRMWD - NO. OF WORDS AVAILABLE ON THE RANDOM ACCESS FILE.
PAGESZ - NO. OF LINES AVAILABLE FOR PRINTING ON A PAGE.
DATFIL - NO. OF E-O-F'S TO BE READ OVER BY TAPEFD ROUTINE IN
ORDER TO POSITION THE DATA TAPE TO DESIRED FILE
STAFIL - NO. OF E-O-F'S TO SKIP OVER TO POSITION STAT FILE

```

FILE MONTOR

C*	ASAV - UNIT NO. ON WHICH TRSTAT WRITES THE TRANSFORMED	MON00770
C*	STATS	MON00780
C*	ASAVFL - NO. OF E-O-F'S TO SKIP OVER TO POSITION TRANSFORMED	MON00790
C*	STATS	MON00800
C*	DOTUNT - UNIT NO. ON WHICH DOT DATA FILE (DOTFIL) IS WRITTEN	MON00810
C*	DOTFIL - NO. OF E-O-F'S TO SKIP OVER TO POSITION DOTFIL FILE	MON00820
C*	NCHPAS - NO. OF CHANNELS PER PASS	MON00830
C*	TRNSFL - NO. OF E-O-F'S TO SKIP OVER FOR TRFORM FILE	MON00840
C*	HMTXFL - NO. OF E-O-F'S TO SKIP OVER FOR HMFIL FILE	MON00850
C*	HISTFL - NO. OF E-O-F'S TO SKIP OVER FOR HISFIL FILE	MON00860
C*	PUNCH - UNIT NO. FOR CARD PUNCH FILE	MON00870
C*	CRDUNT - UNIT NO. FOR CARD READER	MON00880
C*	RANDIO - SCRATCH UNIT FOR RREAD AND RWRITE ROUTINES	MON00890
C	SEND	MON00900
C	DEBUG=-1	MON00910
C	SYSTEM ROUTINE RINIT ASSIGNS THE RANDOM ACCESS DRUM FILE.	MON00920
C	-DRUMAD-- IS THE ADDRESS TO BEGIN WRITING	MON00930
C	-DRUMDS- IS THE NO. OF WORDS AVAILABLE ON THE DRUM FILE.	MON00940
C	THE FOLLOWING PROCESSORS USE THE RANDOM ACCESS DRUM FILE FOR SCRATCH	MON00950
C	-ISOCLS-	MON00960
C	-DISPLY-	MON00970
C	-SELECT-	MON00980
C	-GRAYMP-	MON00990
C	-SIGEXT-	MON01000
C	DEFINE FILE 22(2100,200,U,1D)	MON01010
C	DRUMAD=1	MON01020
C	DRUMDS=420000	MON01030
C	WRITE(22,1)DRUMAD	MON01040
C	10 CONTINUE	MON01050
C	TIME = 0.	MON01060
C	CALL CLOCK(0)	MON01070
C	CALL MSCAN(JGO,DRUG)	MON01080
C	GO TO (20,40,60,80,100,120,140,160,175,180,200,220,240,260,	MON01090
C	* 280,290,300,310,320,330,340,350,360,370),JGO	MON01100
C	20 CONTINUE	MON01110
C	CALL STAT(ARRAY, TOP)	MON01120
C	CALL CLOCK ( 1, 'SSTA' )	MON01130
C	GO TO 10	MON01140
C	30	MON01150
C	40 CONTINUE	MON01160
C	CALL CLSFY(ARRAY, TOP)	MON01170
C	CALL CLOCK ( 1, 'SCLA' )	MON01180
C	GO TO 10	MON01190
C	50	MON01200
C	60 CONTINUE	MON01210
C	CALL DSPLAY(ARRAY, TOP)	MON01220
C	CALL CLOCK ( 1, 'SDIS' )	MON01230
C	GO TO 10	MON01240
C	70	MON01250
C	80 CONTINUE	MON01260
C	CALL SELECT(ARRAY, TOP)	MON01270
C	CALL CLOCK ( 1, 'SSEL' )	MON01280
C	GO TO 10	MON01290
C	90	MON01300
C	100 CONTINUE	MON01310
C	CALL HIST(ARRAY, TOP)	MON01320
C	CALL CLOCK ( 1, 'SHIS' )	MON01330
C	GO TO 10	MON01340
C	GO HERE FOR ISOCLS	MON01350
C	120 CONTINUE	MON01360
C	CALL ISOCLS(ARRAY, TOP)	MON01370
C	CALL CLOCK ( 1, 'SISO' )	MON01380
C	GO TO 10	MON01390
C	130	MON01400
C	C	MON01410
C		MON01420
C		MON01430
C		MON01440
C		MON01450
C		MON01460
C		MON01470
C		MON01480
C		MON01490
C		MON01500
C		MON01510
C		MON01520

FILE MONTOR

C	GO HERE FOR GRAYMAP	MONO	530
140	CONTINUE	MONO	540
	CALL GRAYMP (ARRAY.TOP)	MONO	550
150	CALL CLOCK ( 1. 'SGRA' )	MONO	560
	GO TO 10	MONO	570
C	GO HERE FOR DATA-TRANSFORMATION	MONO	580
		MONO	590
160	CONTINUE	MONO	600
	CALL DATATR (ARRAY.TOP)	MONO	610
170	CALL CLOCK ( 1. 'SDAT' )	MONO	620
	GO TO 10	MONO	630
C	GO HERE FOR SIGEXT MODULE	MONO	640
		MONO	650
175	CONTINUE	MONO	660
	*** SIGEXT	MONO	670
176	CALL CLOCK ( 1. 'SSIG' )	MONO	680
	GO TO 10	MONO	690
C	GO HERE FOR TRSTAT	MONO	700
		MONO	710
180	CONTINUE	MONO	720
	CALL TRSTAT (ARRAY.TOP)	MONO	730
190	CALL CLOCK ( 1. 'STRS' )	MONO	740
	GO TO 10	MONO	750
C	GO HERE FOR NDHIST	MONO	760
		MONO	770
200	CONTINUE	MONO	780
	CALL NDHIST (ARRAY.TOP)	MONO	790
210	CALL CLOCK ( 1. 'SNDH' )	MONO	800
	GO TO 10	MONO	810
C	GO HERE FOR SCTRPL	MONO	820
		MONO	830
220	CONTINUE	MONO	840
	CALL SCTRPL (ARRAY.TOP)	MONO	850
230	CALL CLOCK ( 1. 'SCT' )	MONO	860
	GO TO 10	MONO	870
C	GO HERE FOR DOTDATA	MONO	880
		MONO	890
240	CONTINUE	MONO	900
	CALL DOTDAT (ARRAY.TOP)	MONO	910
250	CALL CLOCK ( 1. 'SDOT' )	MONO	920
	GO TO 10	MONO	930
C	GO HERE FOR LABEL	MONO	940
		MONO	950
260	CONTINUE	MONO	960
	CALL LABEL (ARRAY.TOP)	MONO	970
270	CALL CLOCK ( 1. 'SLAB' )	MONO	980
	GO TO 10	MONO	990
C	GO HERE FOR EQUI-PROBABLE BLOCKS CLASSIFIER	MONO	2000
		MONO	2010
280	CONTINUE	MONO	2020
	CALL EQUIPR (ARRAY.TOP)	MONO	2030
285	CALL CLOCK ( 1. 'SEQU' )	MONO	2040
	GO TO 10	MONO	2050
C	GO HERE FOR MULTI-TEMPORAL CLASSIFIER	MONO	2060
		MONO	2070
290	CONTINUE	MONO	2080
	CALL MULHAY (ARRAY.TOP)	MONO	2090
295	CALL CLOCK ( 1. 'SMUL' )	MONO	2100
	GO TO 10	MONO	2110
C	GO HERE FOR GROUND TRUTH TO MAPFIL	MONO	2120
		MONO	2130
300	CONTINUE	MONO	2140
	CALL GTTCN (ARRAY.TOP)	MONO	2150
305	CALL CLOCK ( 1. 'SGTT' )	MONO	2160
	GO TO 10	MONO	2170
		MONO	2180
		MONO	2190
		MONO	2200
		MONO	2210
		MONO	2220
		MONO	2230
		MONO	2240
		MONO	2250
		MONO	2260
		MONO	2270
		MONO	2280

FILE MONTOR

```

C
C 310 GO HERE FOTR IMAGE DATA MENGE
      CONTINUE
315 CALL DAMRG (ARRAY, TOP)
      CALL CLOCK ( 1, 'SDAM' )
      GO TO 10
C
C      GO HERE FOR AMOEBE
C
C 320 CONTINUE
      CALL AMOERA (ARRAY, TOP)
325 CALL CLOCK ( 1, 'SAMO' )
      GO TO 10
C
C      GO HERE FOR CLASY
C
C 330 CONTINUE
      CALL CLASY (ARRAY, TOP)
335 CALL CLOCK ( 1, 'SCLS' )
      GO TO 10
C
C      GO HERE FOR TESTSP
C
C 340 CONTINUE
      CALL TESTSP (ARRAY, TOP)
345 CALL CLOCK ( 1, 'STES' )
      GO TO 10
C
C      GO HERE FOR GROUND TRUTH DOT UNLOAD
C
C 350 CONTINUE
      CALL GTDDM (ARRAY, TOP)
355 CALL CLOCK ( 1, 'SGTD' )
      GO TO 10
C
C      GO HERE FOR PCG
C
C 360 CONTINUE
      CALL PGSTAT (ARRAY, TOP)
      CALL CLOCK ( 1, 'SPCG' )
      GO TO 10
C
C      GO HERE TO EXIT
C
C 370 IF (NOFILE .GT. 0) REWIND MAPUNT
      END

```

```

MON02290
MON02300
MON02310
MON02320
MON02330
MON02340
MON02350
MON02360
MON02370
MON02380
MON02390
MON02400
MON02410
MON02420
MON02430
MON02440
MON02450
MON02460
MON02470
MON02480
MON02490
MON02500
MON02510
MON02520
MON02530
MON02540
MON02550
MON02560
MON02570
MON02580
MON02590
MON02600
MON02610
MON02620
MON02630
MON02640
MON02650
MON02660
MON02670
MON02680
MON02690
MON02700
MON02710
MON02720
MON02730
MON02740

```



FILE: CLOCK FORTRAN A

```
SUBROUTINE CLOCK(TIME)
INTEGER TIMER
TIME1 = TIMER(0)
TIME = (TIME1/60000.)-TIME
RETURN
END
```

```
CLO00010
CLO00020
CLO00030
CLO00040
CLO00050
CLO00060
```

FILE MONPAC

```

//MONITOR
-----
CALL..  SYSTEM MONITOR (// EXEC LARSYSA )
PURPOSE.. MONITORS THE VARIOUS SYSTEM SUPERVISORS
ROUTINES MSCAN  CLSFY  DSPLY  STAT
          SELECT HIST  GRAYMP  DATATR
          ISUCLS  TPSTAT  NUMIST  SCTRPL
          DOTDATA LABEL  EQUIPH  MULRAY
          GTTCN  DAMRG
          AMOEBA CLASY TESTSP GTDDM PCG EXIT
RETURNS.. NONE
-----

IMPLICIT INTEGER(A-H,O-Z)
COMMON ARRAY
DIMENSION ARRAY(10600)

*ARRAY* IS A BLOCK OF STORAGE PASSED TO EACH PROCESSOR FOR THE
VARIABLE DIMENSIONING OF OTHER ARRAYS. THE ARRAY IS NEVER USED
TO PASS INFORMATION FROM ONE PROCESSOR TO ANOTHER.

DATA TOP/10600/
INCLUDE COMMR6.LIST

INCLUDE COMMR6.LIST
COMMON/GLOBAL/HEAD(=3),MAPTAP,DATAPE,SAVTAP,EMFILE,EMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* HISTUN,HISTFL,SCRWON,MAPFIL
* DOTUNT,DOTFIL,NCHPAS,TRNSFL,EMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRTPNT,MANDIO

GLOBAL COMMON IS USED IN EVERY PROCESSOR. IT IS ALWAYS IN CORE.
ALL PARAMETERS ARE INITIALIZED IN THE MONITOR ROUTINE OR HLKCOM
EXCEPT AS NOTED BELOW
DEFINITIONS
HEAD - STANDARD HEADING PRINTED ON MOST OUTPUT PAGES.
MAPTAP - FORTRAN UNIT NUMBER ON WHICH THE MAPTAP FILE IS
WRITTEN (=2)
DATAPE - UNIT NO. FOR THE IMAGE DATA TAPE (=3)
SAVTAP - UNIT NO. ON WHICH THE STATISTICS FILE IS WRITTEN (=1)
EMFILE - UNIT NO. ON WHICH THE H-MATRIX FILE IS WRITTEN (=10)
EMKEY - TRIGGER INDICATING THAT THE H-MATRIX FILE HAS BEEN
WRITTEN. CAN BE SET IN SELECT CLASSIFY OR DATA-TR.
HISFIL - UNIT NO. ON WHICH THE HISTOGRAM FILE IS WRITTEN (=13)
HISKEY - TRIGGER INDICATING THE HISTOGRAM FILE HAS BEEN
WRITTEN. SET IN HIST PROCESSOR.
TRFORM - UNIT NO. ON WHICH THE TRANSFORMED IMAGE IS WRITTEN BY
THE DATA-TRANSFORMATION PROCESSOR. (=14)
ERIPTP - UNIT NO. ON WHICH THE ISUCLS PROCESSOR WRITES
CLUSTER STATISTICS FOR THE ERIPS SYSTEM. (=15)
ERPKEY - TRIGGER INDICATING THAT THE ERIPS INTERFACE TAPE
HAS BEEN WRITTEN.
MAPUNT - UNIT NO. ON WHICH THE ISUCLS OR DISPLAY PROCESSOR
WRITES THE CLUSTERED OR CLASSIFIED DATA
TO BE DISPLAYED ON THE DMIS DAS
NOFILE - NO. OF FILES WRITTEN ON UNIT 16 (MAP OUTPUT TAPE)
BY DISPLAY AND/OR ISUCLS
SET EITHER IN ISUCLS OR DISPLAY.
DRUMAD - BEGINNING ADDRESS FOR THE RANDOM ACCESS HIGH SPEED
REAL TIME
DRUMDS - DATA FILE. THIS FILE IS USED AS A SCRATCH FILE IN
SEVERAL PROCESSORS. REFERENCES TO SYSTEM ROUTINES
*READ* AND *WRITE* ACCESS THIS FILE.
DRUMDS - NO. OF WORDS AVAILABLE ON THE RANDOM ACCESS FILE.
PAGSIZ - NO. OF LINES AVAILABLE FOR PRINTING ON A PAGE.
DATFIL - NO. OF E-O-F'S TO BE READ OVER BY TAPERD ROUTINE IN
ORDER TO POSITION THE DATA TAPE TO DESIRED FILE
STAFIL - NO. OF E-O-F'S TO SKIP OVER TO POSITION STAT FILE*)

```

MONITOR PAGE 13

FILE MONPAC

C*	ASAV - UNIT NO. ON WHICH TRSTAT WRITES THE TRANSFORMED	MON00770
C*	STATS	MON00780
C*	ASAVFL - NO. OF E-O-F'S TO SKIP OVER TO POSITION TRANSFORMED	MON00790
C*	STATS	MON00800
C*	DOTUNT - UNIT NO. ON WHICH DOT DATA FILE (DOTFIL) IS WRITTEN	MON00810
C*	DOTFIL - NO. OF E-O-F'S TO SKIP OVER TO POSITION DOTFIL FILE	MON00820
C*	NCHPAS - NO. OF CHANNELS PER PASS	MON00830
C*	TRNSFL - NO. OF E-O-F'S TO SKIP OVER FOR TRFORM FILE	MON00840
C*	BMTXFL - NO. OF E-O-F'S TO SKIP OVER FOR BMFIL FILE	MON00850
C*	HISTFL - NO. OF E-O-F'S TO SKIP OVER FOR HISFIL FILE	MON00860
C*	PUNCH - UNIT NO. FOR CARD PUNCH FILE	MON00870
C*	CPDUNT - UNIT NO. FOR CARD READER	MON00880
C*	RADIO - SCRATCH UNIT FOR RREAD AND RWRITE ROUTINES	MON00890
C*END		MON00900
C*	DEBUG=-1	MON00910
C*	SYSTEM ROUTINE RINIT ASSIGNS THE RANDOM ACCESS DRUM FILE.	MON00920
C*	-DRUMAD-- IS THE ADDRESS TO BEGIN WRITING	MON00930
C*	-DRUMWS- IS THE NO. OF WORDS AVAILABLE ON THE DRUM FILE.	MON00940
C*	THE FOLLOWING PROCESSORS USE THE RANDOM ACCESS DRUM FILE FOR SCRAT	MON00950
C*	-ISOCLS-	MON00960
C*	-DISPLY-	MON00970
C*	-SELECT-	MON00980
C*		MON00990
C*	-GRAYNP-	MON01000
C*	-SIGEXT-	MON01010
C*		MON01020
C*	DEFINE FILE 22(640,200,U,10)	MON01030
C*	DRUMAD=1	MON01040
C*	DRUMWS=128000	MON01050
C*	WRITE(22,1)DRUMAD	MON01060
C*	10 CONTINUE	MON01070
C*	TIME = 0.	MON01080
C*	CALL CLOCK(TIME)	MON01090
C*	CALL MSCAN(J60,DRUG)	MON01100
C*	GO TO (20,40,60,80,100,120,140,160,175,180,200,220,240,260,	MON01110
C*	* 280,290,300,310,320,330,340,350,360,370),J60	MON01120
C*		MON01130
C*	20 CONTINUE	MON01140
C*	CALL STAT(ARRAY, TOP)	MON01150
C*	30 CALL CLOCK ( 1, 'SSTA' )	MON01160
C*	GO TO 10	MON01170
C*		MON01180
C*	40 CONTINUE	MON01190
C*	CALL CLSFY(ARRAY, TOP)	MON01200
C*	50 CALL CLOCK ( 1, 'SCLA' )	MON01210
C*	GO TO 10	MON01220
C*		MON01230
C*	60 CONTINUE	MON01240
C*	CALL DSPLY(ARRAY, TOP)	MON01250
C*	70 CALL CLOCK ( 1, 'SDIS' )	MON01260
C*	GO TO 10	MON01270
C*		MON01280
C*	80 CONTINUE	MON01290
C*	CALL SELECT(ARRAY, TOP)	MON01300
C*	90 CALL CLOCK ( 1, 'SSEL' )	MON01310
C*	GO TO 10	MON01320
C*		MON01330
C*	100 CONTINUE	MON01340
C*	CALL HIST(ARRAY, TOP)	MON01350
C*	110 CALL CLOCK ( 1, 'SHIS' )	MON01360
C*	GO TO 10	MON01370
C*		MON01380
C*	GO HERE FOR ISOCLS	MON01390
C*		MON01400
C*	120 CONTINUE	MON01410
C*	CALL ISOCLS(ARRAY, TOP)	MON01420
C*	130 CALL CLOCK ( 1, 'SISO' )	MON01430
C*	GO TO 10	MON01440
C*		MON01450
C*		MON01460
C*		MON01470
C*		MON01480
C*		MON01490
C*		MON01500
C*		MON01510
C*		MON01520

FILE MONPAC

C	GO HERE FOR GRAYMAP	MON01530
140	CONTINUE	MON01540
	CALL GRAYMP (ARRAY, TOP)	MON01550
150	CALL CLOCK ( 1, 'SGRA' )	MON01560
	GO TO 10	MON01570
C	GO HERE FOR DATA-TRANSFORMATION	MON01580
C		MON01590
160	CONTINUE	MON01600
	CALL DATATR (ARRAY, TOP)	MON01610
170	CALL CLOCK ( 1, 'SDAT' )	MON01620
	GO TO 10	MON01630
C	GO HERE FOR SIGEXT MODULE	MON01640
C		MON01650
175	CONTINUE	MON01660
	*** SIGEXT	MON01670
176	CALL CLOCK ( 1, 'SSIG' )	MON01680
	GO TO 10	MON01690
C	GO HERE FOR TRSTAT	MON01700
C		MON01710
180	CONTINUE	MON01720
	CALL TRSTAT (ARRAY, TOP)	MON01730
190	CALL CLOCK ( 1, 'STRS' )	MON01740
	GO TO 10	MON01750
C	GO HERE FOR NDHIST	MON01760
C		MON01770
200	CONTINUE	MON01780
	CALL NDHIST (ARRAY, TOP)	MON01790
210	CALL CLOCK ( 1, 'SNDH' )	MON01800
	GO TO 10	MON01810
C	GO HERE FOR SCTRPL	MON01820
C		MON01830
220	CONTINUE	MON01840
	CALL SCTRPL (ARRAY, TOP)	MON01850
230	CALL CLOCK ( 1, 'SCT' )	MON01860
	GO TO 10	MON01870
C	GO HERE FOR DOTDATA	MON01880
C		MON01890
240	CONTINUE	MON01900
	CALL DOTDAT (ARRAY, TOP)	MON01910
250	CALL CLOCK ( 1, 'SDOT' )	MON01920
	GO TO 10	MON01930
C	GO HERE FOR LABEL	MON01940
C		MON01950
260	CONTINUE	MON01960
	CALL LABEL (ARRAY, TOP)	MON01970
270	CALL CLOCK ( 1, 'SLAB' )	MON01980
	GO TO 10	MON01990
C	GO HERE FOR EQUI-PROBABLE BLOCKS CLASSIFIER	MON02000
C		MON02010
280	CONTINUE	MON02020
	CALL EQUIPRH (ARRAY, TOP)	MON02030
285	CALL CLOCK ( 1, 'SEQU' )	MON02040
	GO TO 10	MON02050
C	GO HERE FOR MULTI-TEMPORAL CLASSIFIER	MON02060
C		MON02070
290	CONTINUE	MON02080
	CALL MULRAY (ARRAY, TOP)	MON02090
295	CALL CLOCK ( 1, 'SMUL' )	MON02100
	GO TO 10	MON02110
C	GO HERE FOR GROUND TRUTH TO MAPFIL	MON02120
C		MON02130
300	CONTINUE	MON02140
	CALL GTTCN (ARRAY, TOP)	MON02150
305	CALL CLOCK ( 1, 'SGTT' )	MON02160
	GO TO 10	MON02170
		MON02180
		MON02190
		MON02200
		MON02210
		MON02220
		MON02230
		MON02240
		MON02250
		MON02260
		MON02270
		MON02280

FILE MONPAC

C			MON02290
C		GO HERE FOTR IMAGE DATA MERGE	MON02300
	310	CONTINUE	MON02310
		CALL DAMKG(ARRAY, TOP)	MON02320
	315	CALL CLOCK ( 1, 'SDAM' )	MON02330
		GO TO 10	MON02340
C		GO HERE FOR AMOEBA	MON02350
C			MON02360
C			MON02370
	320	CONTINUE	MON02380
		CALL AMOEBA(ARRAY, TOP)	MON02390
	325	CALL CLOCK ( 1, 'SAMO' )	MON02400
		GO TO 10	MON02410
C		GO HERE FOR CLASY	MON02420
C			MON02430
	330	CONTINUE	MON02440
		CALL CLASY(ARRAY, TOP)	MON02450
	335	CALL CLOCK ( 1, 'SCLS' )	MON02460
		GO TO 10	MON02470
C		GO HERE FOR TESTSP	MON02480
C			MON02490
	340	CONTINUE	MON02500
		CALL TESTSP(ARRAY, TOP)	MON02510
	345	CALL CLOCK ( 1, 'STES' )	MON02520
		GO TO 10	MON02530
C		GO HERE FOR GROUND TRUTH DOT UNLOAD	MON02540
C			MON02550
	350	CONTINUE	MON02560
		CALL GTDDM(ARRAY, TOP)	MON02570
	355	CALL CLOCK ( 1, 'SGTD' )	MON02580
		GO TO 10	MON02590
C		GO HERE FOR PCG	MON02600
C			MON02610
	360	CONTINUE	MON02620
		CALL PGSTAT(ARRAY, TOP)	MON02630
		CALL CLOCK(1, 'PCG')	MON02640
		GO TO 10	MON02650
C		GO HERE TO EXIT	MON02660
C			MON02670
	370	IF (NOFILE .GT. 0) REWIND MAPUNT	MON02680
		END	MON02690
			MON02700
			MON02710
			MON02720
			MON02730
			MON02740

# 4. MSCAN

## FILE MSCAN

```

SUBROUTINE MSCAN(MGO,DBUG)
IMPLICIT INTEGER (A-H,O-Z)
DIMENSION CODTAB(24),COMMENT(15),DATE(3)
1 ,      HE01(15), HE02(15),ACARD(20)
-----
CALL..      CALL MSCAN(MGO,DBUG)
CONTINUE
ARGS..      JGO      - PROCESSOR PTR
              DBUG    - DEBUG KEY  -1 FIRST ENTRY
                       0 - INCLUDE FLASH
                       1 - EXCLUDE FLASH

PURPOSE..   ANALYZES ALL MONITOR CONTROL CARDS

RETURNS..   JGO - 1 $STAT
                  2 $CLASS
                  3 $DISPLAY
                  4 $SELECT
                  5 $HIST
                  6 $ISOCLS
                  7 $RPMAY
                  8 $DATA-TRANSFORMATION
                  9 $SIG EXT
                 10 $TSTAT
                 11 $NHIST
                 12 $SCTRPL
                 13 $DOTDAT
                 14 $LABEL
                 15 $EQUIPJOB BLOCKS
                 16 $MULTI-TEMPORAL BAYES CLASSIFIER
                 17 $GROUND TRUTH TO CLUSTER MAP
                 18 $DATA MERGING
                 19 $AMOEBA

CONTINUE
                20 $CLASY
                21 $TESTSP
                22 $GTDDM
                23 $PCG
                24 $EXIT

-----
EQUIVALENCE (HE01(1), HE02(4)), (DATE(1), HE02(22)),
1           (HE02(1), HE02(30)), (COMMENT(1), HE02(48))
C INCLUDE COMMA,LIST
COMMON/GLOBAL/HEAD(63),MARTAP,DATAPE,SAVTAP,HMFILE,HMKEY,
*   HISFIL,HISKEY,IRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
*   DRUMAD,DRMUS,PAGSIZ,DAIFIL,STAFIL,ASAV,ASAVFL
*   NHSTUN,NHSTFI,SCTRPL,MAPFIL
*   DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
*   CRDUNT,PRUNT,RANDIO
C$END
COMMON /TAPEND/ IUNIT,IFRST,FSCAN,SAMENU,SAMINC,READY,NSCAN,
* LINC,IO(20),DSL,LHUF(30),JREC(30),IBYTE(30),NBUFS,FILENO,
* LINENO,LININC,ISAMP,NOCHAN,FORMAT
COMMON/ISTOR/ID(250)
DATA CODTAB /'STAT','CLASS','DIS','SEL','HIST','ISO','SGRA',
* 'DAT','SIG','IR','SNUM','SCT','DOT',
* 'LABR','EQUI','MIL','GTT','DAA',
* 'SAMO','ECLS','TES','GTDD','PCG','EXI'/
DATA FF /'FORM'/,LF00M /'1'/,LF00M /'2'/,LF00M /'3'/,LF00M /'4'/
DATA LHCD/'1'/,EHCD/'H'/,EHCD/'E'/
DIMENSION CARD(62)
-----
-----
-----
-----
INIZ
-----

```

MSC00010  
MSC00020  
MSC00030  
MSC00040  
MSC00050  
MSC00060  
MSC00070  
MSC00080  
MSC00090  
MSC00100  
MSC00110  
MSC00120  
MSC00130  
MSC00140  
MSC00150  
MSC00160  
MSC00170  
MSC00180  
MSC00190  
MSC00200  
MSC00210  
MSC00220  
MSC00230  
MSC00240  
MSC00250  
MSC00260  
MSC00270  
MSC00280  
MSC00290  
MSC00300  
MSC00310  
MSC00320  
MSC00330  
MSC00340  
MSC00350  
MSC00360  
MSC00370  
MSC00380  
MSC00390  
MSC00400  
MSC00410  
MSC00420  
MSC00430  
MSC00440  
MSC00450  
MSC00460  
MSC00470  
MSC00480  
MSC00490  
MSC00500  
MSC00510  
MSC00520  
MSC00530  
MSC00540  
MSC00550  
MSC00560  
MSC00570  
MSC00580  
MSC00590  
MSC00600  
MSC00610  
MSC00620  
MSC00630  
MSC00640  
MSC00650  
MSC00660  
MSC00670  
MSC00680  
MSC00690  
MSC00700  
MSC00710  
MSC00720  
MSC00730  
MSC00740  
MSC00750  
MSC00760

FILE MSCAN

C	CODEM = 24	MSC00770
	IF (DHUG.GE.0) GO TO 10	MSC00780
	DHUG = 0	MSC00790
	FORMAT = 1	MSC00800
	SET THE DATE FROM GTDATE	MSC00810
	10 WRITE (6,HEAD)	MSC00820
	READ AND DECODE THE MONITOR CARD	MSC00830
	-----	MSC00840
	SET UP RREAD BUFFER	MSC00850
	RUNIT=30	MSC00860
	CALL REREAD(RUNIT,80)	MSC00870
	NO. PUT THE CARD IN BUFFER	MSC00880
	20 READ (21,25) (ACARD(I),I=1,20)	MSC00890
	FORMAT(20A4)	MSC00900
	25 WRITE (RRUNIT,25) (ACARD(I),I=1,20)	MSC00910
	DEFIN RUNIT	MSC00920
	READ (RRUNIT,30) CODE1, CODE2	MSC00930
	30 FORMAT(2A4)	MSC00940
	DEFIN RUNIT	MSC00950
	WRITE (6,40) CODE1, CODE2	MSC00960
	40 FORMAT(1X,2A4/)	MSC00970
	DO 50 JGO=1, CODEM	MSC00980
	MGO = JGO	MSC00990
	IF (CODTAN(JGO) .EQ. CODE1) RETURN	MSC01000
	50 CONTINUE	MSC01010
	IF (CODE1.NE.FF) GO TO 55	MSC01020
	READ (30,52) CARD	MSC01030
	DEFIN 30	MSC01040
	52 FORMAT(10X,62A1)	MSC01050
	COL=0	MSC01060
	IF (R= NATCHR(CARD, COL))	MSC01070
	IF (LFOR.EQ.LFCOR) FORMAT=2	MSC01080
	IF (LFOR.EQ.LTREF) FORMAT=3	MSC01090
	IF (LFOR.EQ.LFCOR) FORMAT=4	MSC01100
	IF (LFOR.EQ.LFCO) FORMAT=2	MSC01110
	IF (LFOR.EQ.PHCO) FORMAT=3	MSC01120
	IF (LFOR.EQ.PHCO) FORMAT=4	MSC01130
	NOW=0	MSC01140
	53 COLL=COL	MSC01150
	JJ=NUMBER(CARD, COL, IDD, NOW)	MSC01160
	IF (JJ.GE.1) GO TO 20	MSC01170
	COL=COL+1	MSC01180
	IF (COL.LT.60) GO TO 53	MSC01190
	GO TO 20	MSC01200
	55 CONTINUE	MSC01210
	WRITE (6,60)	MSC01220
	GO TO 20	MSC01230
	60 FORMAT(/ / 5V, ***** MSCAN --- PROCESSOR CARD REQUIRED --- AHOM	MSC01240
	IVE CARD NOT A VALID PROCESSOR CARD --- CONTINUING SCAN ***** /)	MSC01250
		MSC01260
		MSC01270
		MSC01280
		MSC01290
		MSC01300
		MSC01310
		MSC01320
		MSC01330
		MSC01340
	END	MSC01350

42  
15

NO. PAGE  
QUALITY

## 5. COMMON BLOCKS AND BLOCK DATA

The common block listings are given in this section. For specific descriptions, definitions of the parameters, and processor and subprogram interfaces, see section 5 of volume III of this user guide. The common block listings are given here in alphabetical order, as they appear in volume III.

COMMON/ARRAY(10600)

COMMON/BESTKN/KPPPTS(60),IPRIOR,KBEST,NCPASS

COMMON/BMTRX/BMATRIX(450)

COMMON/CLASS/APRFLG,BMCOMB,BMFEAT,BMFLG,NOCAT,THIJI,IDATA1,NFILE,STATKY,  
CATNAM(60),CLSSYM(60),CON(60),DET(60),FLDESC,FLDINF(6),KCLSNA(60),  
NOCTCL(60),SUBCAT(60),NOCHAN,CHNVEC(30)

COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60),SUBCAT(60),  
CLSSUB(60),NOMAP,TOTVT3,NOSUB3,PCFDKY,TSTKEY,TRNKEY,THRSKY,STATKY,  
EMPTRS,THRSVA,PLTKEY,BMFLG,BMCOMB,BMFEAT,CDATE(2),FLDSV2,FIELD2,VERTX2,  
FLDSV3,FIELD3,VERTX3,PCTID3,THRES(60),SYMMTX(66),HIGH(60),CON(60),FLDKEY,  
NOFLD2,NOFLD3,NOFET2,FETVC2(30),NOSUB2,NOTRFD,TOTVT2,NOCLS2,KATNO(60),  
NOCAT,FILTER,MAPFMT,DESKEY,DESUNI,DESOTH,CROP,ACROP,AOTHER,ATOTAL,SITE(6),  
ANALYS(5),CAM(15),CRPKEY,KEPPTS(60),DOTKEY,DOTERR

COMMON/DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY,SIZE,LACIE

COMMON/DVNBLK/DFDK,CAYMIN,FII,CCAY,I1D,I1DMEN,ITT,ICNT,N

COMMON/FNTDUM/ITT,ICYCLE

COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,INCVEC(30),ICOUNT,SETWGT,  
EVALBF(100),FETVC4(30),NOFET4,VARSZ4,CORBAS,DTAB4,WGHS14,BESTVC(10),  
DIVSIZ,STATKY,ADRESD,ADRESP,ADRESF,ADRSH1,ADRSH2

COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,HISFIL,HISKEY,  
TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,  
ASAV,ASAVFL,NHSTUN,NHSTFI,SCTRUN,MAPFIL,DOTUNT,DOTFIL,NCHPAS,TRNSFL,  
BMTRFL,HISTFL,PCHUNT,CRDUNT,PRTUNT,RANDIO

COMMON/GRCBLK/MAXFET,NOFEAT,NOFET2,FETVEC(30),FETVC2(30),FLDINF(6),INFMT,  
FILESV,NOHIST,HISVEC(30),NOFLD,FLDPTS,XSIZ,XLOW,XHGH,YSIZ

COMMON/GTBK/NRDR,NPRT,PRTKEY,VLB(6),GTRDU,GTRDF,GTWRU,GTWRF,GTNOF

COMMON/HISTOR/HF



COMMON/IDSTOR/IDD(250)

COMMON/IDWORD/IDWORD(1000)

COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,AVAR2,COVAR2,CLSID2,  
SUBNO2,SUBDS2,FLDSV2,VERTEX2,FETVC2(30),SUBVC2(75),SUBPTR(75),CLSV2(60),  
KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),GRPCHK(61),GROUPS(124)

COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP

COMMON/LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60),SUBRAY(120),  
PTR(60),CATPTR(250),CATDOT(500),DOTVEC(250),COND,MIX,PROC,MAPKEY,  
DOTKEY,STATKY,SUNANG,T,NEARST,DIST,NOFEAT,FETVEC(30),OMAPUN,OMAPFI,  
OSAVTP,OSTAFI,NOSUN,ANGLE(8),SIZE,TOTDT2,FLDINF(6),CLSSYM(62),STADRS,  
MEANAD,TABADR,MAPADR,SUNCOR(30),ODOTUN,ODOTFI,MANSTA,MANDOT,DSPUNT,  
DSPFIL,DSPKEY,PRNSTS,PRNDOT,FLDNAM,VERTEX(22),NOVRT,NSUN,ANGLES(8),  
TOTDT3,FLDADR,VTXADR

COMMON/LISTMM/NPGA(3,2),NAMPGA(209,3,2),LINPGA(209,3,2),SAMPGA(209,3,2),  
DOTLAB(209,4,2),VPGA(3),IPGA

COMMON/MRGDAT/IMOPT,ISOPT,NUMFIL,IDATTP(6),IDATFL(6),NOFEAT,NFEAT(6),  
FETVEC(30,6),ISUN(8,6),SUNCOR(30),FLDINF(6,6),NOSAMP,NOLINE,NSS(6),  
NACROS,NLINES(6),LINPTR(7),LINES(600),FORMM

COMMON/NDIM/NCLRCH,CLRVEC(30),MAXVEC,MAPKEY,CLASS,SUBCLS,FIELD,MEANSW,  
NOVEC,FLDINF(6),SIZE,TOTMNS,CNTR1,CNTR2,ID1,ID2,COLOR1,COLOR2,BUFLEN,  
ID3,COLOR3,NODUMP,IData1,TOTVEC

COMMON/PASS/STOP,LNCAT,NMIN,KRN,STDMAX,DLMIN,SEP,MAP,SPTRIG,IRD,KPTS,NOPTS,  
PUNCH,ICHN,CHNTHS,ICHAIN(62),NWDS,IBEGIN,BEGIN1,BEGIN2,BEGIN3,CLSNAM,  
NOFLD,IPT,TOTWRD,TOTPTS,NCLASS,NOCLS,TOTSUB,TOTFLD,TOTVRT,NOCL,NVRT,  
NXTCLS,NOFEAT,MAXCLS,FETVEC(30),SYMPTX(62),VARSIZ,STATKY,ISOKEY,MAPFMT,  
MAPKEY,SEQUEN(20),PERCEN,SIMERP,IORDER,INUNIT,INFILE,INITM,PMIN,SUBVEC(62),  
NOSUB2,CHNVC(30),NOCHAN,ERCOMP,NOSEQ,MEANDO,MEANDU,SYMDO,SYMDU,ITRIGO,  
ITRIGU,DOFLAG,DUFLAG,DODU,STDOTS(60),NSDOTS,SUNCOR(30),LLNCAT,DVERT(250,2),  
DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2),MXFET1,MAXPOP

COMMON/PASSA/NOFET1,FTVEC1(30)

COMMON/PASSB/NOCLS,NOSUB,NOFEAT,NOFLD,TOTVRT,FETVEC(30),FLDSV1,CLSID1

COMMON/SCRACH/SCR1(2000),SCR2(10500)

COMMON/SCTTER/RSCALE,XYSCALE,CLRVEC(30),NCLRCH,CLRKEY,LOG,FREQ,XMAX,YMAX,  
XMIN,YMIN,BCKGND,XHI,XLO,YLO,XSIZ,YHI,YSIZ,NBINS,SYMPTX(32),BMATRIX(60),  
BVEC(30),NB:CHN,NOFEAT,SCALKY,MENADR,FLDADR,PNTADR,IDADR,NC,BMFEAT,  
BMCOMB,NOVEC,TOTMNS,SIZE,DRMID,DRMID1,DRMCLR,DRMCRI,DRMTNS,DRMTN1,  
DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD,DRMPTR,  
DRMPT1,FETVEC(16),DRMPLT,CSCALE,NOSUB

COMMON/STBASE/SUBSV1, SUBMN1, SUBVR1, SUBSD1, SUBCL1, SAVER1, HSTAL1, SPEC1, COVAR1,  
AVAR1, CLSID1, FLMEN1, FLVAR1, HFTAL1, FLDSV1

COMMON/STCBLK/MAXFET, MAXCLS, MAXFLD, NOFEAT, NOFET2, VARSIZ, NOSPEC, NOHIST,  
SPCBAS, IBLOCK(30), FETVEC(30), FETVC2(30), HISVEC(30), NOFLD, NOCLS,  
FLDINF(6), FLDPPTS, CLSPTS, XSIZ, XHGH, XLOW, YSIZ

COMMON/TAPERD/IUNIT, IFRST, FSCAN, SAMEND, SAMINC, READY, NSCAN, LINC, ID(200),  
DSL, LBUF(30), JREC(30), IBYTE(30), NBUFS, FILENO, LINEND, LININC, NSAMP,  
NOCHAN, FORMT

COMMON/TR/TRNS1(256), TRNS2(26), TRNS3(26), TY(11,19)

COMMON/TRBLCK/OUTFMT, NOFEAT, FLDINF(6), FETVEC(30)

COMMON/WRTAP/ICOUNT, FORMT, UNIT, VARBL(600), IREMD

## 6. HIST PROCESSOR

FILE: HIST

C	SURROUTINE HIST (ARRAY, TOP)	HIS000010
	IMPLICIT INTEGER (F-T)	HIS000020
C	DIMENSION ARRAY (1)	HIS000030
C	//HIST	HIS000040
C	-----	HIS000050
C	-----	HIS000060
C	CALL. CALL HIST (ARRAY, TOP)	HIS000070
C	REQUIRES. NO COMMON BLOCKS	HIS000080
C	ROUTINES SETUPS	HIS000090
C	HISTGM	HIS00100
C	PURPOSE. COORDINATES THE LOGICAL STEPS FOR HISTOGRAMMING	HIS00110
C	DATA	HIS00120
C	RETURNS. NONE	HIS00130
C	-----	HIS00140
C	-----	HIS00150
C	CALL SETUPS TO READ IN CONTROL CARDS	HIS00160
C	CALL SETUPS (FILHS, FLDL, TOTL, TOP)	HIS00170
C	CALL HISTGM (ARRAY (FILHS), ARRAY (FLDL), ARRAY (TOTL))	HIS00180
C	RETURN	HIS00190
C	END	HIS00200
		HIS00210
		HIS00220
		HIS00230
		HIS00240
		HIS00250
		HIS00260
		HIS00270
		HIS00280
		HIS00290
		HIS00300
		HIS00310



FILE: SETUPS

```

40 READ(30,500)MED1
   REWIND 30
   GO TO 10
C
C
C
C
50 READ(30,500)MED2
   REWIND 30
   GO TO 10
C
C
C
C
60 READ(30,500)COMENT
   REWIND 30
   GO TO 10
C
C
C
C
70 READ(30,510)DATE
   REWIND 30
   GO TO 10
C
C
C
C
90 J = NXTCHR(CARD,COL)
   IF (J.EQ.'BLANK') GO TO 10
   COL = COL - 1
   NOHIST = NUMBER(CARD,COL,NUMVC1,NOHIST)
   GO TO 10
C
C
C
C
97 COL = COL - 1
100 J = NXTCHR(CARD,COL)
   IF (J.EQ.'BLANK') GO TO 10
   IF (J.EQ.'YRCD') GO TO 130
   IF (J.EQ.'YRCD') GO TO 140
   GO TO 120
130 M = FIND12(CARD,COL,CHAR)
   IF (CHAR(M).NE.'EQUAL') GO TO 120
   M = NUMBER(CARD,COL,NUMVEC,0)
   YSZ = NUMVEC(1)
   GO TO 97
140 J = NXTCHR(CARD,COL)
   M = FIND12(CARD,COL,CHAR)
   IF (CHAR(M).NE.'EQUAL') GO TO 120
   M = NUMBER(CARD,COL,NUMVEC,0)
   IF (J.EQ.'HPCD') XHGH = NUMVEC(1)
   IF (J.EQ.'HPCD') GO TO 97
   IF (J.EQ.'LPCD') GO TO 120
   XLOW = NUMVEC(1)
   GO TO 97
C
C
C
C
DATAFILE POSITIONING CARD
1701 M=NXTCHR(CARD,COL)
   IF(M.EQ.'BLANK') GO TO 10
   IF(M.EQ.'U.CD') GO TO 1702
   IF(M.EQ.'HPCD') GO TO 1703
1705 WRITE(6,1705)
1704 FORMAT(' ERROR ON DATA FILE CARD *')
   GO TO 10
1702 J=FIND12(CARD,COL,EQUVEC)
   IF(J.EQ.-1) GO TO 1705
   M=NUMBER(CARD,COL,DATAPE,ZERO)
   COL=COL-1
   GO TO 1701
1703 J=FIND12(CARD,COL,EQUVEC)
   IF(J.EQ.-1) GO TO 1705
   FILNO=NUMBER(CARD,COL,DATAFIL,FILNO)
   DATAFIL=DATAFIL-1
   COL=COL-1
   GO TO 1701
C
C
C
C
*END* CARD
150 CONTINUE
   IF(NOHIST.EQ.0) GO TO 1
C

```

```

SET00800
SET00810
SET00820
SET00830
SET00840
SET00850
SET00860
SET00870
SET00880
SET00890
SET00900
SET00910
SET00920
SET00930
SET00940
SET00950
SET00960
SET00970
SET00980
SET00990
SET01000
SET01010
SET01020
SET01030
SET01040
SET01050
SET01060
SET01070
SET01080
SET01090
SET01100
SET01110
SET01120
SET01130
SET01140
SET01150
SET01160
SET01170
SET01180
SET01190
SET01200
SET01210
SET01220
SET01230
SET01240
SET01250
SET01260
SET01270
SET01280
SET01290
SET01300
SET01310
SET01320
SET01330
SET01340
SET01350
SET01360
SET01370
SET01380
SET01390
SET01400
SET01410
SET01420
SET01430
SET01440
SET01450
SET01460
SET01470
SET01480
SET01490
SET01500
SET01510
SET01520
SET01530
SET01540
SET01550
SET01560
SET01570
SET01580

```

FILE: SETUPS

```

C      IS DISPLAY A SURSET OF CHANNEL CARD
C
DO 140 J=1,NOHIST
DO 155 I=1,NOFEAT
IF (NUMVC(J) .EQ. FETVEC(I)) GO TO 160
155 CONTINUE
WRITE(6,560) NUMVC(J)
NUMVC(J) = 0
160 CONTINUE
CALL SORT(NOHIST,IPT,NUMVC,IPTVEC)
KA = IPT(I)
DO 165 I=1,NOHIST
HISVFC(I) = NUMVC(KA)
165 KA = IPTVEC(KA)
1 CONTINUE
C
C      CHECKING XHIGH AND XLOW
C
IF((XHIGH - XLOW) .GE. 100) GO TO 170
XLOW = XHIGH - 100
IF (XLOW .LT. 0) XHIGH = 100
IF (XLOW .LT. 0) XLOW = 0
WRITE(6,570) XHIGH,XLOW
170 CONTINUE
C
C      BASES FOR ARRAY
C
FILMS = 1
FLOTL = NOFEAT*256 + FILMS
TOTTL = NOHIST*XSIZ + FLOTL
TOTPTS = NOHIST*XSIZ + TOTTL - 1
IF(TOTPTS .LE. TOP) RETURN
C
C      RESET NO OF CHANNELS TO BE DISPLAYED AND RECALCULATE ADDRESS TOTTL
C
NOHST1 = NOHIST
NOHIST = (TOP - (FLOTL-1)) / (2*XSIZ)
TOTTL = NOHIST*XSIZ + FLOTL
NOHST = NOHIST + 1
C
C      ZERO OUT CHANNELS THAT ARE NOT TO BE PLOTTED
C
DO 200 I=NOHST,NOHST1
200 HISVFC(I) = 0
WRITE(6,580) NOHIST
650 FORMAT(/ ' TOO MANY CHANNELS ARE BEING HISTOGRAMMED AND PLOTTED. ' /
* ' NO. OF CHANNELS PLOTTED WAS RESET TO ',I5)
RETURN
480 FORMAT(A6,6X,62A1)
490 FORMAT(' INVALID CARD -- IGNORED' / T5,A4,6X,62A1)
500 FORMAT(10X,15A4)
510 FORMAT(10X,3A4)
570 FORMAT(' XHIGH - XLOW WAS LESS THAN 100. XHIGH WAS RESET TO ',I3,
* ' 2X.' OR XLOW WAS RESET TO ',I3)
550 FORMAT(5X,44,6X,62A1)
560 FORMAT(' CHANNEL ',I2,' IS NOT A SURSET OF THE CHANNELS GIVEN ON C
* CHANNEL CARD ')
630 FORMAT(/ ' INPUT SUMMARY ')
120 WRITE(6,640) CODE,CARD
640 FORMAT(/ ' 1X.A4,6X,62A1 / ' BAD SUPERVISOR CONTROL CARD' /)
GO TO 10
END

```

SET01500  
SET01600  
SET01610  
SET01620  
SET01630  
SET01640  
SET01650  
SET01660  
SET01670  
SET01680  
SET01690  
SET01700  
SET01710  
SET01720  
SET01730  
SET01740  
SET01750  
SET01760  
SET01770  
SET01780  
SET01790  
SET01800  
SET01810  
SET01820  
SET01830  
SET01840  
SET01850  
SET01860  
SET01870  
SET01880  
SET01890  
SET01900  
SET01910  
SET01920  
SET01930  
SET01940  
SET01950  
SET01960  
SET01970  
SET01980  
SET01990  
SET02000  
SET02010  
SET02020  
SET02030  
SET02040  
SET02050  
SET02060  
SET02070  
SET02080  
SET02090  
SET02100  
SET02110  
SET02120  
SET02130  
SET02140  
SET02150  
SET02160  
SET02170  
SET02180  
SET02190  
SET02200

FILE: SORT

```
SUBROUTINE SORT(IA,IPT,NUMVEC,IPTVEC)
.....C
CALL      -- SORT(IA,IPT,NUMVEC,IPTVEC)
          IA      -- NO. OF ELEMENTS TO BE SORTED IN ASCEN CI
                  DING ORDER
          IPT      -- CONTAINS REG. AND ENDING POINTER FOR
                  IPTVEC ARRAY
          NUMVEC   -- ARRAY CONTAINING ELEMENTS TO BE SORTED
          IPTVEC   -- ARRAY CONTAINING POINTERS FOR NUMVEC CI

          EXAMPLE:
                  IPT = 2
                  IPT = R
          SMALLEST NO. = NUMVEC(IPT(1))
          NEXT NO.    = NUMVEC(IPTVEC(IPT(1)))
          LAST NO.    = NUMVEC(IPTVEC(IPT(2)))
.....C

DIMENSION IPTVEC( 30 ),IPT(2),NUMVEC( 30 )

ICT      = 0
NOHST1 = IA
DO 60 I = 1,IA
  ICT = ICT + 1
  IF ( ICT .GT. 1 ) GO TO 10
  IPT(1) = 1
  IPT(2) = 1
  IF ( ICT .EQ. IPT(1) ) GO TO 60
10 KA = IPT(1)
  KB = IPT(2)
  IF (NUMVEC(I) .EQ. 0) NOHST1 = NOHST1 - 1
  IF (NUMVEC(I) .EQ. 0) GO TO 60
  IF (NUMVEC(I) .LE. NUMVEC(KA)) GO TO 30
  IF (NUMVEC(I) .GE. NUMVEC(KB)) GO TO 40
20 KB = KA
  KA = IPTVEC(KA)
  IF (NUMVEC(ICT) .LE. NUMVEC(KA)) GO TO 50
  GO TO 20
30 IPTVEC(ICT) = IPT(1)
  IPT(1) = ICT
  KB = IPT(2)
  GO TO 60
40 IPT(2) = ICT
  IPTVEC(KB) = ICT
  GO TO 60
50 IPTVEC(ICT) = KA
  IPTVEC(KB) = ICT
60 CONTINUE
  IA = NOHST1
  RETURN
END
```

## 7. GRAYMAP PROCESSOR

FILE: GRAYMP

	SUBROUTINE GRAYMP (ARRAY, TOP)	GRA00010
C	IMPLICIT INTEGER (A-Z)	GRA00020
C	DIMENSION ARRAY (TOP), RINLEV (30, 16)	GRA00030
C	DIMENSION SYMBOL (16, 2)	GRA00040
C	INCLUDE COMPK3.LIST	GRA00050
C	INCLUDE COMPK4.LIST	GRA00060
C	INCLUDE COMPK6.LIST	GRA00070
C	COMMON /GRCHLK/ MAXFET, NOFEAT, NOFET2, FETVEC (30),	GRA00080
C	FETVC2 (30), FLDINF (6), INFMT, FILESV, NOHIST,	GRA00090
C	MISVEC (30), NOFLD, FLDPTS	GRA00100
C	• XS17, XLOW, XHIGH, YSIZ	
C	DIMENSION MED1 (15), MED2 (15), DATE (3), COMENT (15)	COM00010
C	EQUIVALENCE (MED1 (1), HEAD (4)), (DATE (1), HEAD (22)),	COM00020
C	(MED2 (1), HEAD (30)), (COMENT (1), HEAD (48))	COM00030
C	2 COMMON /GLOBAL/ HEAD (63), MAPTAP, DATAPE, SAVTAP, RMFILE, RMKEY,	COM00010
C	MISFIL, MISKEY, TRFORM, ERIPTP, ERPKY, MAPUNT, NOFILE,	COM00020
C	DRUMAD, DRMWDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL	COM00030
C	• NHSTUN, NHSTFI, SCTRUN, MAPFIL	COM00040
C	• DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTFPL, HISTFL, PCHUNT,	COM00050
C	• CRDUNT, PRTUNT, RANDIO	COM00060
C	END	GRA00120
C	COMMON /HISTOR/HF	GRA00130
C	CALL SETUP6 (ARRAY, RINCNT, RINLEV, NUMBIN, SYMROL, SYMCNT, SYMDIM)	GRA00140
C	IF ((RINCNT.EQ.1).OR.(MISKEY.EQ.1)) GO TO 1	GRA00150
C	FILMS=1	GRA00160
C	FLDTL=4000	GRA00170
C	TOTTL=9000	GRA00180
C	HF=1	GRA00190
C	CALL HISTGM (ARRAY (FILMS), ARRAY (FLDTL), ARRAY (TOTTL))	GRA00200
C	CALL SETUP6 (ARRAY, RINCNT, RINLEV, NUMBIN, SYMROL, SYMCNT, SYMDIM)	GRA00210
C	1 CALL PICT (ARRAY, RINLEV, NUMBIN, SYMBOL, SYMCNT, SYMDIM)	GRA00220
C	HF=0	GRA00230
C	RETURN	GRA00240
C	END	GRA00260
C		GRA00270



FILE: HEADNG

```
SUBROUTINE HEADNG(TYPE,FETNUM,RINLEV,NUMBIN,FLDINP,
* SYMBOL,NSAMP,FET,SYMDIM,TCOL)
  IMPLICIT INTEGER(A-Z)
  CALL.. CALL HEADNG(TYPE,FETNUM)
  ARGS.. TYPE - REFERS TO COLUMN HEADING
          FETNUM - REFERS TO LOCATION OF FEATURE IN FETVC2 ARRAY
  REQUIRES. COMMONS /INFORM/ /INFORS/
              /GLOBAL/ /HELP/
  PURPOSE.. PRINTS OUT HEADING INFORMATION
  RETURNS NONE

  INCLUDE COMPK3.LIST
  INCLUDE COMPK4.LIST
  INCLUDE COMPK6.LIST
  COMMON /GRCHLK/MXFET,NOFEAT,NOFET2,FETVEC(30),
*        FETVC2(30),FLDINF(6),INFMT,FILESV,NOHIST,
*        HISVEC(30),NOFLD,          FLOPTS
* ,XSIZ,XLOW,XHGH,YSIZ
  DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
  EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
  2          (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
  COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
*        HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
*        DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
*        ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
*        ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
*        CRDUNT,PRUNT,RANDIO
CSEND
  DIMENSION SYMBOL(16,2),FET(1),FLDINP(7)
  DIMENSION TCOL(3,110),RINLEV(30,16)
  EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND),
*            (FLDINF(3),LININC),(FLDINF(4),SAMSTR),
*            (FLDINF(5),SAMEND),(FLDINF(6),SAMINC)
  IF (TYPE.EQ.2) GO TO 110
  IF (TCOL.EQ.1) GO TO 120
  J=FETNUM
103 FIRST=0
  WRITE(6,101) FIRST,(BINLEV(J,MA),MA=1,NUMBIN)
101 FORMAT(/12,16(3X,I3))
  DO 102 IZ=1,4
  WRITE(6,104)((SYMBOL(MA,1),SYMBOL(MA,1),SYMBOL(MA,1),SYMBOL(MA,1),
* SYMBOL(MA,1),SYMBOL(MA,1)),MA=1,NUMBIN)
104 FORMAT(2X,96A1)
  IF (SYMDIM.EQ.1) GO TO 102
  WRITE(6,105)((SYMBOL(MA,2),SYMBOL(MA,2),SYMBOL(MA,2),SYMBOL(MA,2),
* SYMBOL(MA,2),SYMBOL(MA,2)),MA=1,NUMBIN)
105 FORMAT(1H+.1X,96A1)
102 CONTINUE
120 CONTINUE
  CALL SETMRG(66,0,66)
C CALCULATE AND PRINT SAMPLE NUMBERS
  JG=0
  SS=FLDINP(4)
  SE=FLDINP(5)
  DO 106 I=SS,SE,SAMINC
  JG=JG+1
  TCOL(1,JG)=I/100
  TCOL(2,JG)=MOD(I,100)/10
  TCOL(3,JG)=MOD(I,10)
106 CONTINUE
110 IF (TYPE.EQ.2) WRITE(6,111)
  DO 107 I=1,3
107 WRITE(6,108) (ICOL(I,J),J=1,NSAMP)
108 FORMAT(10X,110I1)
  WRITE(6,111)
111 FORMAT(1H0)
  IF (TYPE.EQ.2) CALL SETMRG(66,4,62)
  RETURN
END
```

FILE: PICT

```

SUBROUTINE PICT(BUF,BINLEV,NUMBIN,SYMBOL,SYMCNT,SYMDIM)
C
C
C-----
CALL.. CALL PICT(IDATA)
ARGs.. IDATA - SCANNER DATA
ROUTINES HEADNG TAPHDR FLDINF LINERD LAREAD
PURPOSE.. PICTORIALY DISPLAYS FEATURES REQUESTED
RETURNS NONE
C-----
INCLUDE COMBK6.LIST
INCLUDE COMBK3.LIST
INCLUDE COMBK4.LIST
COMMON /GRCBLK/MAXFET,NOFFET,NOFFET2,FETVEC(30),
* FETVC2(30),FLDINF(6),INFMT,FILESV,NOHIST,
* HISVEC(30),NOFLD, FLOPTS
* ,XSIZ,XLOW,XHGH,YSIZ
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWDs,PAGSIZ,DATEFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCRUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
C$END
EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND),
* (FLDINF(3),LININC),(FLDINF(4),SAMSTR),
* (FLDINF(5),SAMEND),(FLDINF(6),SAMINC)
C
DIMENSION BINLEV(30,16)
DIMENSION LCHAR(2,256),LIN(110),IDATA(12000),BUF(110,20),JSTAT(20)
DIMENSION SYMBOL(16,2),FET(1),FLDINF(6)
DIMENSION VERTCS(2,11),FL(8)
C
DATA BLANK/' '
DATA OPAR/'( ','CPAR')','COMMA','/'
C
C READ HEADER RECORD ON DATA TAPE
CALL TAPHDR(DATAP,DATEFIL)
C READ FIELD DEFINITION CARDS
20 RUNNO=LAREAD(FLDNAM,VERTCS,FLDINF,NC)
IF(RUNNO.LE.0) GO TO 1
C CHECK TO SEE IF INFORMATION WILL FIT ON DRUM
NRUES=20
PTS=(FLDINF(5)-FLDINF(4))/FLDINF(6)+1
SPTS=PTS
LINES=(FLDINF(2)-FLDINF(1))/FLDINF(3)+1
NOFFET4=NOFFET2
27 NRUESZ=PTS*NOFFET4
TSAMP=LINES*NRUESZ
IF(TSAMP.LE.DRMWDs) GO TO 26
C REDUCE NO. OF CHANNELS BY ONE,CHECK TO SEE IF FIELD WILL FIT
NOFFET4=NOFFET4-1
GO TO 27
26 IF(NOFFET2.NE.NOFFET4) WRITE(6,28) NOFFET4
29 FORMAT(' THE NO. OF CHANNELS FOR THIS FIELD HAS BEEN REDUCED TO',
* 13,' SO ALL THE INFORMATION WILL FIT ON DRUM')/' MAKE ANOTHER RUNP
* TO GRAYMAP OTHER CHANNELS')
CALL FLDINT(FLDINF,FETVC2,NOFFET4)
ADRES=DRUMAD
DO 29 I=1,LINES
CALL LINE2D(IDATA,ENDTAP)
IF(ENDTAP.NE.0) GO TO 30
CALL RWRITE(ADRES,IDATA,NRUESZ,LSTAT)
31 IF(LSTAT.EQ.1) GO TO 31
ADRES=ADRES+NRUESZ
IF(ADRES.LE.DRUMAD+DRMWDs) GO TO 29
WRITE(6,33)

```

FILE: PICT

```
33 FORMAT(' FIELD TOO LARGE,TERMINATING')
CALL CMERR
29 CONTINUE
30 CONTINUE
FLDINP(1)=FLDINF(1)
FLDINP(2)=FLDINF(2)
FLDINP(3)=FLDINF(3)
FLDINP(6)=FLDINF(6)
NFET=1
C FOR EACH FEATURE
DO 4 J=1,NOFET4
PTS=SPTS
NADRES=DRUMAD+(J-1)*SPTS
FLDINP(4)=0
FLDINP(5)=0
FET(1)=FETVC2(J)
WRITE(6,HEAD)
MM=NC-1
300 FORMAT(T24,'SAMPLE LINE      NO. OF/
*2X,'CHANNEL  FIELDNAME      INC      INC      VERTICES      VERTICES(SAMP
*LE,LINE)
*7X,I3,7X,A4,7X,I4,2X,I4,7X,I2,7X,5(A1,I4,A1,I4,A1,2X)/
*5I,5(A1,I4,A1,I4,A1,2X))
WRITE(6,300) FET(1),FLDNAM      ,SAMINC,LININC,MM,
*((OPAR,VERTCS  (1,NM),COMMA,VERTCS(2,NM),CPAR),NM=1,MM)
TCOL=0
C SET UP VALUE-SYMBOL TABLE
DO 9 JD=1,SYMDIM
I2=BINLEV(J,1)+1
DO 8 I3=1,I2
8 LCHAR(JD,I3)=SYMBOL(1,JD)
DO 10 J1=2,NUMBIN
I1=BINLEV(J,J1-1)+2
I2=BINLEV(J,J1)+1
DO 10 I3=1,I2
10 LCHAR(JD,I3)=SYMBOL(J1,JD)
9 CONTINUE
FLDINP(4)=FLDINF(4)
AD=0
7 PPTS=PTS
ADRES=NADRES+AD
IF (PPTS.LE.110) GO TO 5
FLDINP(5)=FLDINP(4)+109*FLDINP(6)
GO TO 6
5 FLDINP(5)=FLDINF(5)
6 PPTS=PPTS-110
C INITIALIZE TAPE READING FOR THIS FIELD
CALL FLDINT(FLDINP,FET,NFET)
NSAMP=(FLDINP(5)-FLDINP(4))/FLDINP(6)+1
IF(NSAMP.GE.110) GO TO 101
IF(NSAMP.EQ.PPTS) GO TO 101
WRITE(6,100) FLDINP(5)
PTS=0
FLDINF(4)=FLDINP(4)
FLDINF(5)=FLDINP(5)
100 FORMAT(' YOU HAVE ASKED FOR TOO MANY SAMPLES, THE LAST SAMPLE IS',
* I5)
101 CONTINUE
LINES=(FLDINP(2)-FLDINP(1))/FLDINP(3)+1
TYPE=1
FETNUM=J
CALL HEADNG(TYPE,FETNUM,BINLEV,NUMBIN,FLDINP,SYMBOL,
* NSAMP,FET,SYMDIM,TCOL)
TCOL=1
LINCNT=0
C READ AND FILL 20 BUFFERS
DO 44 JA=1,NUBFS
CALL RREAD(ADRES,RUF(1,JA),NSAMP,JSTAT(JA))
ADRES=ADRES+NBUSZ
LINCNT=LINCNT+1
44 CONTINUE
LINE=LINSTR
IRUF=1
C FINISHED READING
36 IF(JSTAT(IRUF).EQ.1) GO TO 36
DO 200 AM=1,NSAMP
200 LIN(MM)=LANK
CALL FLDINT(VERTCS,NC,FL,LINE,NS,JJ)
DO 14 JU=1,SYMDIM
L=1
```

PIC00800  
PIC00810  
PIC00820  
PIC00830  
PIC00840  
PIC00850  
PIC00860  
PIC00870  
PIC00880  
PIC00890  
PIC00900  
PIC00910  
PIC00920  
PIC00930  
PIC00940  
PIC00950  
PIC00960  
PIC00970  
PIC00980  
PIC00990  
PIC01000  
PIC01010  
PIC01020  
PIC01030  
PIC01040  
PIC01050  
PIC01060  
PIC01070  
PIC01080  
PIC01090  
PIC01100  
PIC01110  
PIC01120  
PIC01130  
PIC01140  
PIC01150  
PIC01160  
PIC01170  
PIC01180  
PIC01190  
PIC01200  
PIC01210  
PIC01220  
PIC01230  
PIC01240  
PIC01250  
PIC01260  
PIC01270  
PIC01280  
PIC01290  
PIC01300  
PIC01310  
PIC01320  
PIC01330  
PIC01340  
PIC01350  
PIC01360  
PIC01370  
PIC01380  
PIC01390  
PIC01400  
PIC01410  
PIC01420  
PIC01430  
PIC01440  
PIC01450  
PIC01460  
PIC01470  
PIC01480  
PIC01490  
PIC01500  
PIC01510  
PIC01520  
PIC01530  
PIC01540  
PIC01550  
PIC01560  
PIC01570  
PIC01580

FILE: PICT

```
DO 201 IA=1,NSAMP
KA=(IA-1)*SAMINC+FLDINP(4)
DO 202 JK=L,JJ,2
IF(KA.LT.FL(JK)) GO TO 201
IF(KA.GT.FL(JK+1)) GO TO 203
IDT=RUF(IA,IRUF)+1
LIN(IA)=LCHAR(JD,IDT)
GO TO 201
203 L=L+2
IF(L.GT.JJ) GO TO 205
202 CONTINUE
201 CONTINUE
205 CONTINUE
IF(JD.NE.2) GO TO 21
WRITE(6,17) LINE.(LIN(JK),JK=1,NSAMP)
WRITE(4,102) LINE
102 FORMAT(1H+,T122,I5)
GO TO 14
17 FORMAT(1H+,I5,4X,110A1)
21 WRITE(6,15) LINE.(LIN(JK),JK=1,NSAMP)
15 FORMAT(16,4X,110A1)
14 CONTINUE
LINE=LINE+LININC
LINCNT=LINCNT+1
IF(LINCNT.GT.LINES) GO TO 37
CALL RREAD(ADRES,RUF(1,IBUF),NSAMP,JSTAT(IBUF))
ADRES=ADRES+NBUFSZ
37 IBUF=IBUF+1
IF(IRUF.GT.NBUFS) IHUF=1
IF(LINE.LE.LINEND) GO TO 36
TYPE=2
CALL HEADNG(TYPE,FETNUM,BINLEV,NUMBIN,FLDINP,SYMBOL,
* NSAMP,FET,SYNDIM,TCOL)
FLDINP(4)=FLDINP(6)+FLDINP(5)
AD=AD+110
IF(PTS.GT.0) GO TO 7
4 CONTINUE
GO TO 20
1 IF(RUNNO.EQ.0) RETURN
GO TO 20
END
```

PIC01590  
PIC01600  
PIC01610  
PIC01620  
PIC01630  
PIC01640  
PIC01650  
PIC01660  
PIC01670  
PIC01680  
PIC01690  
PIC01700  
PIC01710  
PIC01720  
PIC01730  
PIC01740  
PIC01750  
PIC01760  
PIC01770  
PIC01780  
PIC01790  
PIC01800  
PIC01810  
PIC01820  
PIC01830  
PIC01840  
PIC01850  
PIC01860  
PIC01870  
PIC01880  
PIC01890  
PIC01900  
PIC01910  
PIC01920  
PIC01930  
PIC01940  
PIC01950  
PIC01960  
PIC01970  
PIC01980  
PIC01990

ORIGINAL PAGE IS  
OF 11 PAGES

FILE: SETUP6

```
      SUBROUTINE SETUP6(FILHIS,BINCNT,BINLEV,NUMBIN,SYMBOL,SYMCNT,      SET00010
      * SYMDIM)                                                         SET00020
C      IMPLICIT INTEGER(A-Z)                                           SET00030
C      -----                                                         SET00040
C      CALL.. CALL SETUP6(FILHIS)                                       SET00050
C      ARGV.. FILHIS - HISTOGRAM DATA ARRAY                           SET00060
C      REQUIRES COMMONS /INFORM/ /INFORS/                               SET00070
C      /GLOBAL/ /HELP/                                                 SET00080
C      ROUTINES NXTCHR FIND12 NUMBER                                    SET00090
C      PURPOSE.. READS AND ANALYSES CONTROL CARDS FOR 'GRAYMAP' STEP   SET00100
C      RETURNS.. SUPERVISOR INFORMATION                                SET00110
C      -----                                                         SET00120
C      INCLUDE COMRK3.LIST                                             SET00130
C      INCLUDE COMRK4.LIST                                             SET00140
C      INCLUDE COMRK6.LIST                                             SET00150
C      COMMON /GRCLK/MAXFFT,NOFEAT,NOFET2,FETVEC(30),                  SET00160
C      * FETVC2(30),FLDINF(6),INFMT,FILESV,NOHIST,                      SET00170
C      * HISVEC(30),NOFLD, FLDPTS                                       SET00180
C      * XSIZ,XLOW,XHGH,YSIZ                                           SET00190
C      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)                 SET00200
C      EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),              SET00210
C      2 (HED2(1),HEAD(3)),(COMENT(1),HEAD(48))                        SET00220
C      COMMON/GLOBAL/HEAD(63),MAPTAP,DATEPE,SAVTAP,RMFILE,RMKEY,      SET00230
C      * HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,           SET00240
C      * DRUMAD,DRMADS,PAGSIZ,DATEFIL,STAFIL,ASAV,ASAVL              SET00250
C      * NHSTUN,NHSTFI,SCTRUN,MAPFIL                                    SET00260
C      * DOTUNT,DOTFIL,NCHPAS,TRNSFL,RMTRFL,HISTFL,PCHUNT,           SET00270
C      * CROUT,PRUNT,RANDIO                                             SET00280
C      -----                                                         SET00290
C      COMMON /HISTOR/HF                                               SET00300
C      EQUIVALENCE (FLDINF(1),LINTR),(FLDINF(2),LINEND),              SET00310
C      * (FLDINF(3),LININC),(FLDINF(4),SAMSTR),                       SET00320
C      * (FLDINF(5),SAMEND),(FLDINF(6),SAMINC)                         SET00330
C      DIMENSION FRVEC1(3),FRVEC2(3),NUMVEC(30)                       SET00340
C      DIMENSION SYMBOL(15,2),ACARD(20)                               SET00350
C      DIMENSION CINDEX(10),BINLEV(30,16),EQUVEC(2),                 SET00360
C      * SINVEC(3),CARD2(62),HGTPT(30),FILHIS(NOFEAT,256),COMMA(2)   SET00370
C      EQUIVALENCE (SINVEC(3),EQUAL)                                   SET00380
C      DATA CINDEX//CHAN//,BINL//,SYMB//,FORM//,HED1//,             SET00390
C      * HED2//,COMM//,DATE//,END//,DATA//                            SET00400
C      DATA EQUVEC/1//,//,//,//,//,//,//,//,//,//,//,//,//,//,   SET00410
C      DATA SINVEC/2//,//,//,//,//,//,//,//,//,//,//,//,//,//,   SET00420
C      * BLANK//,//,COMMA/1//,//,//,FRVEC1/2//,//,//,//,//,//,//,   SET00430
C      * FRVEC2/2//,//,//,//,//,//,//,//,//,//,//,//,//,//,//,   SET00440
C      DATA MRCD//,//,ORCD//,//,XRCD//,//,DLRCD//,//,DOTBCD//,//,   SET00450
C      1 FOLRCD//,//,MNSPCD//,//,SLHRCD//,//,STRACD//,//,RLKBCD//,//, SET00460
C      2 FRCD//,//,URCD//,//,//,//,//,//,//,//,//,//,//,//,//,//,   SET00470
C      IK=1                                                            SET00480
C      NOFET2=0                                                         SET00490
C      MAXFFT=30                                                         SET00500
C      IF (HF,F0,1) GO TO 80                                           SET00510
C      SYMDIM=0                                                         SET00520
C      NUMBIN=0                                                         SET00530
C      BINCNT=0                                                         SET00540
C      -----                                                         SET00550
C      SETUP REREAD BUFFER                                             SET00560
C      CALL REREAD(30,80)                                              SET00570
C      14 COL=0                                                         SET00580
C      PUT NEXT CARD IN BUFFER                                         SET00590
C      READ(21,200)(ACARD(I),I=1,20)                                  SET00600
C      200 FORMAT(20A4)                                                SET00610
C      WRITE(30,200)(ACARD(I),I=1,20)                                  SET00620
C      REWIND 30                                                       SET00630
C      -----                                                         SET00640
C      -----                                                         SET00650
C      -----                                                         SET00660
C      -----                                                         SET00670
C      -----                                                         SET00680
C      -----                                                         SET00690
C      -----                                                         SET00700
C      -----                                                         SET00710
C      -----                                                         SET00720
C      -----                                                         SET00730
C      -----                                                         SET00740
C      -----                                                         SET00750
C      -----                                                         SET00760
C      -----                                                         SET00770
C      -----                                                         SET00780
C      -----                                                         SET00790
```

FILE: SETUP6

```

C HED1 CARD
  8 READ(30,25) HED1
  REWIND 30
  GO TO 14
C HED2 CARD
  9 READ(30,25) HED2
  REWIND 30
  GO TO 14
  25 FORMAT(10X,15A4)
C COMMENT CARD
  10 READ(30,25) COMENT
  REWIND 30
  GO TO 14
C DATE CARD
  11 M=NXTCHR(CARD2,COL)
  IF(M.EQ.BLANK) GO TO 14
  READ(30,25) DATE
  REWIND 30
  GO TO 14
C
C DATAFILE POSITIONING CARD
C
  1701 M=NXTCHR(CARD2,COL)
  IF(M.EQ.BLANK) GO TO 14
  IF (M.EQ. HRCO) GO TO 1702
  IF (M.EQ. HPCD) GO TO 1703
  1705 WRITE(6,1704)
  1704 FORMAT(' ERROR ON DATA FILE CARD *')
  GO TO 14
  1702 J=FNDF12(CARD2,COL,EQUVEC)
  IF(J.EQ.-1) GO TO 1705
  M=NUMBER(CARD2,COL,DATAPE,ZERO)
  COL=COL-1
  GO TO 1701
  1703 J=FNDF12(CARD2,COL,EQUVEC)
  IF(J.EQ.-1) GO TO 1705
  FILNO=NUMBER(CARD2,COL,DATAFIL,FILNO)
  DATAFIL=DATAFIL-1
  COL=COL-1
  GO TO 1701
C *FND* CARD
  12 IF(MSYM.EQ.1) GO TO 26
C DEFAULT SYMBOLS
  SYMCNT=10
  SYMDIM=2
  SYMBOL(1,1) = MRCO
  SYMBOL(1,2) = DLHRCO
  SYMBOL(2,1) = XRCO
  SYMBOL(2,2) = EQLHRCO
  SYMBOL(3,1) = ORCO
  SYMBOL(3,2) = MNSRCD
  SYMBOL(4,1) = ORCO
  SYMBOL(4,2) = OHCO
  SYMBOL(5,1) = STRRCD
  SYMBOL(5,2) = MNSPCD
  SYMBOL(6,1) = EQLHCO
  SYMBOL(6,2) = EQLHCO
  SYMBOL(7,1) = DOTRCD
  SYMBOL(7,2) = DUTHCO
  SYMBOL(8,1) = MNSHCO
  SYMBOL(8,2) = MNSRCD
  SYMBOL(9,1) = SLHRCO
  SYMBOL(9,2) = SLHRCO
  SYMBOL(10,1) = BLKBCO
  SYMBOL(10,2) = BLKBCO
C CHECK TO SEE IF BINLEVELS INPUT
  26 IF(BINCNT.EQ.1) GO TO 27
  NUMHTN=SYMCNT
  IF(HISFIL.NE.13) STOP 2
C READ HISTOGRAM AND CALCULATE BINLEVELS
  IF(HISKEY.EQ.1) GO TO 80
  NOFEAT=NOFEAT2
  DO 81 IZ=1,NOFEAT2
  81 FETVEC(IZ)=FETVC2(IZ)
  RETURN
  80 READ(HISFIL) NOFEAT.(FETVEC(I),I=1,NOFEAT)
  READ(HISFIL) ((FILHIS(I,J),J=1,256),I=1,NOFEAT)
  REWIND HISFIL

```

SET01590  
 SET01600  
 SET01610  
 SET01620  
 SET01630  
 SET01640  
 SET01650  
 SET01660  
 SET01670  
 SET01680  
 SET01690  
 SET01700  
 SET01710  
 SET01720  
 SET01730  
 SET01740  
 SET01750  
 SET01760  
 SET01770  
 SET01780  
 SET01790  
 SET01800  
 SET01810  
 SET01820  
 SET01830  
 SET01840  
 SET01850  
 SET01860  
 SET01870  
 SET01880  
 SET01890  
 SET01900  
 SET01910  
 SET01920  
 SET01930  
 SET01940  
 SET01950  
 SET01960  
 SET01970  
 SET01980  
 SET01990  
 SET02000  
 SET02010  
 SET02020  
 SET02030  
 SET02040  
 SET02050  
 SET02060  
 SET02070  
 SET02080  
 SET02090  
 SET02100  
 SET02110  
 SET02120  
 SET02130  
 SET02140  
 SET02150  
 SET02160  
 SET02170  
 SET02180  
 SET02190  
 SET02200  
 SET02210  
 SET02220  
 SET02230  
 SET02240  
 SET02250  
 SET02260  
 SET02270  
 SET02280  
 SET02290  
 SET02300  
 SET02310  
 SET02320  
 SET02330  
 SET02340  
 SET02350  
 SET02360  
 SET02370

FILE: SETUP6

```
DO 44 I=1,NOFEAT
44 HGTPT(I)=0
DO 45 I=1,NOFEAT
DO 45 J=1,256
45 HGTPT(I)=HGTPT(I)+FILHIS(I,J)
C IF FEATURES CARD NOT INPUT,FETVC2 ARRAY= FETVEC ARRAY,NOFET2=NOFEAT
IF(NOFFT2.NE.0) GO TO 60
NOFET2=NOFEAT
DO 61 I=1,NOFEAT
61 FETVC2(I)=FETVEC(I)
GO TO 100
60 IK=1
KT=NOFFT2
102 DO 28 I=1,KT
DO 28 J=1,NOFEAT
C CHECK TO SEE IF FEATURES HISTOGRAMMED
IF(FETVC2(I).NE.FETVEC(J)) GO TO 29
GO TO 28
29 CONTINUE
WRITE(6,30) FETVC2(I)
30 FORMAT(1X,'THIS CHANNEL IS NOT HISTOGRAMMED',I3//)
NOFET2=NOFET2-1
IF(I.GT.KT) GO TO 100
IK=I+1
DO 101 II=IK,KT
101 FETVC2(II-1)=FETVC2(II)
KT=NOFFT2
IK=I
GO TO 102
28 CONTINUE
100 DO 103 I=1,NOFFT2
DO 104 J=1,NOFEAT
IF(FETVC2(I).NE.FETVEC(J)) GO TO 104
GO TO 32
104 CONTINUE
C COMPUTE RINLEVELS FOR FEATURE
32 I=HGTPT(J)/NUMBIN
LNUM=NUMBIN-1
M=1
N=0
K=0
DO 33 JJ=1,LNUM
34 K=K+1
N=N+FILHIS(J,K)
IF(N.LT.M) GO TO 34
M=M+1
RINLEV(I,JJ)=K-1
IF(JJ.NE.1) GO TO 111
IF(K.EQ.1) RINLEV(I,1)=1
GO TO 33
111 IF(RINLEV(I,JJ).EQ.RINLEV(I,JJ-1)) RINLEV(I,JJ)=RINLEV(I,JJ-1)+1
33 CONTINUE
RINLEV(I,NUMBIN)=255
103 CONTINUE
27 CONTINUE
C PRINT OUT SETUP6 CARDS
WRITE(6,40) (FETVC2(I),I=1,NOFET2)
40 FORMAT(1X,'SUPERVISOR INFORMATION FOR GRAYMAP',1X,'CHANNELS GRAYMAP',
*PPED ARE',30I4)
C WRITE OUT SYMBOLS
WRITE(6,41) (SYMBOL(KZ,1),KZ=1,SYMCNT)
41 FORMAT(1 SYMOLS USED IN GRAYMAP ARE',16(2X,A1))
IF(SYMDIM.EQ.2) WRITE(6,42) (SYMBOL(KZ,2),KZ=1,SYMCNT)
42 FORMAT(1H',27X,16(2X,A1))
RETURN
END
```

SET02380  
SET02390  
SET02400  
SET02410  
SET02420  
SET02430  
SET02440  
SET02450  
SET02460  
SET02470  
SET02480  
SET02490  
SET02500  
SET02510  
SET02520  
SET02530  
SET02540  
SET02550  
SET02560  
SET02570  
SET02580  
SET02590  
SET02600  
SET02610  
SET02620  
SET02630  
SET02640  
SET02650  
SET02660  
SET02670  
SET02680  
SET02690  
SET02700  
SET02710  
SET02720  
SET02730  
SET02740  
SET02750  
SET02760  
SET02770  
SET02780  
SET02790  
SET02800  
SET02810  
SET02820  
SET02830  
SET02840  
SET02850  
SET02860  
SET02870  
SET02880  
SET02890  
SET02900  
SET02910  
SET02920  
SET02930  
SET02940  
SET02950  
SET02960  
SET02970  
SET02980  
SET02990  
SET03000  
SET03010  
SET03020  
SET03030

## 8. STAT PROCESSOR

FILE: STAT

```

SUBROUTINE STAT(ARRAY, TOP)
PURPOSE.. COORDINATES THE VARIOUS ROUTINES
           FOR 'STATISTICS' STEP
IMPLICIT INTEGER (A-H, O-Z)
DOUBLE PRECISION ARRAY(1500)
DIMENSION KPPPTS(60)
INCLUDE COMHKB, LIST
STAT COMMON BLOCK
COMMON /STBASE/SUBSV1, SUBMN1, SURVP1, SURSD1, SUBCL1, SAVER1, HSTAL1,
*SPEC1, COVAR1, AVAR1, CLSID1, FLVAP1, FLVAP1, HFTAL1, FLOSV1
COMMON /STCHK/ MAXFFT, MAXCLS, MAXFLD, NOFEAT, NOFET2,
*VARSIZ, NOSPEC, NOHIST, SPCBAS, IRLOCK(30), FETVEC(30),
*FETVC(30), HISVEC(30), NOFLD, NOCLS,
* FLDINF(5), FLDPTS, CLSPTS, XSIZ, XHGH, XLOW, YSIZ
COMMON BLOCK STBASE CONTAINS THE BASE ADDRESSES FOR THE STATISTICS
STORED IN 'ARRAY'

SUBSV1 - BASE ADDRESS IN 'ARRAY' FOR SUBCLASS INFORMATION
        (5*SURNO)
        FOR EACH SUBCLASS INDEX 1-CLASS NUMBER
                                2-STARTING FIELD NUMBER
                                3-ENDING FIELD NUMBER
                                4-SUBCLASS NAME
SURMN1 - BASE ADDRESS IN 'ARRAY' FOR SUBCLASS MEANS
SURVP1 - BASE ADDRESS IN 'ARRAY' FOR SUBCLASS VARIANCES
SURSD1 - BASE ADDRESS IN 'ARRAY' FOR SUBCLASS NAMES
SUBCL1 - BASE ADDRESS IN 'ARRAY' FOR CLASS NUMBER
SAVER1 - BASE ADDRESS IN 'ARRAY' FOR TRAINING FIELD VERTICES
HSTAL1 - BASE ADDRESS IN 'ARRAY' FOR SUBCLASS HISTOGRAM TOTALS
SPEC1 - BASE ADDRESS IN 'ARRAY' FOR SPECTOGRAM INFORMATION
        (5*OSPEC)
COVAR1 - BASE ADDRESS IN 'ARRAY' FOR FIELD COVARIANCES
AVAR1 - BASE ADDRESS IN 'ARRAY' FOR FIELD MEANS
CLSID1 - BASE ADDRESS IN 'ARRAY' FOR CLASS NAMES
FLVAP1 - BASE ADDRESS IN 'ARRAY' FOR FIELD MEANS
FLVAP1 - BASE ADDRESS IN 'ARRAY' FOR FIELD VARIANCES
HFTAL1 - BASE ADDRESS IN 'ARRAY' FOR FIELD HISTOGRAM TOTALS
FLOSV1 - BASE ADDRESS IN 'ARRAY' FOR FIELD INFORMATION (10*MAXFLD)
CONTINUE
        FOR EACH FIELD INDEX 1-FIELD NAME
                                2-CLASS NUMBER
                                3-SUBCLASS NUMBER
                                4-NUMBER OF VERTICES
                                5-STARTING LINE NUMBER
                                6-ENDING LINE NUMBER
                                7-STARTING SAMPLE NUMBER
                                8-ENDING SAMPLE NUMBER
                                9-LINE INCREMENT
                                10-SAMPLE INCREMENT

CONTINUE
COMMON BLOCK STCHK CONTAINS INFORMATION NEEDED BY ROUTINES IN STAT
MAXFFT - MAXIMUM NUMBER OF CHANNELS
MAXCLS - MAXIMUM NUMBER OF CLASSES
MAXFLD - MAXIMUM NUMBER OF FIELDS
NOFEAT - NUMBER OF CHANNELS REQUESTED
VARSIZ - SIZE OF EACH COVARIANCE MATRIX (NOFEAT*(NOFEAT+1))/2
NOSPEC - NUMBER OF GROUPS OF SUBCLASSES TO PLOT
NOHIST - NUMBER OF CHANNELS TO HISTOGRAM
SPCBAS - MINIMUM RADIANCE VALUE ON Y AXIS OF SPECTRAL PLOT
IRLOCK - ARRAY CONTAINING TRIGGERS TO CERTAIN OPTIONS IN STAT
FETVEC - ARRAY OF CHANNELS SELECTED
HISVEC - ARRAY OF CHANNELS TO HISTOGRAM
NOFLD - NUMBER OF FIELDS
NOCLS - NUMBER OF CLASSES
FLDINF - FIELD INFORMATION ARRAY
FLDPTS - NUMBER OF POINTS IN FIELD
CLSPTS - NUMBER OF POINTS IN CLASS
XSIZ - XHIGH-XLOW =101
XLOW - MINIMUM X VALUE TO BE HISTOGRAMMED =0
XHGH - MAXIMUM X VALUE TO BE HISTOGRAMMED =255
YSIZ - HEIGHT OF Y AXIS IN HISTOGRAM =15
CALL SETUP1(ARRAY, TOP, MAXSUB)
CALL LEARN(ARRAY(SPEC1), ARRAY(COVAR1), ARRAY(AVAR1))

```



FILE: STAT

```

*      ARRAY (CLSID), ARRAY (SURSV), ARRAY (FLMEN), ARRAY (FLVAR) STA00800
*      ARRAY (SURMN), ARRAY (SUBVR), ARRAY (SURSO), ARRAY (SUBCL) STA00810
*      ARRAY (MFTAL), ARRAY (MSTAL), ARRAY (FLOSV),
*      ARRAY (SAVER), KEPPTS, MAXSUB) STA00830
2  WRITE (A,2) STA00840
   FORMAT (////////// 2X, '*** SSTAT - COMPLETED *** //') STA00850
   RETURN STA00860
   END STA00870
```

FILE: CLSSPC

```

SURROUTINE CLSSPC(MEAN,SURSTD,IDVEC,PTRVEC,PLOT,
  C  *TITLE,NOFEAT,FETVEC,SPCRAS)
  C  IMPLICIT INTEGER (A-H,O-Z)
  C  LOGICAL OVRFLG
  REAL MEAN(1),SUBSTD(1),BIAS,INCR,MENI,DEVI
  C  DOUBLE PRECISION DEV(1),DMEAN(1)
  DIMENSION PLOT(4,NOFEAT,49),PTRVEC(5)
  DIMENSION FETVEC(30)
  C  * SYMVFC(4),TAB(12),ERRLIN(7)
  C  INCLUDE COMR6.LIST
  COMMON/GLOBAL/HEAD(63),MAPTAP,DATEPE,SAVTAP,RMFILE,RMKEY,
  * HISFIL,HISKEY,THEFORM,FRIPTP,ERPKEY,MAPUNT,NOFILE,
  * DRUMAD,DRM-DS,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVL
  * NHSTUN,NHSTFI,STRUN,MAPFIL
  * DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
  * CROUT,PRTUNT,RANDIO
  C$FND
  C  DATA SYMVFC/'1','1','1','1','1','1','1','1','1','1','1','1',
  1  BLANK /' ' /
  C  DATA TAB /'12','20','28','36','44','52',
  1  '60','68','76','84','92','99' /
  C  DATA ERRLIN/'(1H','1','1','1','1','1','1H)','13' /
  C  DATA DASH/'-----' /
  C  C
  C  C
  C  SPEC INIZ
  WRITE(6,HEAD)
  WRITE(6,10021)
  10021 FORMAT( / 34X,'SPECTRAL PLOT (MEAN,PLUS AND MINUS ONE STD. DEV. )
  1 FOR:' / )
  WRITE(6,1002) TITLE ,(DASH,I=1,2)
  1002 FORMAT(44X,'TRAINING SUBCLASS ',A4/44X,4A4,A2/)
  WRITE(6,20021)
  WRITE(6,5002) SYMVFC(2),IDVEC
  GO TO 250
  C  C
  C  C
  C  ENTRY FLDSPC(DMEAN,DEV,IDVEC,PTRVEC,PLOT,MEAN,SUBSTD,FLDNAM,
  *NOFEAT,FETVEC,SPCRAS)
  WRITE(6,HEAD)
  WRITE(6,10021)
  WRITE(6,1004) IDVEC,(DASH,I=1,4)
  1004 FORMAT(47X,'TRAINING FIELD ',A4/46X,4A6/)
  WRITE(6,20021)
  20021 FORMAT( / 47X,'PLOT LEGEND:' / 47X,'-----' )
  WRITE(6,2002) SYMVFC(2),IDVEC
  2002 FORMAT(60X,A1,' = FIELD ',A4)
  C  C
  C  DO 100 J=1,NOFEAT
  100 MEAN(J) = DMEAN(J)
  DO 399 I=1,NOFEAT
  399 SUBSTD(I)=DEV(I)
  250 JPSTR = 2
  JPLOT = 2
  JBTA5 = 1
  C  C
  C  C
  C  COMBINED INIZ
  300 BIAS = SPCRAS
  WRITE(6,4002)BLANK
  CNT = BIAS
  OVRFLW = 0
  OVRFLG = .TRUE.
  DO 350 J=1,NOFEAT
  DO 350 I=1,4
  DO 350 K=1,44
  350 PLOT(I,J,K) = BLANK
  LOC=1
  ISTOP=NOFEAT
  IF(NOFFAT.GT.12)ISTOP=12

```

CLS00010  
CLS00020  
CLS00030  
CLS00040  
CLS00050  
CLS00060  
CLS00070  
CLS00080  
CLS00090  
CLS00100  
CLS00110  
CLS00120  
CLS00130  
CLS00140  
CLS00150  
CLS00160  
CLS00170  
CLS00180  
CLS00190  
CLS00200  
CLS00210  
CLS00220  
CLS00230  
CLS00240  
CLS00250  
CLS00260  
CLS00270  
CLS00280  
CLS00290  
CLS00300  
CLS00310  
CLS00320  
CLS00330  
CLS00340  
CLS00350  
CLS00360  
CLS00370  
CLS00380  
CLS00390  
CLS00400  
CLS00410  
CLS00420  
CLS00430  
CLS00440  
CLS00450  
CLS00460  
CLS00470  
CLS00480  
CLS00490  
CLS00500  
CLS00510  
CLS00520  
CLS00530  
CLS00540  
CLS00550  
CLS00560  
CLS00570  
CLS00580  
CLS00590  
CLS00600  
CLS00610  
CLS00620  
CLS00630  
CLS00640  
CLS00650  
CLS00660  
CLS00670  
CLS00680  
CLS00690  
CLS00700  
CLS00710  
CLS00720  
CLS00730  
CLS00740  
CLS00750  
CLS00760  
CLS00770  
CLS00780  
CLS00790

FILE: CLSSPC

```
C      SET UP 'PLOT' MATRIX
C
400  JKSV=0
      DO 690 JP=JPSTR,JPLOT
      MENBAS = (PTRVEC(JP-JBIAS)-1)*NOFEAT
4002  IF( OVRFLG) WRITE(6,4002)
      FORMAT(A4)
      OVRFLG = .FALSE.
      JK=JKSV
      JF = 0
      DO 500 JFEAT=LOC,ISTOP
         JK = JK+JFEAT
         MENI = MFAN(MENHAS+JFEAT)
         DEVI=SUHSTD(MENHAS+JFEAT)
         MENLOW = (MENI-DEVI-RIAS)/INCR+0.5
         MENHGH = (MENI+DEVI-RIAS)/INCR+0.5
         IF( MENLOW .GE. 1) GO TO 430
         OVRFLW = MENI-DEVI+0.5
         MENLOW = 1
430   IF(MENHGH .LE. NOLINE) GO TO 450
         OVRFLW = MENI+DEVI+0.5
         MENHGH = NOLINE
450   DO 490 J =MENLOW,MENHGH
490   PLOT(JP,JFEAT,J) = SYMVEC(JP)
         JF = JF + 1
         IF(OVRFLW .EQ. 0) GOTO 500
C*****JF = JF + 1 THIS STATEMENT MOVED UP ONE LINE *****
         FRRLIN(3)=TAR(JF)
         WRITE(6,ERRLIN) SYMVEC(JP),OVRFLW
         OVRFLG = .TRUE.
         OVRFLW = 0
500   CONTINUE
690   CONTINUE
C
C      PRINT OUT 'PLOT' MATRIX
      WRITE(6,3004) CNT,CNT
3004  FORMAT(4X,I3,2X,'I-----I',12('-----I'),2X,I3)
      CNT = CNT+INCR
700  DO 790 K=1,NOLINE
      WRITE(6,7002)CNT,((PLOT(I,J,K),I=1,4),J=LOC,ISTOP)
      WRITE(6,7003)CNT
7002  FORMAT(4X,I3,2X,'I',12(4X,4A1))
7003  FORMAT(' ',111X,'I',2X,I3)
      CNT = CNT+INCR
790  CONTINUE
C
800  WRITE(6,3004) CNT, CNT
C
      WRITE(6,8002) ( FETVEC(I), I=LOC,ISTOP)
8002  FORMAT( / 1X, 'CHANNEL NO.', 3X, I2, ( 11(6X,I2) ) )
      WRITE(6,8003) (DASH,I=1,3)
8003  FORMAT(1X, 3A4 ///)
C
      IF(ISTOP.EQ.NOFEAT)RETURN
      CNT=BIAS
      JKSV=JK
      LOC=LOC+12
      ISTOP=ISTOP+12
      IF(ISTOP.GT.NOFEAT)ISTOP=NOFEAT
      GO TO 400
C
C-----
C
      ENTRY MULSPC(MFAN,SUHSTD,JDVEC,PTRVEC,PLOT,
      *NOFEAT,FETVEC,SPCRAS)
      DIMENSION JDVEC(1:1),RUF(4)
      WRITE(6,20021)
      JPSTR = 1
      JPLOT = PTRVEC(5)
      IF(JPLOT .NE. 1) GOTO 900
      JPSTR = 2
      JPLOT = 2
      JBIAS = 1
      WAT=PTRVEC(1)
      WRITE(6,4002) SYMVEC(2),JDVEC(1,WAT)
      GOTO 300
900  DO ?? JKL = JPSTR,JPLOT
      WAT=PTRVEC(JKL)
```

FILE: CLSSPC

```
22 RUF(JKL)=JDVEC(J,WAT)
WRITE(6,9002) (SYMVEC(I),RUF(I),I=JPSTR,JPLOT)
9002 FORMAT(57X,A1,' = SUBCLASS ',A4)
      JRIAS = 0
      GO TO 300
      END
```

CLS01590  
CLS01600  
CLS01610  
CLS01620  
CLS01630  
CLS01640

FILE: FLDCOV

```

SUBROUTINE FLDCOV(COR,DEV,MEAN,VAR,PTS,GO,FLDNAM,
NOFEAT,MAXFFT,VARSIZ)
C
C IMPLICIT INTEGER (A-H,O-Z)
C INCLUDE COMPKA.LIST
COMMON/GLOBAL/MEAN(63),MAYTAP,DATAPF,SAVTAP,BMFILE,BMKEY,
HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE,
DRUMAD,DRMADS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
,NHSTUN,NHSTFI,SCTRUN,MAPFIL
DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
CRDUNT,PRUNT,WANDIO
CSEND
C PURPOSE.. CALCULATES THE COVARIANCE AND CORRELATION MATRICES
C FROM THE RAW DATA FOR THE FIELDS
C-----
C DOUBLE PRECISION COR(VARSIZ),DEV(NOFEAT),MEAN(NOFEAT),VAR(VARSIZ)
C-----
DATA RCDT,0/ '2' / , DASH/'-----' /
C-----
IF( GO ,NE. 1) GO TO 20
WRITE(A,MEAN)
WRITE(A,10) NOFEAT
10 FORMAT(10// 'THE MEAN,STANDARD DEVIATION,COVARIANCE,AND CORRELATIO
IN ' / 12 ' CHANNELS) FOR:' //)
WRITE(A,11) FLDNAM ,(DASH,I=1,3)
11 FORMAT(150,'TRAINING FIELD ',A4/T49,4A4/)

EQUATIONS :
          **      N          **
          **      *   **     **
COVAR(1,2) = 1
          **      * X X - U U
          **      * 1 2 1 2
          **      *   **     **
          **      * J=1     **
          **      *   **     **
          **      * N      **
          **      *   **     **
MEAN(1) = 1
          **      * X = U
          **      * 1 1
          **      *   **     **
          **      *   **     **
STDEV(2) = ** COVAR(2,2)
          **      *   **     **

20 JK = 0
PTS1 = PTS
PTS2 = PTS-1
IF (PTS2 .LT. 1) PTS2 = 1
N = NOFEAT
DO 40 JA=1,N
J = JA
DO 30 K=1,J
JK = JK+1
VAR(JK) = (VAR(JK)-MEAN(J)*MEAN(K)/PTS1)/PTS2
30 CONTINUE
DEV(J) = DSQRT(DABS(VAR(JK)))
40 CONTINUE
JK = 0
DO 50 J=1,NOFEAT
MEAN(J) = MEAN(J)/PTS1
DO 50 K=1,J
JK = JK+1
COV(JK) = 0.0
IF (DEV(K)*DEV(J).LT.1.0E-25) GO TO 50
COR(JK) = VAR(JK)/(DEV(J)*DEV(K))

```

FLD00010  
FLD00020  
FLD00030  
FLD00040  
FLD00050  
FLD00060  
FLD00070  
FLD00080  
FLD00090  
FLD00100  
FLD00110  
FLD00120  
FLD00130  
FLD00140  
FLD00150  
FLD00160  
FLD00170  
FLD00180  
FLD00190  
FLD00200  
FLD00210  
FLD00220  
FLD00230  
FLD00240  
FLD00250  
FLD00260  
FLD00270  
FLD00280  
FLD00290  
FLD00300  
FLD00310  
FLD00320  
FLD00330  
FLD00340  
FLD00350  
FLD00360  
FLD00370  
FLD00380  
FLD00390  
FLD00400  
FLD00410  
FLD00420  
FLD00430  
FLD00440  
FLD00450  
FLD00460  
FLD00470  
FLD00480  
FLD00490  
FLD00500  
FLD00510  
FLD00520  
FLD00530  
FLD00540  
FLD00550  
FLD00560  
FLD00570  
FLD00580  
FLD00590  
FLD00600  
FLD00610  
FLD00620  
FLD00630  
FLD00640  
FLD00650  
FLD00660  
FLD00670  
FLD00680  
FLD00690  
FLD00700  
FLD00710  
FLD00720  
FLD00730  
FLD00740  
FLD00750  
FLD00760  
FLD00770  
FLD00780  
FLD00790

FILE: FLNCOV

```

C          50 CONTINUE
C          -----
C          IF (GO.EQ. 0) RETURN
C          DO 70 LOC=1,NOFEAT,12
C          STOP = LOC+11
C          IF (STOP.GT. NOFEAT) STOP = NOFEAT
C          WRITE (6,100) (MEAN(I), I=LOC,STOP)
C          WRITE (6,100) (DASH,I=1,2)
C          WRITE (6,110) (DEV(I), I=LOC,STOP)
C          WRITE (6,100) (DASH,I=1,2)
C          WRITE (6,120)
C          70 CONTINUE
C          100 FORMAT(10MEAN:1,12F9.2 )
C          1001 FORMAT(1X,A4,A2/)
C          110 FORMAT(10ST DEV:1, 12F9.2 )
C          120 FORMAT('0')
C
C          WRITE (6,130) (DASH, I=1,5)
C          130 FORMAT('0' / ' COVARIANCE MATRIX' / 1X,4A4,A2 )
C          CALL DWRIMX(VAR,NOFEAT,BCDTWO)
C          IF (NOFEAT.LF. MAXFET) GO TO 140
C          WRITE (6,HEAD)
C
C          140 WRITE (6,150) (DASH, I=1,5)
C          150 FORMAT(1X) // ' CORRELATION MATRIX' /1X,5A4 )
C          CALL DWRIMX(COR,NOFEAT,BCDTWO)
C          RETURN
C          -----
C          FENTRY CLSCOV(COM,DEV,MEAN,VAR,PTS,GO,
C          *TITLE,NOFEAT,MAXFET,VARSIZ)
C          IF (GO.EQ.0) GO TO 20
C          WRITE (6,HEAD)
C          WRITE (6,10) NOFEAT
C          WRITE (6,100) TITLE (DASH,I=1,5)
C          160 FORMAT(147,'TRAINING SUBCLASS' ,A4/T4H,4A4,A2)
C          GO TO 20
C          END
C          FLD00800
C          FLD00810
C          FLD00820
C          FLD00830
C          FLD00840
C          FLD00850
C          FLD00860
C          FLD00870
C          FLD00880
C          FLD00890
C          FLD00900
C          FLD00910
C          FLD00920
C          FLD00930
C          FLD00940
C          FLD00950
C          FLD00960
C          FLD00970
C          FLD00980
C          FLD00990
C          FLD01000
C          FLD01010
C          FLD01020
C          FLD01030
C          FLD01040
C          FLD01050
C          FLD01060
C          FLD01070
C          FLD01080
C          FLD01090
C          FLD01100
C          FLD01110
C          FLD01120
C          FLD01130
C          FLD01140
C          FLD01150
C          FLD01160
C          FLD01170
C          FLD01180
C          FLD01190
C          FLD01200
C          FLD01210

```

FILE LEARN

```

SUBROUTINE LEARN(SPEC,COVAR,AVAK,CLSDES,SUBSAV,          LEA00010
*             FLDMEN,FLDVAR,SUBMEN,SUBVAR,SUBSTD,      LEA00020
*             SUBCLS,HFTALY,HSTALY,FLUSAV,SAVERT,KEPPTS,MAXSUB) LEA00030
C
  IMPLICIT INTEGER(A-Z)
  REAL XSCALE,XSHFT,
* COVAR(VARSIZ),AVAK(NUFEAT,MAXSUB),SUBSTU(NUFEAT,MAXSUB) LEA00040
* DOUBLE PRECISION FLDMEN(NUFEAT),FLDVAR(VARSIZ),SUBMEN(NUFEAT), LEA00050
* SUBVAR(VARSIZ),CDR(465),DEV(30) LEA00060
C
  INCLUDE CUMK6,LIST LEA00070
C
  INCLUDE CUMK8,LIST LEA00080
C
  INCLUDE CUMK6,LIST LEA00090
  DIMENSION HED1(15),HED2(15),DATE(3),CUMEN(15) LEA00100
  EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)), LEA00110
  (HED2(1),HEAD(30)),(CUMEN(1),HEAD(48)) LEA00120
  2 COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY, LEA00130
* HMFIL,HISKEY,TRFUM,ERLPT,ERPKEY,MAPUNT,NUFILE, LEA00140
* DRUMAD,DRMMS,PASIZ,DATFIL,STAFIL,ASAV,ASAVFL LEA00150
* ,NHSTUN,NHSTFI,ICTRUN,MAPFIL LEA00160
* ,DUTUNT,DUTFIL,ICHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, LEA00170
* CROUNT,PRUNT,KANDIU LEA00180
C STAT COMMON BLOCK LEA00190
  COMMON /STBASE/SUBSV1,SUBMN1,SUBVR1,SUBSD1,SUBCL1,SAVER1,HSTAL1, LEA00200
* SPEC1,CUVAL1,AVAK1,CLSID1,FLMEN1,FLVAK1,HFTAL1,FLUSV1 LEA00210
  COMMON /STCBLK/ MAXFET,MAXCLS,MAXFLD,NUFEAT,NUFET2, LEA00220
* VARSIZ,NUSPEC,NUMIST,SPCBAS,IBLUCK(30),FEIVEC(30), LEA00230
* FETVC2(30),HISVEC(30),NUFLD,NUCLS, LEA00240
* FLDINF(6),FLDPTS,CLSPTS,XSIZ,AMH,XLUW,YSIZ LEA00250
C$END LEA00260
C
  DIMENSION LHIST(30),VERTCS(2,11),FL(8) LEA00270
  DIMENSION SPEC(5,NUSPEC),FLDSAV(10,MAXFLD),CLSDES(MAXSUB), LEA00280
* HFTALY(NUMIST,XSIZ),HSTALY(NUMIST,XSIZ),KEPPTS(MAXSUB), LEA00290
* IDATA(12000),DUMPTR(5),SUBSAV(4,MAXSUB), LEA00300
* SAVERT(22,MAXFLD),SUBCLS(1),SUBDES(500) LEA00310
C
  EQUIVALENCE (LHIST,BESTVC) LEA00320
  EQUIVALENCE ((IBLUCK(1),NUTHG),(IBLUCK(2),PCHKEY), LEA00330
* (IBLUCK(3),SSFKEY),(IBLUCK(4),CFDKEY), LEA00340
* (IBLUCK(5),HSBKEY),(IBLUCK(6),HFDKEY), LEA00350
* (IBLUCK(7),SSLKEY),(IBLUCK(8),SFDKEY), LEA00360
  4 (IBLUCK(9),NUTH2),(IBLUCK(10),CALKEY), LEA00370
  5 (IBLUCK(11),PCFKEY),(IBLUCK(12),PCCLKY), LEA00380
  6 (IBLUCK(13),TSTRKY),(IBLUCK(14),TRNKY), LEA00390
  7 (IBLUCK(15),THRSKY),(IBLUCK(16),STATKY), LEA00400
  8 (IBLUCK(17),PCALKY) LEA00410
  EQUIVALENCE (IDATA(1),CDR(1)) LEA00420
  EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND), LEA00430
  1 (FLDINF(3),LININC),(FLDINF(4),SAMSTR), LEA00440
  2 (FLDINF(5),SAMEND),(FLDINF(6),SAMINC), LEA00450
  3 (FLDINF(7),FLDIYP) LEA00460
C
  DATA ENDCKD/'SEND',DUMPTR/1,3*0,1/,DASH/'----'/, LEA00470
* BLANK/' ',OPAK/'(',CPAR/')',CUMMA/'',PUNCH/7/ LEA00480
C-----LEA00490
C-----LEA00500
C-----LEA00510
C-----LEA00520
C-----LEA00530
C-----LEA00540
C-----LEA00550
C-----LEA00560
C-----LEA00570
C-----LEA00580
C-----LEA00590
C-----LEA00600
  INIZ LEA00610
C-----LEA00620
C* SET UP LOGICAL ARRAY FOR FEATURES TO BE HISTOGRAMMED. LEA00630
  SUBNU=0 LEA00640
  DO 7 I=1,NUMIST LEA00650
  DO 5 J=1,NUFEAT LEA00660
  IF(HISVEC(I).EQ.FETVEC(J))GO TO 6 LEA00670
  5 CONTINUE LEA00680
  LHIST(I)=J LEA00690
  7 CONTINUE LEA00700
  DO 922 I=1,MAXSUB LEA00710
  922 SUBCLS(I)=0 LEA00720
C* READ HEADER RECORD ON DATA TAPE LEA00730
  CALL TAPHDR(DATAPE,DATFIL) LEA00740
  REWIND SAVTAP LEA00750
  IF(STAFIL.EQ.0) GO TO 541 LEA00760
  CALL FSDSFL(SAVTAP,STAFIL,NSTAT) LEA00770
  IF(NSTAT.EQ.0) GO TO 541 LEA00780
  WRITE(6,542) NSTAT LEA00790

```

QUALITY

FILE LEARN

```

542 FORMAT(' BAD POSITIONING OF SAVTAP, TERMINATING ',I3) LEA00800
CALL CMEKK LEA00810
541 CONTINUE LEA00820
CALSW = CALKEY LEA00830
BADFLG = 0 LEA00840
NOFLD = 0 LEA00850
NOCLS = 0 LEA00860
10 CONTINUE LEA00870
WRITE(6,HEAD) LEA00880
IF(HSBKEY+HFDKEY.EQ.0) GO TO 14 LEA00890
XSCALE=FLOAT(1-XSIZ)/(XHIGH-XLOW) LEA00900
XSHFT=-XHIGH*XSCALE+1.0 LEA00910
DO 20 I=1,XSIZ LEA00920
DO 20 J=1,NUHIST LEA00930
20 HFTALY(J,I) = 0 LEA00940
IF(HSBKEY+HFDKEY.NE.2) GO TO 14 LEA00950
DO 30 I=1,XSIZ LEA00960
DO 30 J=1,NUHIST LEA00970
30 HSTALY(J,I)=0 LEA00980
GO TO 14 LEA00990
C CLASSES LEA1000
11 READ(30,12) TITLE LEA1010
12 FORMAT(10X,A4) LEA1020
REWIND 30 LEA1030
NOCLS=NOCLS+1 LEA1040
CLSTOT=NOCLS LEA1050
CLSDS(NUCLS)=TITLE LEA1060
GO TO 14 LEA1070
C SUBCLASSES LEA1080
13 READ(30,12) TITLE LEA1090
REWIND 30 LEA1100
SUBNU=SUBNU+1 LEA1110
IF(SUBNU.GT.MAXSUB) GO TO 490 LEA1120
SCLTOT=SUBNU LEA1130
SUBSAV(4,SUBNU)=TITLE LEA1140
C STARTING FIELD LEA1150
SUBSAV(2,SUBNU) = NOFLD + 1 LEA1160
SUBSAV(1,SUBNU)=NOCLS LEA1170
SUBCLS(NUCLS)=SUBCLS(NUCLS)+1 LEA1180
C READ FIELD CARD LEA1190
14 CFLAG=LAREAD(FLDNAM,VERTCS,FLDINF,NC) LEA1200
C END, CLASS, AND SUBCLASS LEA1210
IF (CFLAG .EQ. 0) GO TO 60 LEA1220
IF (CFLAG .EQ. -1) GO TO 11 LEA1230
IF (CFLAG .EQ. -2) GO TO 13 LEA1240
NOFLD=NOFLD+1 LEA1250
IF(NOFLD.GT.MAXFLD) GO TO 510 LEA1260
FLDSAV(1,NOFLD)=FLDNAM LEA1270
FLDSAV(2,NOFLD)=NOCLS LEA1280
FLDSAV(3,NOFLD)=SUBNU LEA1290
FLDSAV(4,NOFLD)=NC LEA1300
FLDSAV(5,NOFLD)=LINSTR LEA1310
FLDSAV(6,NOFLD)=LINEND LEA1320
FLDSAV(7,NOFLD)=SAMSTR LEA1330
FLDSAV(8,NOFLD)=SAMEND LEA1340
FLDSAV(9,NOFLD)=LININC LEA1350
FLDSAV(10,NOFLD)=SAMINC LEA1360
SUBSAV(3,SUBNU) = NOFLD LEA1370
K=0 LEA1380
DO 111 J=1,11 LEA1390
DO 111 I=1,2 LEA1400
K=K+1 LEA1410
111 SAVERT(K,NOFLD)=VERTCS(I,J) LEA1420
TOTVRT=1+TOTVRT+NC LEA1430
GO TO 14 LEA1440
60 CONTINUE LEA1450
WRITE(SAVTAP) NOCLS,SUBNU,NOFEAT,NOFLD,TOTVRT,(FETVEC(I),I=1,NOFEAT) LEA1460
* T) LEA1470
DO 61 I=1,NOFLD LEA1480
TNC=2+FLDSAV(4,I) LEA1490
WRITE(SAVTAP) (FLDSAV(J,I),J=1,4) LEA1500
61 WRITE(SAVTAP) (SAVERT(J,I),J=1,TNC) LEA1510
WRITE(SAVTAP) (CLSDS(J),J=1,NUCLS), (SUBCLS(J),J=1,NOCLS), LEA1520
* (SUBSAV(4,J),J=1,SUBNU) LEA1530
IF(PCHKEY.NE.1) GO TO 62 LEA1540
WRITE(PCHUNT,63) LEA1550
63 FORMAT('MODULE TRAINING FIELD DECK') LEA1560
WRITE(PCHUNT,64) NOCLS,SUBNU,NOFEAT,NOFLD,TOTVRT LEA1570
64 FORMAT('NUCLS ',I4,' NUSUB ',I2,' NOFEAT ',I2,' NOFLD ',I3, LEA1580

```



FILE LEARN

```

* ' TUTVRT ',14)
WRITE(PCHUNT,165) (FETVEC(I),I=1,NUFEAT)
165 FORMAT('CHNVEC',4X,30I2)
DO 65 I=1,NUFLD
WRITE(PCHUNT,66) (FLDSAV(J,I),J=1,4)
66 FORMAT(A4,6X,12,6X,12,8X,12)
TNC=2*FLDSAV(4,1)
65 WRITE(PCHUNT,67) (SAVERT(J,I),J=1,TNC)
67 FORMAT('VERTICES ',14I5)
WRITE(PCHUNT,68) (CLSDES(J),J=1,NUCLS)
68 FORMAT('CLSDES ',9(2X,A4))
WRITE(PCHUNT,69) (SUBCLS(J),J=1,NUCLS)
69 FORMAT('SUBNU ',24(1X,I2))
WRITE(PCHUNT,90) (SUBSAV(4,J),J=1,SUBNU)
90 FORMAT('SUBDES ',10(A4,1X))
62 CONTINUE
WRITE(6,41) (DASH,I=1,20)
DO 40 K=1,NUFLD
JJ=2*(FLDSAV(4,K)-1)
MP=FLDSAV(2,K)
KJ=10
IF(JJ.LE.10) KJ=JJ
MPP=FLDSAV(3,K)
WRITE(6,42) K,FLDSAV(1,K),CLSDES(MP),SUBSAV(4,MPP),FLDSAV(10,K),
*FLDSAV(9,K),((OPAK,SAVERT(I,K),CUMMA,SAVERT(I+1,K),CPAK),I=1,KJ,2)
IF(JJ.LE.10) GO TO 2017
WRITE(6,43) ((OPAK,SAVERT(I,K),CUMMA,SAVERT(I+1,K),CPAK),I=1,JJ,2)
*)
2017 CONTINUE
42 FORMAT(4X,14,2X,A4,4X,A4,4X,A4,5X,14,3X,14,4X,
*5(A1,14,A1,14,A1,2X))
43 FORMAT(5X,5(A1,14,A1,14,A1,2X))
40 CONTINUE
41 FORMAT(/TSU,'TRAINING FIELDS'/T49,4A4//
*7X,'FIELD',I36,'SAMPLE LINE'//
*5X,'NU. NAME',4X,'CLASS',3X,'SUBCLASS INC INC',
*4X,'VERTICES(SAMPLE,LINE)'//
*4X,3A4,2X,A4,A2,2X,2A4,1X,3A4,4X,5A4,A1)
SUBNU=0
70 SUBNU=SUBNU+1
SUBPTS=0
DO 71 I=1,NUFEAT
71 SUBMEN(I)=0.
DO 72 I=1,VARSIZ
72 SUBVAK(I)=0.
FIELD1=SUBSAV(2,SUBNU)
FIELDL=SUBSAV(3,SUBNU)
DO 301 N=FIELD1,FIELDL
DO 73 I=1,NUFEAT
73 FLDMEN(I)=0.
DO 74 I=1,VARSIZ
74 FLDVAK(I)=0.
LINSTR=FLDSAV(5,N)
LINEND=FLDSAV(6,N)
LININC=FLDSAV(9,N)
SAMSTR=FLDSAV(7,N)
SAMEND=FLDSAV(8,N)
SAMINC=FLDSAV(10,N)
CALL FLDINT(FLDINF,FETVEC,NUFEAT)
LINES=(LINEND-LINSTR)/LININC+1
PTS=(SAMEND-SAMSTR)/SAMINC+1
NSAMP=PTS
FLDPTS=0
DO 17 JLINES=1,LINES
CALL LINEKD(IDATA,ENDTAP)
IF(ENDTAP.EQ.-1) GO TO 16
IF(JLINES.NE.1) GO TO 8
NLINES=LINSTR
GO TO 9
8 NLINES=NLINES+LININC
9 CONTINUE
K=0
DO 93 I=1,11
DO 93 J=1,2
K=K+1
93 VERTCS(J,I)=SAVERT(K,N)
CALL FLDINT(VERTCS,FLDSAV(4,N),FL,NLINES,NS,JJJ)
KK=0
NN=1
LEAO1590
LEAO1600
LEAO1610
LEAO1620
LEAO1630
LEAO1640
LEAO1650
LEAO1660
LEAO1670
LEAO1680
LEAO1690
LEAO1700
LEAO1710
LEAO1720
LEAO1730
LEAO1740
LEAO1750
LEAO1760
LEAO1770
LEAO1780
LEAO1790
LEAO1800
LEAO1810
LEAO1820
LEAO1830
LEAO1840
LEAO1850
LEAO1860
LEAO1870
LEAO1880
LEAO1890
LEAO1900
LEAO1910
LEAO1920
LEAO1930
LEAO1940
LEAO1950
LEAO1960
LEAO1970
LEAO1980
LEAO1990
LEAO2000
LEAO2010
LEAO2020
LEAO2030
LEAO2040
LEAO2050
LEAO2060
LEAO2070
LEAO2080
LEAO2090
LEAO2100
LEAO2110
LEAO2120
LEAO2130
LEAO2140
LEAO2150
LEAO2160
LEAO2170
LEAO2180
LEAO2190
LEAO2200
LEAO2210
LEAO2220
LEAO2230
LEAO2240
LEAO2250
LEAO2260
LEAO2270
LEAO2280
LEAO2290
LEAO2300
LEAO2310
LEAO2320
LEAO2330
LEAO2340
LEAO2350
LEAO2360
LEAO2370

```

FILE LEARN

```

KC=0
DO 1 J=1,NUFEAT
INDEX1=(J-1)*NSAMP
JJ=J
DO 3 K=1,JJ
KK=KK+1
INDEX2=(K-1)*NSAMP
L=1
DO 2 JPT=1,NSAMP
KPT=(JPT-1)*SAMINC+SAMSTR
DO 4 JK=L,JJ,2
IF(KPT.LT.FL(JK)) GO TO 2
IF(KPT.GT.FL(JK+1)) GO TO 15
IDJ=IDATA(INDEX1+JPT)
IF(J.EQ.1) FLDP1,=FLDP1+1
IF(K.EQ.1) FLDMEN(J)=FLDMEN(J)+IDJ
FLDVAR(KK)=FLDVAR(KK)+IDJ*IDATA(INDEX2+JPT)
IF(LHIST(NN).NE.J) GO TO 2
KC=1
IPUT=IDATA(JPT+INDEX1)*XSCALE+XSHT+0.501
IF(IPUT.LT.1) IPUT=1
IF(IPUT.GT.XSIZ) IPUT=XSIZ
HFTALY(NN,IPUT)=HFTALY(NN,IPUT)+1
IF(JPT.EQ.NSAMP) NN=NN+1
GO TO 2
15 L=L+2
IF(L.GT.JJJ) GO TO 53
4 CONTINUE
2 CONTINUE
IF(JPT.EQ.NSAMP) GO TO 3
53 IF(KC.EQ.1) NN=NN+1
KC=0
3 CONTINUE
1 CONTINUE
17 CONTINUE
16 CONTINUE
SUBPTS=SUBPTS+FLDP1
DO 200 I=1,NUFEAT
200 SUBMEN(I)=SUBMEN(I)+FLDMEN(I)
DO 21 I=1,VARSIZ
21 SUBVAR(I)=SUBVAR(I)+FLDVAR(I)
TITLE=SUBSAV(4,SUBNO)
IF(CFKEY+SFOKEY+HFDKEY.EQ.0) GO TO 301
IF(CFKEY+SFOKEY.EQ.0) GO TO 280
CALL FLDCOV(COV,DEV,FLDMEN,FLDVAR,FLDP1,CFKEY,FLDSAV(1,N),
*NUFEAT,MAXFET,VARSIZ)
C PLOT SPECTRAL RESPONSE FOR FIELDS
IF(SFOKEY.EQ.0) GO TO 270
CALL FLDSPEC(FLDMEN,DEV,FLDSAV(1,N),DUMPTK,(DATA,
*FLDMEN,FLDVAR,TITLE,NUFEAT,FETVEC,SPCBAS)
270 IF(HFDKEY.EQ.0) GO TO 301
280 IF(HSBKEY.EQ.0) GO TO 300
DO 290 I=1,XSIZ
DO 290 J=1,NOHIST
290 HSTALY(J,I)=HSTALY(J,I)+HFTALY(J,I)
300 CONTINUE
CALL FLDHIS(HFTALY,IDATA,FLDSAV(1,N),XSIZ,XMGH,XLOW,YSIZ,
*NOHIST,FLDP1,TITLE,HISVEC)
301 CONTINUE
C CALCULATE COVAR MTX AND MEAN VECTOR FOR SUBCLASS
CALL CLSCOV(COV,DEV,SUBMEN,SUBVAR,SUBPTS,SSFKEY,
*TITLE,NUFEAT,MAXFET,VARSIZ)
C SAVE SUBCLASS MEAN,COVAR,STD DEV
130 DO 31 I=1,NUFEAT
AVAR(1,SUBNO)=SUBMEN(I)
31 SUBSTD(1,SUBNO)=DEV(I)
DO 32 J=1,VARSIZ
32 COVAR(J)=SUBVAR(J)
KEPPTS(SUBNO)=SUBPTS
C PLOT SPECTRAL RESPONSE FOR EACH SUBCLASS
IF(SSLKEY.EQ.0) GO TO 33
CALL CLSSPEC(AVAR(1,SUBNO),SUBSTD(1,SUBNO),TITLE,DUMPTK,IDATA,
*TITLE,NUFEAT,FETVEC,SPCBAS)
C PRINT SUBCLASS HIST
33 IF(HSBKEY.EQ.0) GO TO 390
TITLE=SUBSAV(4,SUBNO)
IF(HFDKEY.EQ.0) GO TO 380
CALL CLSHIS(HSTALY,IDATA,TITLE ,XSIZ,XMGH,XLOW,YSIZ,
*NOHIST,FLDP1,HISVEC)

```

LEAU2380  
LEAU2390  
LEAU2400  
LEAU2410  
LEAU2420  
LEAU2430  
LEAU2440  
LEAU2450  
LEAU2460  
LEAU2470  
LEAU2480  
LEAU2490  
LEAU2500  
LEAU2510  
LEAU2520  
LEAU2530  
LEAU2540  
LEAU2550  
LEAU2560  
LEAU2570  
LEAU2580  
LEAU2590  
LEAU2600  
LEAU2610  
LEAU2620  
LEAU2630  
LEAU2640  
LEAU2650  
LEAU2660  
LEAU2670  
LEAU2680  
LEAU2690  
LEAU2700  
LEAU2710  
LEAU2720  
LEAU2730  
LEAU2740  
LEAU2750  
LEAU2760  
LEAU2770  
LEAU2780  
LEAU2790  
LEAU2800  
LEAU2810  
LEAU2820  
LEAU2830  
LEAU2840  
LEAU2850  
LEAU2860  
LEAU2870  
LEAU2880  
LEAU2890  
LEAU2900  
LEAU2910  
LEAU2920  
LEAU2930  
LEAU2940  
LEAU2950  
LEAU2960  
LEAU2970  
LEAU2980  
LEAU2990  
LEAU3000  
LEAU3010  
LEAU3020  
LEAU3030  
LEAU3040  
LEAU3050  
LEAU3060  
LEAU3070  
LEAU3080  
LEAU3090  
LEAU3100  
LEAU3110  
LEAU3120  
LEAU3130  
LEAU3140  
LEAU3150  
LEAU3160

FILE LEARN

```

GO TO 390
380 CALL CLSHIS(HFTALY, IDATA, TITLE ,XSIZ,XMH,XLOW,YSIZ, LEA03170
*NOMIST,FLDPTS,HISVEC) LEA03180
390 WRITE(SAVTAP) KEPPTS(SUBNO),(CUVAR(I),I=1,VAKSIZ), LEA03190
* (AVAR(I,SUBNO),I=1,NUFEAT) LEA03200
IF(PCHKY.NE.1) GO TO 94 LEA03210
WRITE(PCHUNT,95) KEPPTS(SUBNO) LEA03220
95 FOKMAT('NUPTS ',6X,18) LEA03230
WRITE(PCHUNT,96) (AVAR(I,SUBNO),I=1,NUFEAT) LEA03240
96 FOKMAT('MEANS',5E15.8) LEA03250
WRITE(PCHUNT,97) (CUVAR(I),I=1,VAKSIZ) LEA03260
97 FOKMAT('CUVAR',5E15.8) LEA03270
94 IF(SUBNO.LT.SCLTOT) GO TO 70 LEA03280
ENDFILE SAVTAP LEA03290
REWIND SAVTAP LEA03300
----- LEA03310
PUBLISH THE MULTISPECTRAL PLOTS LEA03320
----- LEA03330
C
C
C
410 IF (SPEC(1,1).NE.0) GO TO 450 LEA03340
JK = 0 LEA03350
DO 430 I=1,NOSPEC LEA03360
DO 420 J=1,4 LEA03370
II=1 LEA03380
JJ=J LEA03390
JK =JK+1 LEA03400
SPEC(J,I) = JK LEA03410
IF(JK.EQ.SUBNO) GO TO 440 LEA03420
420 CONTINUE LEA03430
430 SPEC(5,I) = 4 LEA03440
440 SPEC(5,II)=JJ LEA03450
NOSPEC=II LEA03460
450 CONTINUE LEA03470
DO 480 I=1,NUSPEC LEA03480
K = SPEC(5,I) LEA03490
JJ = 0 LEA03500
DO 460 J=1,K LEA03510
IF(SPEC(J,I).GT.SUBNO) GO TO 460 LEA03520
JJ = JJ +1 LEA03530
SPEC(JJ,I) = SPEC(J,I) LEA03540
460 CONTINUE LEA03550
IF (JJ.EQ.0) GO TO 480 LEA03560
SPEC(5,I) = JJ LEA03570
WRITE(6,HEAD) LEA03580
WRITE(6,465) LEA03590
465 FOKMAT(/ / 27X,'COMPOSITE SPECTRAL PLOT (MEAN,PLUS AND MINUS ONE S LEA03600
10. DEV. ) FOR ' / ) LEA03610
WRITE(6,470)(SPEC(J,I),J=1,JJ) LEA03620
470 FOKMAT(38X,'TRAINING SUBCLASS(ES) ',414/41X,'-----' / ) LEA03630
DO 98 JI=1,SUBNO LEA03640
98 SUBDES(JI)=SUBSAV(4,JI) LEA03650
IF (JJ.LT.4) WRITE(6,471) ( DASH,II=1,5 ) LEA03660
471 FOKMAT(41X,4A4,A2 / ) LEA03670
CALL MULSPC(AVAR(1,1),SUBSTD(1,1),SUBDES,SPEC(1,1),IDATA, LEA03680
*NUFEAT,FETVEC,SPCBAS) LEA03690
480 CONTINUE LEA03700
CALL SETMKG(66,4,62) LEA03710
RETURN LEA03720
----- LEA03730
C
C
C
ERROR ROUTINES LEA03740
----- LEA03750
C
C
C
490 BADFLG = 2 LEA03760
WRITE(6,500) MAXSUB,MAXSUB LEA03770
500 FOKMAT(/ / 5X,'*** STAT/LEARNN MAX NO. OF',18,3X, 'SUBCLASSES EXCEL LEA03780
*EXCEED--FIRST',18,3X,'SUBCLASSES USED--REMAINDER IGNORED' / / ) LEA03790
GO TO 530 LEA03800
C
C
C
510 BADFLG = 1 LEA03810
WRITE(6,520) MAXFLD,MAXFLD,SUBNO LEA03820
520 FOKMAT(/ / / 5X,'***** STAT/LEARNN --- MAX. OF',18,3X, LEA03830
1 'FIELDS EXCEEDED --- ',18,3X,'FIELDS RETAINED FOR',18,3X, LEA03840
*'SUBCLASSES' / 5X,'***REMAINDER OF INPUT TRAINING FIELDS NOT USED' / ) LEA03850
C
C
C
530 READ (21,540)I LEA03860
540 FOKMAT (A4) LEA03870
IF (I.NE.ENDCRD) GO TO 530 LEA03880
GO TO 60 LEA03890
C
C

```

8-12  
62

STATISTICS  
SECURITY

FILE LEARN

END

LEA03960

~~8-13~~

64

FILE: SETUP1

```

SUBROUTINE SETUP1(SPCVEC, TOP, MAXSUB)                               SET00010
C                                                                    SET00020
C                                                                    SET00030
C                                                                    SET00040
C                                                                    SET00050
C                                                                    SET00060
C                                                                    SET00070
C                                                                    SET00080
C                                                                    SET00090
C                                                                    SET00100
C                                                                    SET00110
C                                                                    SET00120
C                                                                    SET00130
C                                                                    SET00140
C                                                                    SET00150
C                                                                    SET00160
C                                                                    SET00170
C                                                                    SET00180
C                                                                    SET00190
C                                                                    SET00200
C                                                                    SET00210
C                                                                    SET00220
C                                                                    SET00230
C                                                                    SET00240
C                                                                    SET00250
C                                                                    SET00260
C                                                                    SET00270
C                                                                    SET00280
C                                                                    SET00290
C                                                                    SET00300
C                                                                    SET00310
C                                                                    SET00320
C                                                                    SET00330
C                                                                    SET00340
C                                                                    SET00350
C                                                                    SET00360
C                                                                    SET00370
C                                                                    SET00380
C                                                                    SET00390
C                                                                    SET00400
C                                                                    SET00410
C                                                                    SET00420
C                                                                    SET00430
C                                                                    SET00440
C                                                                    SET00450
C                                                                    SET00460
C                                                                    SET00470
C                                                                    SET00480
C                                                                    SET00490
C                                                                    SET00500
C                                                                    SET00510
C                                                                    SET00520
C                                                                    SET00530
C                                                                    SET00540
C                                                                    SET00550
C                                                                    SET00560
C                                                                    SET00570
C                                                                    SET00580
C                                                                    SET00590
C                                                                    SET00600
C                                                                    SET00610
C                                                                    SET00620
C                                                                    SET00630
C                                                                    SET00640
C                                                                    SET00650
C                                                                    SET00660
C                                                                    SET00670
C                                                                    SET00680
C                                                                    SET00690
C                                                                    SET00700
C                                                                    SET00710
C                                                                    SET00720
C                                                                    SET00730
C                                                                    SET00740
C                                                                    SET00750
C                                                                    SET00760
C                                                                    SET00770
C                                                                    SET00780
C                                                                    SET00790
C                                                                    SET00800

PURPOSE.. HEADS AND ANALYZES SUPERVISOR CONTROL CARDS
FOR THE 'STATISTICS' STEP

IMPLICIT INTEGER (A-H,0-Z)

DIMENSION SINVEC(3), SPCVEC(5,20), CARD2(62), NUMVEC(30)
DIMENSION CINDEX(13), OPTCOD(6), EQUVEC(2), ACARD(20)

INCLUDE COMRK4
INCLUDE COMBK6, LIST
INCLUDE COMRKA, LIST
EQUIVALENCE (HED1(1), HEAD(4)), (DATE(1), HEAD(22)),
              (HED2(1), HEAD(30)), (COMENT(1), HEAD(48))
COMMON/GLOBAL/HEAD(63), MPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
* HISFIL, HISKEY, TRFORM, ERIP, ERKEY, MAPUNT, NOFILE,
* DRUMAD, DRUMDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVL
* ,NHSTUN, NHSTFI, SCTRUN, MAPFIL
* ,DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
* CROUT, PRUNT, RANDIO
C STAT COMMON BLOCK
COMMON /STBASE/SUBSV1, SUBMN1, SUBVR1, SUBSD1, SUBCL1, SAVER1, HSTAL1,
* SPEC1, COVAH1, AVAH1, CLSID1, FLMEN1, FLVAR1, HFTAL1, FLOS1
COMMON /STCALK/ MAXFET, MAXCLS, MAXFLD, NOFEAT, NOFET2,
* VARSIZ, NOSPEC, NOHIST, SPCBAS, IBLOCK(30), FETVEC(30),
* FETVC2(30), HISVEC(30), NOFLD, NOCLS,
* FLDINF(6), FLDPTS, CLSPTS, XSI7, XHGH, XLOW, YSIZ
C$END
EQUIVALENCE (IBLOCK( 1), NOTHG), (IBLOCK( 2), PCHKEY),
              (IBLOCK( 3), SSFKEY), (IBLOCK( 4), CFDKEY),
              (IBLOCK( 5), HSRKEY), (IBLOCK( 6), HFOKEY),
              (IBLOCK( 7), SSLKEY), (IBLOCK( 8), SFOKEY),
              (IBLOCK( 9), NOTHG2), (IBLOCK(10), CALKEY),
              (IBLOCK(11), PCFDKY), (IBLOCK(12), PCCLKY),
              (IBLOCK(13), ISTKEY), (IBLOCK(14), TRNKEY),
              (IBLOCK(15), TRRSKY), (IBLOCK(16), STATKY),
              (IBLOCK(17), PCALKY)

DATA SINVEC/2,.,., '= /, EQUAL/= /, BLANK/ ' /
DATA CB CD /'C', /, FB CD /'F', /, LB CD /'L', /, SR CD /'S', /, XB CD /'X', /,
1 YB CD /'Y', /, BLANK /' ', /, HB CD /'H', /
3 NB CD /'N', /, OB CD /'O', /, AB CD /'A', /
UB CD /'U', /

DATA CINDEX/13/, EQUVEC/1, /= /

DATA OPTCOD/'M', 'P', 'C', 'H', 'S', /
DATA CINDEX/'OPTI', 'CHAN', 'HIST', 'SPEC', 'IBLO',
1 'SIZE', 'DATE', '*END', 'COMM',
2 'HED1', 'HED2', 'DATA', 'STAT' /

INIZ

READ AND UNPACK HEADER RECORD TO SET MAXFET ACCORDING TO ID(5)
1 DO 2 I=1,30
  FETVEC(I) = 1
  HISVEC(I) = 1
2 IBLOCK(I) = 0
  NBLOCK = 0
  CALKEY = 0
  CCLKEY = 1
  MAXCLS = 15
  MAXSUB = 15
  MAXFET = 30
  SYMMAX = 30
  NOSPEC = 0
  SPCBAS = 75
  YSIZ = 14
  XSIZ = 0
  XLOW = 120
  XHGH = 220
  INFMT = 0
  NOFEAT = 0
```

FILE: SETUP1

```

C          CCCCCC          SET00800
          READ AND ANALYZE SUPERVISOR CONTROL CARDS          SET00810
          -----          SET00820
          SETUP REREAD BUFFER          SET00830
          CALL REREAD(30,80)          SET00840
          C 200 COL = 0          SET00850
          NOW READ THE CARD INTO THE BUFFER          SET00860
          READ(21,150)(ACARD(I),I=1,20)          SET00870
          150 FORMAT(20A4)          SET00880
          WRITE(30,150)(ACARD(I),I=1,20)          SET00890
          REWIND 30          SET00900
          2002 READ(30,2002) CODE,CARD2          SET00910
          FORMAT (A4,6X,62A1)          SET00920
          REWIND 30          SET00930
          225 WRITE(6,2252) CODE,CARD2          SET00940
          2252 FORMAT (15,A4,6X,62A1)          SET00950
          DO 230 I=1,CINMAX          SET00960
          IF (CINDEX(I) .EQ. CODE)          SET00970
          1 GO TO(10,600,700,800,1100,1200,1400,900,1500,1600,
          2 1700,1710,1720),I          SET00980
          230 CONTINUE          SET00990
          GO TO 1000          SET01000
          OPTION CARD          SET01010
          -----          SET01020
          C 10 M = NXTCHR(CARD2,COL)          SET01030
          IF (M .EQ. BLANK ) GO TO 200          SET01040
          IF ( M .EQ. OPTCOD(1) ) GO TO 20          SET01050
          SETFLG = 1          SET01060
          IF ( M .NE. NBCD ) GOTO 14          SET01070
          J = COL-1          SET01080
          M = NXTCHR(CARD2,COL)          SET01090
          IF ( M .NE. OBCD ) GOTO 12          SET01100
          SETFLG = 0          SET01110
          J = COL          SET01120
          C 12 COL = J          SET01130
          M = NXTCHR(CARD2,COL)          SET01140
          DO 15 I=2,5          SET01150
          14 IF ( M .EQ. OPTCOD(I) ) GO TO (40,30,25,30,30), I          SET01160
          15 CONTINUE          SET01170
          C 40 IF ( M .EQ. BLANK ) GO TO 200          SET01180
          M = COL + 10          SET01190
          WRITE(6,402) M          SET01200
          402 FORMAT(/ 1X,1** STAT/SETUP1 --- ERROR IN OPTION(S) REQUESTED - S          SET01210
          1CAN OF OPTION(S) DISCONTINUED AT CARD COLUMN',I5,2X,1**' /)          SET01220
          GO TO 200          SET01230
          C 20 M = FIND12(CARD2,COL,SINVEC)          SET01240
          IF ( SINVEC(M) .NE. EQUAL ) GO TO 40          SET01250
          M = NUMBER(CARD2,COL,NUMVEC,29)          SET01260
          IF( NUMVEC(30) .LE. 0 ) GO TO 40          SET01270
          MAXSUB=NUMVEC(30)          SET01280
          GO TO 10          SET01290
          C 25 J = 20          SET01300
          M = NXTCHR(CARD2,COL)          SET01310
          IF ( M .EQ. OBCD ) J=3          SET01320
          IF ( M .EQ. ABCD ) J=9          SET01330
          IF ( J .LT. 20 ) GOTO 32          SET01340
          GOTO 40          SET01350
          C 30 J = I*2-3          SET01360
          32 M = FIND12(CARD2,COL,SINVEC)          SET01370
          IF ( SINVEC(M) .NE. EQUAL ) GOTO 38          SET01380
          M = NXTCHR(CARD2,COL)          SET01390
          IF ( M .EQ. CB CD ) IBLOCK(J) = SETFLG          SET01400
          IF ( M .EQ. FB CD ) IBLOCK(J+1) = SETFLG          SET01410
          M = FIND12(CARD2,COL,SINVEC)          SET01420
          IF ( M .LE. 0 ) GOTO 200          SET01430
          GOTO 10          SET01440
          C 38 IBLOCK(J) = SETFLG          SET01450
          SET01460
          SET01470
          SET01480
          SET01490
          SET01500
          SET01510
          SET01520
          SET01530
          SET01540
          SET01550
          SET01560
          SET01570
          SET01580

```

FILE: SETUP1

```
IBLOCK(J+1) = SETFLG
IF ( M .LE. 0 ) GOTO 200
GOTO 10

C C C C C
CHANNELS
-----

600 J = NXTCHR(CARD2,COL)
IF ( J .EQ. BLANK ) GOTO 200
COL = COL-1
NOFEAT = NUMBER(CARD2,COL,FETVEC,NOFEAT)

C C C C C
ELIMINATE OUT-OF-RANGE REQUESTED FEATURES, IF ANY,
ORDER THE RESULTING FEATURE VECTOR

NM1 = NOFEAT - 1
IF( NM1 .LE. 0 ) NM1 = 1
DO 611 I=1,NM1,1
IP1 = I + 1
IF( IP1 .GT. NOFEAT ) GO TO 611
DO 610 J=IP1,NOFEAT,1
IF( FETVEC(I) .EQ. FETVEC(J) ) FETVEC(J) = I*MAXFET+J
610 CONTINUE
611 CONTINUE
II = 0
DO 612 I=1,NOFEAT,1
CHK = FETVEC(I)
IF( CHK .LE. 0 .OR. CHK .GT. MAXFET ) GO TO 612
II = II + 1
FETVEC(II) = FETVEC(I)
612 CONTINUE
NOFEAT = II
IF( NOFEAT .EQ. 0 ) GO TO 1303
NM1 = NOFEAT - 1
IF( NM1 .LE. 0 ) NM1 = 1
DO 614 I=1,NM1,1
IP1 = I + 1
IF( IP1 .GT. NOFEAT ) GO TO 614
DO 613 J=IP1,NOFEAT,1
IF( FETVEC(I) .LT. FETVEC(J) ) GO TO 613
TEMP = FETVEC(I)
FETVEC(I) = FETVEC(J)
FETVEC(J) = TEMP
613 CONTINUE
614 CONTINUE
GO TO 200

C C C C C
HISTOGRAM CARD
-----

700 J = NXTCHR(CARD2,COL)
IF ( J .EQ. BLANK ) GOTO 200
COL = COL-1
NOHIST = NUMBER(CARD2,COL,HISVEC,NOHIST)

C C C C C
ELIMINATE OUT OF RANGE REQUESTED SUBCLASSES, IF ANY, AND
ORDER THE RESULTING SUBCLASS VECTOR

NM1 = NOHIST - 1
IF( NM1 .LE. 0 ) NM1 = 1
DO 711 I=1,NM1,1
IP1 = I + 1
IF( IP1 .GT. NOHIST ) GO TO 711
DO 710 J=IP1,NOHIST,1
IF( HISVEC(I) .EQ. HISVEC(J) ) HISVEC(J) = I*SYMMAX+J
710 CONTINUE
711 CONTINUE
II = 0
DO 712 I=1,NOHIST,1
CHK = HISVEC(I)
IF( CHK .LE. 0 .OR. CHK .GT. SYMMAX ) GO TO 712
II = II + 1
HISVEC(II) = HISVEC(I)
712 CONTINUE
NOHIST = II
IF( NOHIST .EQ. 0 ) GO TO 1303
NM1 = NOHIST - 1
IF( NM1 .LE. 0 ) NM1 = 1
```

SET01590  
SET01600  
SET01610  
SET01620  
SET01630  
SET01640  
SET01650  
SET01660  
SET01670  
SET01680  
SET01690  
SET01700  
SET01710  
SET01720  
SET01730  
SET01740  
SET01750  
SET01760  
SET01770  
SET01780  
SET01790  
SET01800  
SET01810  
SET01820  
SET01830  
SET01840  
SET01850  
SET01860  
SET01870  
SET01880  
SET01890  
SET01900  
SET01910  
SET01920  
SET01930  
SET01940  
SET01950  
SET01960  
SET01970  
SET01980  
SET01990  
SET02000  
SET02010  
SET02020  
SET02030  
SET02040  
SET02050  
SET02060  
SET02070  
SET02080  
SET02090  
SET02100  
SET02110  
SET02120  
SET02130  
SET02140  
SET02150  
SET02160  
SET02170  
SET02180  
SET02190  
SET02200  
SET02210  
SET02220  
SET02230  
SET02240  
SET02250  
SET02260  
SET02270  
SET02280  
SET02290  
SET02300  
SET02310  
SET02320  
SET02330  
SET02340  
SET02350  
SET02360  
SET02370

FILE: SETUP1

	DO 714 I=1,NM1,1	SET02380
	IP1 = I + 1	SET02390
	IF( IP1 .GT. NOHIST) GO TO 714	SET02400
	DO 713 J=IP1,NOHIST,1	SET02410
	IF( HISVEC(I) .LT. HISVEC(J)) GO TO 713	SET02420
	TEMP = HISVEC(I)	SET02430
	HISVEC(I) = HISVEC(J)	SET02440
	HISVEC(J) = TEMP	SET02450
713	CONTINUE	SET02460
714	CONTINUE	SET02470
	GO TO 200	SET02480
C	SPEC CARD	SET02490
C	-----	SET02500
C	800 J = NXCCHR(CARD2, COL)	SET02510
	IF ( J .EQ. BLANK ) GOTO 200	SET02520
	COL = COL - 1	SET02530
	NOSPEC = NOSPEC + 1	SET02540
	IF( NOSPEC .GT. 20) GO TO 200	SET02550
	J = NUMBER(CARD2, COL, NUMVEC, 0)	SET02560
	IF ( J .GT. 4 ) J = 4	SET02570
	DO 810 I=1, J	SET02580
	IF ( NUMVEC(I) .LE. 0 ) GOTO 815	SET02590
810	SPCVEC(I, NOSPEC) = NUMVEC(I)	SET02600
	I = J + 1	SET02610
815	SPCVEC(5, NOSPEC) = I - 1	SET02620
	GOTO 200	SET02630
C	IBLOCK CARD	SET02640
C	-----	SET02650
C	1100 J = NXCCHR(CARD2, COL)	SET02660
	IF ( J .EQ. BLANK ) GOTO 200	SET02670
	COL = COL - 1	SET02680
	NBLOCK = NUMBER(CARD2, COL, NUMVEC, NBLOCK)	SET02690
	DO 1110 I=1, NBLOCK, 1	SET02700
	IF ( NUMVEC(I) .EQ. 1) IBLOCK(I) = 1	SET02710
1110	CONTINUE	SET02720
	GO TO 200	SET02730
C	SIZE CARD	SET02740
C	-----	SET02750
C	97 COL=COL-1	SET02760
1200	J = NXCCHR(CARD2, COL)	SET02770
	IF( J .EQ. BLANK ) GOTO 200	SET02780
	IF ( J .EQ. XHCD) GO TO 1220	SET02790
	IF ( J .EQ. SHCD) GO TO 1230	SET02800
	IF ( J .EQ. YBCD) GO TO 1240	SET02810
	GO TO 1000	SET02820
C	1220 J = NXCCHR(CARD2, COL)	SET02830
	M = FIND12(CARD2, COL, SINVEC)	SET02840
	IF( SINVEC(M) .NE. EQUAL ) GO TO 1000	SET02850
	M = NUMBER(CARD2, COL, NUMVEC, 29)	SET02860
	IF( J .EQ. LHCD) XLOW = NUMVEC(30)	SET02870
	IF( J .EQ. MHCD) XHIGH = NUMVEC(30)	SET02880
	IF( J .NE. SHCD) GO TO 97	SET02890
	XSIZE = NUMVEC(30)	SET02900
	GO TO 97	SET02910
C	1230 M = FIND12(CARD2, COL, SINVEC)	SET02920
	IF( SINVEC(M) .NE. EQUAL ) GO TO 1000	SET02930
	M = NUMBER(CARD2, COL, NUMVEC, 29)	SET02940
	SPCBAS = NUMVEC(30)	SET02950
	GO TO 97	SET02960
C	1240 M = FIND12(CARD2, COL, SINVEC)	SET02970
	IF( SINVEC(M) .NE. EQUAL ) GO TO 1000	SET02980
	M = NUMBER(CARD2, COL, NUMVEC, 29)	SET02990
	XSIZE = NUMVEC(30)	SET03000
	GO TO 97	SET03010
C	DATE CARD	SET03020
C	-----	SET03030
C	1400 M = NXCCHR(CARD2, COL)	SET03040
	IF( M .EQ. BLANK ) GO TO 200	SET03050
		SET03060
		SET03070
		SET03080
		SET03090
		SET03100
		SET03110
		SET03120
		SET03130
		SET03140
		SET03150
		SET03160



FILE: SETUP1

<pre> 999 READ (30,999) DATE     FORMAT (10X,15A4)     REWIND 30     GO TO 200      COMMENT CARD     -----  1500 READ (30,999) COMENT     REWIND 30     GO TO 200      MED1 CARD     -----  1600 READ (30,999) MED1     REWIND 30     GO TO 200      MED2 CARD     -----  1700 READ (30,999) MED2     REWIND 30     GOTO 200      DATA FILE CARD  1710 M = NXTCHR(CARD2,COL)     IF (M.EQ. BLANK) GO TO 200     IF (M.EQ. UHCD) GO TO 1715     IF (M.EQ. FHCD) GO TO 1717 1713 WRITE(6,753)     753 FORMAT(' ERROR ON DATA FILE CARD')     GO TO 200 1715 J = FIND12(CARD2,COL,EQUVEC)     IF (J.EQ. -1) GO TO 1713     M = NUMBER(CARD2,COL,DATAPE,ZERO)     COL = COL - 1     GO TO 1710 1717 J = FIND12(CARD2,COL,EQUVEC)     IF (J.EQ. -1) GO TO 1713     M = NUMBER(CARD2,COL,DATAFIL,ZERO)     DATAFIL = DATAFIL - 1     IF (DATAFIL.LT. 0) DATAFIL = 0     COL = COL - 1     GO TO 1710      STAT FILE CARD  1720 M = NXTCHR(CARD2,COL)     IF (M.EQ. BLANK) GO TO 200     IF (M.EQ. UHCD) GO TO 1725     IF (M.EQ. FHCD) GO TO 1727 1723 WRITE(6,755)     755 FORMAT(' ERROR ON STAT FILE CARD')     GO TO 200 1725 J = FIND12(CARD2,COL,EQUVEC)     IF (J.EQ. -1) GO TO 1723     M = NUMBER(CARD2,COL,SAVTAP,ZERO)     COL = COL - 1     GO TO 1720 1727 J = FIND12(CARD2,COL,EQUVEC)     IF (J.EQ. -1) GO TO 1723     M = NUMBER(CARD2,COL,STAFIL,ZERO)     STAFIL = STAFIL - 1     IF (STAFIL.LT. 0) STAFIL = 0     COL = COL - 1     GO TO 1720      CALCULATE BASES OF THE ARRAYS     -----  900 CONTINUE     IF (NOSPEC.GT. 20) NOSPEC = 20     IF (NOSPEC.NE. 0) GO TO 950     NOSPEC = (MAXCLS*3)/4     SPCVEC(1,1) = 0 </pre>	<pre> SET03170 SET03180 SET03190 SET03200 SET03210 SET03220 SET03230 SET03240 SET03250 SET03260 SET03270 SET03280 SET03290 SET03300 SET03310 SET03320 SET03330 SET03340 SET03350 SET03360 SET03370 SET03380 SET03390 SET03400 SET03410 SET03420 SET03430 SET03440 SET03450 SET03460 SET03470 SET03480 SET03490 SET03500 SET03510 SET03520 SET03530 SET03540 SET03550 SET03560 SET03570 SET03580 SET03590 SET03600 SET03610 SET03620 SET03630 SET03640 SET03650 SET03660 SET03670 SET03680 SET03690 SET03700 SET03710 SET03720 SET03730 SET03740 SET03750 SET03760 SET03770 SET03780 SET03790 SET03800 SET03810 SET03820 SET03830 SET03840 SET03850 SET03860 SET03870 SET03880 SET03890 SET03900 SET03910 SET03920 SET03930 SET03940 SET03950 </pre>
--	---

FILE: SETUP1

```

950 VARSIZ = NOFEAT*(NOFEAT+1)/2
IF (XSIZ.LE.0) XSIZ=XHGH-XLOW+1
IF (XSIZ.GT.101) XSIZ=101
SPEC1=(5*NOSPEC+1)/2*2
COVAR1=(VARSIZ+1)/2*2
AVAR1=(NOFEAT*MAXSUB+1)/2*2
CLSID1=(MAXSUB+1)/2*2
SUBSV1=(5*MAXSUB+1)/2*2
FLMEN1=NOFEAT*2
FLVAR1=VARSIZ*2
SUBMN1=NOFEAT*2
SURVR1=VARSIZ*2
SURSD1=(NOFEAT*MAXSUB+1)/2*2
SUBCL1=(MAXSUB+1)/2*2
MFTAL1=(XSIZ*NOHIST+1)/2*2*HFDKEY
HSTAL1=(XSIZ*NOHIST+1)/2*2*HSBKEY
SIZE=SPEC1*COVAR1*AVAR1*CLSID1*SUBSV1*FLMEN1*FLVAR1*
* SUBMN1*SURVR1*SURSD1*SUBCL1*MFTAL1*HSTAL1
MAXFLD=(TOP-SIZE-32)/32
IF (MAXFLD.LE.0) GO TO 1300
SPEC1=1
COVAR1=SPEC1*(5*NOSPEC+1)/2
AVAR1=COVAR1*(VARSIZ+1)/2
CLSID1=AVAR1*(NOFEAT*MAXSUB+1)/2
SUBSV1=CLSID1*(MAXSUB+1)/2
FLMEN1=SUBSV1*(5*MAXSUB+1)/2
FLVAR1=FLMEN1*NOFEAT
SUBMN1=FLVAR1*VARSIZ
SURVR1=SUBMN1*NOFEAT
SURSD1=SURVR1*VARSIZ
SURCL1=SURSD1*(NOFEAT*MAXSUB+1)/2
MFTAL1=SURCL1*(MAXSUB+1)/2
HSTAL1=MFTAL1*(XSIZ*NOHIST+1)/2*HFDKEY
FLDSV1=HSTAL1*(XSIZ*NOHIST+1)/2*HSBKEY
SAVER1=FLDSV1*(10*MAXFLD)/2
TIPTOP=SAVER1*(22*MAXFLD)/2
BADCOR=TOP-2*TIPTOP
IF (BADCOR.LT.0) GO TO 1300
PRINT OUT OPTIONS
-----
WRITE(6,HEAD)
IF (PCHKEY+SSFKEY+CFDKEY+HSHKEY+HFDKEY+SSLKEY
1 *SFOKEY+CALKEY.LE.0) GOTO 960
WRITE(6,9001)
IF (CFDKEY.EQ.1) WRITE(6,9002)
IF (SFOKEY.EQ.1) WRITE(6,9004)
IF (SSLKEY.EQ.1) WRITE(6,9006)
IF (PCHKEY.EQ.1) WRITE(6,9008)
IF (HFDKEY.EQ.1) WRITE(6,9012)
IF (HSHKEY.EQ.1) WRITE(6,9014)
IF (CALKEY.EQ.1) WRITE(6,9016)
IF (SSFKEY.EQ.1) WRITE(6,9018)
C
9001 FORMAT(1X,'YOU HAVE SELECTED THE FOLLOWING $STAT PROCESSOR OPTIO
INS: '///)
9002 FORMAT(15,' PRINT MEAN AND COVARIANCE FOR EACH FIELD')
9004 FORMAT(15,' PRINT SPECTRAL PLOT FOR EACH FIELD')
9006 FORMAT(15,' PRINT SPECTRAL PLOT FOR EACH SUBCLASS')
9008 FORMAT(15,' PUNCH MEAN AND COVARIANCE MATRIX FOR EACH SUBCLASS')
9012 FORMAT(15,' PRINT A HISTOGRAM FOR EACH FIELD')
9014 FORMAT(15,' PRINT A HISTOGRAM FOR EACH SUBCLASS')
9016 FORMAT(15,' *** USE CALIBRATED DATA ***')
9018 FORMAT(15,' PRINT MEAN AND COVARIANCE FOR EACH SUBCLASS')
960 WRITE(6,9502) HADCOR,MAXFLD,MAXSUB,(FETVEC(I),I=1,NOFEAT)
9502 FORMAT('0'///'SUPERVISOR INFORMATION: '///15,'UNUSED CORE',16,' LOCA
ATIONS',15,'MAXIMUM NO. OF FIELDS...',13/15,'MAXIMUM NO. OF SUBCLAS
SES...',13/15,'CHANNELS SELECTED ARE ',15(13,' ')/127,15(13,' '))
9504 FORMAT(15,'HISTOGRAM CHANNELS ARE ',15(13,' ')/127,15(13,' '))
IF (SFCVEC(1,1).EQ.0) GOTO 971
WRITE(6,9505)
9505 FORMAT(15,'MULTISPECTRAL PLOTS ARE...')
DO 970 J=1,NOSPEC
K=SFCVEC(5,J)
WRITE(6,9506) (SFCVEC(I,J),I=1,K)
9506 FORMAT(144,' ')/T31,' ')/4(12,' '))
970 CONTINUE
971 CONTINUE

```

SET03960  
SET03970  
SET03980  
SET03990  
SET04000  
SET04010  
SET04020  
SET04030  
SET04040  
SET04050  
SET04060  
SET04070  
SET04080  
SET04090  
SET04100  
SET04110  
SET04120  
SET04130  
SET04140  
SET04150  
SET04160  
SET04170  
SET04180  
SET04190  
SET04200  
SET04210  
SET04220  
SET04230  
SET04240  
SET04250  
SET04260  
SET04270  
SET04280  
SET04290  
SET04300  
SET04310  
SET04320  
SET04330  
SET04340  
SET04350  
SET04360  
SET04370  
SET04380  
SET04390  
SET04400  
SET04410  
SET04420  
SET04430  
SET04440  
SET04450  
SET04460  
SET04470  
SET04480  
SET04490  
SET04500  
SET04510  
SET04520  
SET04530  
SET04540  
SET04550  
SET04560  
SET04570  
SET04580  
SET04590  
SET04600  
SET04610  
SET04620  
SET04630  
SET04640  
SET04650  
SET04660  
SET04670  
SET04680  
SET04690  
SET04700  
SET04710  
SET04720  
SET04730  
SET04740

FILE: SETUP1

```

C
C
C
980 RETURN
      ERROR ROUTINES
-----
1000 WRITE (6,10002) CODE, CARD2
10002 FORMAT(///5X, '///// FROM SUBR. SETUP1 --- BAD CONTROL CARD ENC
10UNTERED --- INPUT CARD IS ...//8X,2H'',A4.6X,62A1.2H'' //)
      GO TO 200
C
1300 WRITE (6,1302)
1302 FORMAT(/// 5X, '///// FROM SUBR. SETUP1 --- DECREASE OPTIONS'
1 // 5X, '***** TERMINATING PROGRAM EXECUTION FROM SUBR. SETUP1 *
2 *** /|H|)
      GO TO 1305
1303 WRITE (6,10002) CODE, CARD2
      WRITE (6,13031) MAXFET
13031 FORMAT(//5X, 'CHECK CHANNELS ON SUBCLASS NO.S REQUESTED-CANNOT BE
1 LESS THAN OR EQUAL ZERO, OR GREATER THAN',I5//5X,
2 ***** TERMINATING PROGRAM EXECUTION FROM SUBR. SETUP1 *****
3 /|H|)
1305 CALL EXIT
      END
SET04750
SET04760
SET04770
SET04780
SET04790
SET04800
SET04810
SET04820
SET04830
SET04840
SET04850
SET04860
SET04870
SET04880
SET04890
SET04900
SET04910
SET04920
SET04930
SET04940
SET04950
SET04960
SET04970

```

## 9. ISOCLS PROCESSOR

See listings for the TESTSP processor (section 23) for an iterative self-organizing clustering procedure using sample values of pixels clustered in packed form on disk storage.

FILE ISOCLS

```

SUBROUTINE ISOCLS (ARRAY, TOP)
.....
THIS PROGRAM PERFORMS A MODIFIED VERSION OF THE CLUSTERING
ALGORITHM (ISODATA) ORIGINALLY DEVELOPED BY BALL AND HALL OF
STANFORD RESEARCH INSTITUTE. THE ALGORITHM HAS BEEN MODIFIED
ON THE RECOMMENDATIONS OF ED KAN (LEC).

THE PROGRAM EXPECTS MULTISPECTRAL SCANNER DATA
IN EITHER THE LAPSYS 22 OR THE UNIVERSAL
FORMAT. THE DATA TAPE SHOULD BE ASSIGNED TO FORTRAN UNIT 3.
.....
IMPLICIT INTEGER (A-X)
INCLUDE COMBK5,LIST
INCLUDE COMNT5,LIST
INCLUDE COMBK6,LIST
INCLUDE COMBK16,LIST
COMMON/PASS/STOP,LNCAT,NMIN,KRN,STDMAX,DLMIN,SEP,
MAP,SPTRIG,IRD,KPTS,NOPTS,PUNCH,
* ICHN,CHNTHS,ICHAIN(62),NWDS,IREGIN,REGIN1,
* IREGIN2,REGIN3,CLSNAM,NOFLD,IPI,TOTWRD,TOTPTS,
* NCLASS,NOCLS,TOTSUR,TOTFLD,TOTVRT,NOCL,NVRT
* ,NXTCLS,NOFFAT,MAXCLS,FETVEC(30),SYMMTX(62)
* ,VARSIZ,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
* ,IORDER,INUNIT,INFILE,INITM,PMIN,SUBVEC(62),NOSUB2,CHNVC(30)
* ,NOCHAN,ERCOMP,NOSEQ,MEAND0,MEANDU,
* SYMD0,SYMDU,ITRIG0,ITRIGU,DOFLAG,
* DUFLAG,D0DU,STDOTS(60),NSD0TS,SUNCOR(30),LLNCAT,
* DVERT(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),ND0U(2)
* ,MXFET1,MAXPOP
REAL SUNCOR

COMMON BLOCK 'PASS' IS USED ONLY BY THE ISOCLS PROCESSOR.

ISOCLS USES THE RANDOM ACCESS DRUM FILE AS FOUR DISTINCT FILES.
SEE DEFINITIONS OF IREGIN,BEGIN1,BEGIN2,BEGIN3 BELOW

DEFINITIONS

ISTOP - MAX. NO. OF ITERATIONS FOR THE CLUSTERING PROCEDURE
      SET IN SETUP7 ROUTINE. (USER INPUT)
LNCAT - CURRENT NO. OF CLUSTERS. SET INITIALLY IN RDFILE OR IS
      ISOCLS. THEN ONLY IN ISODAT.
NMIN - MIN. NO. OF POINTS TO ALLOW PER CLUSTER
      SET IN SETUP7 ROUTINE. (USER INPUT)
KRN - PRINT CLUSTER SUMMARY EVERY 'KRN' ITERATION(S)
      SET IN SETUP7 ROUTINE. (USER INPUT)
STDMAX - STANDARD DEVIATION FOR SPLITTING CLUSTERS
      SET IN SETUP7 ROUTINE. (USER INPUT)
DLMIN - MIN. DISTANCE BETWEEN CLUSTERS FOR COMBINING.
SEP - DISTANCE TO SEPARATE CLUSTERS. SET EITHER IN SETUP7,
      BY USER INPUT, OR IN ID
      BY USER INPUT, OR IN ISODAT.
MAP - PRINT A CLUSTER MAP EVERY 'MAP' ITERATION(S) - SETUP7
SPTRIG - TRIGGER TELLING WHETHER OR NOT 'SEP' WAS INPUT. -SETUP7
IRD - NO. OF RECORDS TO READ FROM DATA FILE. COMPUTED IN
      ISOCLS
NOPTS - NO. OF POINTS IN EACH RECORD. COMPUTER IN ISOCLS
CONTINUE
KPTS - NO. OF POINTS IN LAST RECORD. COMPUTER IN ISOCLS
PUNCH - TRIGGER TELLING WHETHER OR NOT TO PUNCH THE MODULE
      STAT DECK. - SETUP7
ICHN - TRIGGER TELLING WHETHER OR NOT CHAINING IS TO BE DONE
CHNTHS - MIN. DISTANCE BETWEEN CLUSTERS FOR CHAINING - SETUP7
ICHAIN - ARRAY CONTAINING CHAINED CLUSTER NUMBERS. SET IN
      'CHAIN' ROUTINE.
NWDS - TOTAL NO. OF WORDS AVAILABLE FOR DRUM STORAGE OF
      IMAGE DATA TO BE CLUSTERED - SET IN ISOCLS
IREGIN - BEGINNING DRUM FILE ADDRESS FOR INPUT INITIAL CLUSTER
      CENTERS - SET IN ISOCLS
REGINS - BEGINNING DRUM FILE ADDRESS FOR TEMPORARY STORAGE OF
      CLASS STATISTICS - SET IN ISOCLS ROUTINE
BEGIN1 - BEGINNING DRUM FILE ADDRESS FOR IMAGE DATA

```

FILE ISOCLS

```

C*      BEGIN2 - BEGINNING DRUM FILE ADDRESS FOR 'IPLACE' (CLUSTER TO IS000770
C*      WHICH CORRESPONDING POINT BELONGS.) IS000780
C*      CLSNAM - NAME OF CLASS CURRENTLY BEING PROCESSED. - RDDATA IS000790
C*      NOFLD - NO. OF FIELDS INPUT FOR THIS CLASS - RDDATA IS000800
C*      IPT - NO. OF WORDS OF STORAGE USED IN 'ARRAY' FOR FIELD AND IS000810
C*      CLASS INFORMATION FOR THIS CLASS. - RDDATA IS000820
C*      TOTWRD - TOTAL WORDS WRITTEN ON DRUM FILE BEGINNING AT ADDRESS IS000830
C*      BEGIN1 - RDDATA IS000840
C*      TOTPTS - TOTAL POINTS TO BE CLUSTERED FOR CURRENT CLASS - RDDATA IS000850
C*      NCLASS - NO. OF CLASSES TO BE CLUSTERED FOR CURRENT CALL TO IS000860
C*      ISOCLS, USER INPUT - SETUP7. IS000870
C*      NOCLS - CURRENT CLASS NO. - ISOCLS IS000880
C*      TOTSUB - TOTAL CLUSTERS FOR THIS CALL TO ISOCLS IS000890
C*      TOTFLD - TOTAL FIELDS FOR ALL CLASSES - ISOCLS IS000900
C*      TOTVRT - TOTAL VERTICES FOR ALL FIELDS - ISOCLS IS000910
C*      NOCL - NO. OF CLASSES SINCE LAST CALL TO SETUP - RDDATA IS000920
C*      IEY0331 COMMENTS DELETED *****
C*      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,RMFILE,RMKEY, IS001200
C*      HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE, IS001210
C*      DPUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL IS001220
C*      .NHSTUN,NHSTFI,SCTRUN,MAPFIL IS001230
C*      .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, IS001240
C*      CRDUNT,PRUNT,RANDIO IS001250
C*      COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP IS001260
CSEND DIMENSION KVAR(11500) IS001270
KVARDM = 11500 IS001280
DIMENSION ARRAY(1) IS001290
DIMENSION COVAR(465) IS001300
DIMENSION NN(60) IS001310
DATA SYMDA /* /*,SYMDB /* /* IS001320
MAXPOP=62 IS001330
MXFET1=30 IS001340
IBEGIN=DRUMAD IS001350
C* RESERVE ENOUGH DRUM STORAGE FOR MAXIMUM INITIAL MEANS IS001360
C* BEGIN3=IBEGIN + MAXPOP*MXFET1 + MXFET1 + 2 IS001370
C* CALL SETUP TO READ CARD INPUT AND INITIALIZE DEFAULT VALUES IS001380
C* ITIME=1 IS001390
C* NOCLS = 0 IS001400
C* TOTFLD = 0 IS001410
C* TOTVRT = 0 IS001420
C* TOTSUB = 0 IS001430
C* CORBAS=1 IS001440
C* ITRIGU = 0 IS001450
C* ITRIGO=0 IS001460
C* SYMDO = SYMDA IS001470
C* SYMDU = SYMDB IS001480
C* MEANDU = 0 IS001490
C* MEANDU = 255 IS001500
1 CALL SETUP7(ARRAY(CORBAS),TOP,ITIME) IS001510
IDUM = MAXCLS IS001520
IF(ITIME.GT.1)GO TO 2 IS001530
VARSIZ=NOFEAT*(NOFEAT+1)/2 IS001540
BEGIN1 = BEGIN3 + NCLASS*MAXPOP*(VARSIZ + NOFEAT + 1) IS001550
NWDS=DRMWDS-(BEGIN1-DRUMAD) IS001560
2 ITIME=ITIME+1 IS001570
NOCL=0 IS001580
C* CALL RDDATA TO COORDINATE READING OF DATA IS001590
C* 5 MAXDIM = TOP-CORBAS IS001600
FD1=CORBAS IS001610
CALL RDDATA(ARRAY(FD1),MAXDIM,KVAR,KVARDM,LAST) IS001620
MAXCLS = IDUM + NDOU IS001630
WRITE(6,210) NDOU(1),NDOU(2) IS001640
210 FORMAT(1X,/' DO/OU CLUSTER POP FOR THIS CLASS ',217) IS001650
BEGIN2=BEGIN1 + TOTWRD IS001660
N1 = FD1 + IPT IS001670
MEANS1=N1 + MAXCLS IS001680
STDEV1=MEANS1 + MAXCLS*NOFEAT IS001690
ITOP = STDEV1 + MAXCLS*NOFEAT IS001700
MAXDIM=TOP-ITOP IS001710

```

FILE ISOCLS

```

NOPTS = MAXDIM/(NOFEAT+1)
IDAT1 = TTOP
IF (NSDOTS.EQ.0) GO TO 4
DOTDMF = NOCHAN
TYPST = 1
CALL RDOTS (ARRAY (MEANS1), STDOTS, NSDOTS,
* TYPST, DOTDMF, DOTDMC, DOTDUM, COVAR,
* NOCHAN, CHNVC, DTD, COVAR,
* DOTDM, DOTDM, DOTDM, DOTDM, DOTDM, DOTDM, KVAR)
LNCAT = NSDOTS
DO 500 I = 1, NSDOTS
DO 500 K = 1, NOFEAT
III = (I-1)*NOFEAT + K
II = III + MEANS1 - 1
500 ARRAY (II) = KVAR (III)
IF (NOCHAN.EQ.NOFEAT) GO TO 8
WRITE (6, 110)
110 FORMAT (1H, 'NO CHANNELS FOR STARTING NOT EQUAL THAT FOR CLUSTER')
GO TO 9
4 CONTINUE
IF (ISOKEY.EQ.1) GO TO 7
C* SURVEC-SUBCLASSES FROM STATISTICS FILE FOR INTIAL MEANS.
C* NOSUB2-NUMBER OF INITIAL MEANS.
C* CHNVC-NUMBER OF CHANNELS FROM STATISTICS FILE. NOCHAN MUST EQUAL
IF (INITM.EQ.1) GO TO 6
LNCAT=1
GO TO 8
6 LNCAT=NOSUB2
CALL GFST (INUNIT, INFILE, ARRAY (MEANS1), DUM, NOSUB2, SUBVEC, NOCHAN
* ,CHNVC, ARRAY (TTOP), COVAR, 0)
LNCAT = NOSUB2
GO TO 8
7 CONTINUE
IF (ISOKEY.EQ.1) CALL RDFILE (ARRAY (MEANS1), ARRAY (TTOP))
8 CONTINUE
IF (NOPTS.GT.0) GOTO 10
WRITE (6, 100) MAXDIM
100 FORMAT (' DIMENSION LIMITS EXCEEDED IN ISOCLS BY', I6,
* ' REDUCE CHANNELS OR MAX. CLUSTERS')
9 CALL CMERR
10 CONTINUE
IRD=TOTPTS/NOPTS
IF (MOD (TOTPTS, NOPTS).EQ.0) GO TO 20
KPTS=MOD (TOTPTS, NOPTS)
IRD=IRD+1
IF (IRD.EQ.1) NOPTS=KPTS
GO TO 25
20 KPTS=NOPTS
25 CONTINUE
C* CALL ISODAT TO PERFORM CLUSTERING
C*
A1=1
A2=A1+ MAXCLS*NOFEAT
CLD1=A2 + MAXCLS*NOFEAT
KPLCE = NOPTS*NOFEAT + IDAT1
CALL ISODAT (ARRAY (IDAT1), ARRAY (KPLCE), ARRAY (MEANS1), ARRAY (N1),
* ARRAY (STDEV1), KVAR (CLD1), ARRAY (FD1), KVAR (A1),
* KVAR (A2))
C*
C* CHAIN CLUSTERS WHOSE DISTANCES ARE LESS THAN DLMIN
C*
LNCAT=LNCAT+DODU
IF (ICHN.GT.0) CALL CHAIN (KVAR (CLD1))
C*
C* PRINT FINAL RESULTS
C*
CALL PRINT (-1, ARRAY (KPLCE), ARRAY (MEANS1), ARRAY (STDEV1),
* KVAR (CLD1), ARRAY (FD1), ARRAY (N1))
C*
C* CREATE MAP OUTPUT TAPE FOR PMIS DAS IF DESIRED
C*
IF (MAPFMT.GT.0) CALL DSTAPE (ARRAY (KPLCE), KVAR (1), ARRAY (MEANS1),
* ARRAY (FD1))
LNCAT=LNCAT-DODU
C*

```

15001790  
15001800  
15001810  
15001820  
15001830  
15001840  
15001850  
15001860  
15001870  
15001880  
15001890  
15001900  
15001910  
15001920  
15001930  
15001940  
15001950  
15001960  
15001970  
15001980  
15001990  
15002000  
15002010  
15002020  
15002030  
15002040  
15002050  
15002060  
15002070  
15002080  
15002090  
15002100  
15002110  
15002120  
15002130  
15002140  
15002150  
15002160  
15002170  
15002180  
15002190  
15002200  
15002210  
15002220  
15002230  
15002240  
15002250  
15002260  
15002270  
15002280  
15002290  
15002300  
15002310  
15002320  
15002330  
15002340  
15002350  
15002360  
15002370  
15002380  
15002390  
15002400  
15002410  
15002420  
15002430  
15002440  
15002450  
15002460  
15002470  
15002480  
15002490  
15002500  
15002510  
15002520  
15002530  
15002540

FILE ISOCLS

```

C*   CALCULATE COVARIANCE MATRIX FOR EACH CLUSTER
C*   IF (VARSIZ*LNCAT.GT.KVARDM)GO TO 30
      CALL COVAR1(KVAR,ARRAY(IDAT1),ARRAY(KPLCE),ARRAY(MEANS1),
      *   ARRAY(N1),IBAD)
C
C*   CHECK FOR AT LEAST ONE SUBCLASS DELETED FOR SINGULAR MATRIX
C*   IF (IBAD.NE.0)STOP=0
      IF (IBAD.NE.0)GO TO 25
C*
C*   DO 26 II=1,LNCAT
26  NN(TOTSUB+II) = ARRAY(N1+II-1)
      TOTSUB = TOTSUB + LNCA
      NOCLS = NOCLS + 1
      TOTFLD = TOTFLD + NOFLD
      TOTVRT = TOTVRT + NVRT
      ARRAY(FD1+1)=IPT + FD1
      ARRAY(FD1+2)=LNCAT
      ARRAY(FD1+3)=NOFLD
C*
C*   WRITE STATS FOR THESE CLUSTERS ON SCRATCH FILE 18
C*   IF (NOCLS.EQ.1) ADRES=BEGIN3
      IN=NOFEAT*LNCAT
      CALL RWRITE(ADRES,ARRAY(MEANS1),IN,JSTAT)
      ADRES=ADRES+IN
      IN=VARSIZ*LNCAT
      CALL RWRITE(ADRES,KVAR,IN,LSTAT)
      ADRES=ADRES+IN
C*   60 IF (LSTAT.EQ.1) GO TO 60
C*
C*   GO READ IN ANOTHER CLASS
C*   CORBAS=CORBAS+IPT
      IF (LAST.NE.1)GO TO 5
      IF (NOCLS.LT.NCLASS)GO TO 1
C*
C*   NOW READ SCRATCH FILE AND STORE ON SAVTAP FILE AND PUNCH ON
C*   CARDS IF REQUESTED.
C*
      FLD1 = 1
      VERTX1 = FLD1 + TOTFLD*4
      CLSNM1 = VERTX1 + TOTVRT*2
      NOSUB1 = CLSNM1 + NOCLS
      SUBNM1 = NOSUB1 + NOCLS
C
C*   RETRIEVE INFORMATION FROM 'ARRAY'
C*   CALL GETINF(ARRAY(1),KVAR(FLD1),KVAR(VERTX1),KVAR(CLSNM1),
      *   KVAR(NOSUB1),KVAR(SUBNM1),NOCLS,TOTSUB)
C
C*   SWITCH = 1
C*
C*   OUTPUT STATS
C*
      CALL LARMAN(SAVTAP,STAFIL,NOCLS,TOTSUB,NOFEAT,TOTFLD,TOTVRT,
      *   FETVEC,KVAR(FLD1),KVAR(VERTX1),KVAR(CLSNM1),KVAR(NOSUB1),
      *   KVAR(SUBNM1),NN,REGIN3,VARSIZ,PUNCH,DUMMY,STATKY,SWTCH)
      RETURN
30  KV=KVARDM
      WRITE(6,200)KV
      CALL CMERR
200  FORMAT(' DIMENSION LIMIT OF ',16,' FOR COVARIANCES EXCEEDED')
      RETURN
      END

```

IS002550  
 IS002560  
 IS002570  
 IS002580  
 IS002590  
 IS002600  
 IS002610  
 IS002620  
 IS002630  
 IS002640  
 IS002650  
 IS002660  
 IS002670  
 IS002680  
 IS002690  
 IS002700  
 IS002710  
 IS002720  
 IS002730  
 IS002740  
 IS002750  
 IS002760  
 IS002770  
 IS002780  
 IS002790  
 IS002800  
 IS002810  
 IS002820  
 IS002830  
 IS002840  
 IS002850  
 IS002860  
 IS002870  
 IS002880  
 IS002890  
 IS002900  
 IS002910  
 IS002920  
 IS002930  
 IS002940  
 IS002950  
 IS002960  
 IS002970  
 IS002980  
 IS002990  
 IS003000  
 IS003010  
 IS003020  
 IS003030  
 IS003040  
 IS003050  
 IS003060  
 IS003070  
 IS003080  
 IS003090  
 IS003100  
 IS003110  
 IS003120  
 IS003130  
 IS003140  
 IS003150  
 IS003160  
 IS003170  
 IS003180  
 IS003190  
 IS003200  
 IS003210





FILE: COVAR1

```
50 CONTINUE
   IACEPT=PMIN+NOFEAT
   IF(IACEPT.LT.NOFEAT)GO TO 58
C
C
C   CHECK FOR SINGULAR COVARIANCE MATRIX
   DO 51 I=1,LNCAT
   CALL CHLODET(COVAR(1,I),NOFEAT,DUMM,DET)
   IF(DET.LT.TOL)GO TO 52
51 CONTINUE
   GO TO 58
C
C
C   DELETE SINGULAR COVARIANCE MATRIX CLUSTER
52 WRITE(6,160)I
   IF(LNCAT.EQ.1)CALL CMERR
   IRAD=1
   LNCAT=LNCAT-1
   LLNCAT=LLNCAT-1
   DO 53 II=1,LNCAT
   DO 53 III=1,NOFEAT
   MFANS(III,II)=MEANS(III,II+1)
53 CONTINUE
58 RETURN
160 FORMAT(2X,'CLUSTER',I5,' DELETED FOR SINGULARITY')
C
55 IF(STATKY.NE.1)RETURN
   WRITE(6,HEAD)
   WRITE(6,150)CLSNAM
   DO 80 I=1,LNCAT
   WRITE(6,90)I
   DO 70 LOC=1,NOFEAT,12
   ISTOP=LOC+11
   IF(ISTOP.GT.NOFEAT)ISTOP=NOFEAT
   WRITE(6,140)(CH,FETVEC(J),J=LOC,ISTOP)
   II=1
   KINC=1
   DO 60 J=LOC,NOFEAT
   K=J*(J+1)/2-II+1
   JK=K+KINC-1
   WRITE(6,100)(COVAR(M,I),M=K,JK)
   II=II+1
60 IF(KINC.LT.ISTOP.AND.KINC.LT.12)KINC=KINC+1
   WRITE(6,110)
70 CONTINUE
80 CONTINUE
   RETURN
90 FORMAT(///' COVARIANCE MATRIX FOR CLUSTER',I4/)
100 FORMAT(/6X,12F9.2)
110 FORMAT(///)
120 FORMAT(1H1)
140 FORMAT(9X,12(A3,I2,' '),3X)
150 FORMAT(/' COVARIANCES FOR CLASS',2X,A4//)
   END
```

COV00770  
COV00780  
COV00790  
COV00800  
COV00810  
COV00820  
COV00830  
COV00840  
COV00850  
COV00860  
COV00870  
COV00880  
COV00890  
COV00900  
COV00910  
COV00920  
COV00930  
COV00940  
COV00950  
COV00960  
COV00970  
COV00980  
COV00990  
COV01000  
COV01010  
COV01020  
COV01030  
COV01040  
COV01050  
COV01060  
COV01070  
COV01080  
COV01090  
COV01100  
COV01110  
COV01120  
COV01130  
COV01140  
COV01150  
COV01160  
COV01170  
COV01180  
COV01190  
COV01200  
COV01210  
COV01220  
COV01230  
COV01240  
COV01250  
COV01260  
COV01270  
COV01280  
COV01290

FILE: ISODAT

```

SUBROUTINE ISODAT(C,IPLACE,MEANS,N,STDEV,CLD,FLDINF,AVP,AMN)
IMPLICIT INTEGER (A-Z)
IMPLICIT INTEGER (A-Z)
INCLUDE COMAK5,LIST
INCLUDE COMBK6,LIST
C
CMS360
CMS360
C
INCLUDE CMK16,LIST
COMMON/PASS/STOP,LNCAT,NMIN,KRN,STDMAX,DLMIN,SEP,
MAP,SPTRIG,IRD,KPTS,NOPTS,PUNCH,
* ICHN,CHNTHS,ICHAIN(62),NWDS,IREGIN,REGIN1,
REGIN2,BEGIN3,CLSNAM,NOFLD,IPT,TOTWRD,TOTPTS,
NCLASS,NOCLS,TOTSUR,TOTFLD,TOIVRT,NOCL,NVRT
* ,NXTCLS,NOFEAT,MAXCLS,FETVEC(30),SYMMTX(62)
* ,VAPSI7,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
* ,IOPDR,INUNTI,INFILE,INITM,PMIN,SUBVEC(62),NOSUB2,CHNVC(30)
* ,NOCHAN,FRCOMP,NOSEQ,MEANDO,MEANDU,
* SYMDO,SYMDO1,ITRIG0,ITRIGU,DOFLAG,
* DUFLAG,DODU,STDOTS(60),NSDOTS,SUNCOR(30),LLNCAT,
* DVERT(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2)
* ,MXFFT1,MXPPOP
REAL SUNCOR
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWDS,PAGSI7,DATFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP
CSEND
EQUIVALENCE (SGMIN,STDMAX)
REAL MEANS,STDEV,STDMAX,SEP,AVP,C,AMN,SGMA,RND,
* TEST,DMIN,DLMIN,CLD,TIME,PERCEN,DIJ
REAL ESUM,ESOT,MEAN(30,62),SDIJ
LOGICAL DEL
DIMENSION AVP(NOFFEAT,MAXCLS),ISGMA(62)
DIMENSION C(NOFFEAT,NOPTS),IPLACE(NOPTS)
DIMENSION AMN(NOFFEAT,MAXCLS),SGMA(62)
DIMENSION MEANS(NOFFEAT,MAXCLS),N(MAXCLS)
DIMENSION STDEV(NOFFEAT,MAXCLS),CLD(MAXCLS,MAXCLS)
DIMENSION FLDINF(1)
REAL SQJM
DIMENSION PTR(62)
DATA SS/'S',CC/'C'/
EQUIVALENCE (KDIM,NOFEAT),(LNCAT,INCAT)
DEL=.FALSE.
ISFQ=0
MAXCL=MAXCLS-DODU
IDUM=LNCAT+DODU-MAXCLS
IF (IDUM.GT.0) LNCAT=LNCAT-IDUM
ISTOP=STOP
SPLFIN=0
KKT=1
DO 5 I=1,30
SUNCOR(I)=1.
5 IF (ISUNC.NE.0.OR.ISUNT.NE.0) CALL SUNFAC(SUNCOR,SUNANG,
* FETVEC,NOFEAT,ISUNC,ISUNT)
C*
LK=K
C* ASSIGN DATA TO CLUSTERS
C*
10 CONTINUE
LNCAT=LNCAT+DODU
IF (DOFLAG.EQ.0) GO TO 12
DO 11 J=1,NOFEAT
MEANS(J,LNCAT+1)=MEANDO
11 IF (DUFLAG.EQ.0) GO TO 14
DO 13 J=1,NOFEAT
MEANS(J,LNCAT)=MEANDU
13 CONTINUE
DO 15 K=1,LLNCAT
DO 15 J=1,NOFEAT
15 MEAN(J,K)=MEANS(J,K)
IF (LNCAT.LE.1).AND.KKT.GT.1) GO TO 530
CALL PSPLIT(MEANS,STDEV,N,CLD,C,IPLACE,AVP,AMN,MEANS)
CALL CLOCK(TIME)
IF (MOD(KKT,KRN).EQ.0) WRITE(6,120)KKT,TIME
120 FORMAT(' CUMULATIVE TIME AFTER ASSIGNING DATA TO CLUSTERS FOR ITER
* ATION',I4,' IS',F10.6)
IS000010
IS000020
IS000030
IS000040
IS000050
IS000060
IS000070
IS000080
IS000090
IS000100
IS000110
IS000120
IS000130
IS000140
IS000150
IS000160
IS000170
IS000180
IS000190
IS000200
IS000210
IS000220
IS000230
IS000240
IS000250
IS000260
IS000270
IS000280
IS000290
IS000300
IS000310
IS000320
IS000330
IS000340
IS000350
IS000360
IS000370
IS000380
IS000390
IS000400
IS000410
IS000420
IS000430
IS000440
IS000450
IS000460
IS000470
IS000480
IS000490
IS000500
IS000510
IS000520
IS000530
IS000540
IS000550
IS000560
IS000570
IS000580
IS000590
IS000600
IS000610
IS000620
IS000630
IS000640
IS000650
IS000660
IS000670
IS000680
IS000690
IS000700
IS000710
IS000720
IS000730
IS000740
IS000750
IS000760
IS000770
IS000780
IS000790
```

FILE: ISODAT

```
IF (EPCOMP.NE.1) GO TO 135
ESUM=0.0
DO 132 J=1,NOFEAT
DO 132 K=1,LNCAT
FSUM=FSUM+N(K)*(STDEV(J,K))**2/TOTPTS
132 CONTINUE
ESQT=SQRT(ESUM/NOFEAT)
WRITE(6,133) ESQT,PERCEN,STDMAX
133 FORMAT(1X,///' ERCOMP= ',F7.3,' PERCEN = ',F5.3,' STDMAX = ',
*F7.3/)
C*
C* CALCULATE DISTANCES BETWEEN CLUSTER CENTERS
C*
135 CALL CLDIST(CLD,STDEV,MEANS)
C*
C* IF STOP EQUALS ZERO DELETE SMALL CLUSTERS
C*
LNCAT=LLNCAT
IF (MOD(KKT,MAP)) 150,140,150
140 CALL PRINT(KKT,IPLACE,MEANS,STDEV,CLD,FLDINF,N)
GO TO 161
150 IF (MOD(KKT,KRN)) 161,160,161
160 CONTINUE
CALL PRINT(KKT,IPLACE,MEANS,STDEV,CLD,FLDINF,N)
161 CONTINUE
LNCAT=LLNCAT-DODU
IF (ISTOP.EQ.0) GO TO 162
C
C FOR ITERATION N CHECK N(K) AGAINST PMIN * NOFEAT
C
IF (ISEQ.NE.NOSEQ) GO TO 169
ISTOP = 0
DO 163 K = 1,LNCAT
IF (N(K) - (PMIN * NOFEAT)) 167,163,163
163 CONTINUE
IF (.NOT.DEL) RETURN
DO 164 KK=1,LLNCAT
DO 164 KKK=1,NOFEAT
164 MEANS(KKK,KK) = MEAN(KKK,KK)
CALL PSPLIT(MEANS,STDEV,N,CLD,C,IPLACE,AVP,AMN,MEANS)
DO 165 KK=1,LLNCAT
DO 165 KKK=1,NOFEAT
165 MEAN(KKK,KK) = MEANS(KKK,KK)
CALL CLDIST(CLD,STDEV,MEANS)
RETURN
167 WRITE(6,168)K,N(K),PMIN,NOFEAT
168 FORMAT(/' CLUSTER ',I3,' REMOVED FOR HAVING ONLY ',I6,' POINTS. '/
*' MIN. POINTS IS (' ,I4,' * ',I2,' )')
RETF=1
LK=K
GO TO 570
171 K=LK
DEL = .TRUE.
GO TO 162
169 CONTINUE
170 CONTINUE
C
C ON ITERATIONS 1 THRU N-1 CHECK N(K) AGAINST NMIN
C
DO 180 K=1,INCAT
IF (N(K)-NMIN) 190,180,180
180 CONTINUE
IF (DEL) CALL CLDIST(CLD,STDEV,MEANS)
GO TO 220
190 IF (MOD(KKT,KRN)) 200,195,200
195 WRITE(6,210)K,N(K),NMIN
200 RETF=2
LK=K
GO TO 570
201 K=LK
DEL = .TRUE.
GO TO 170
210 FORMAT('0 CLUSTER ',I2,' REMOVED FOR HAVING ONLY ',I6,
2 ' ELEMENTS. MIN. NO. ELEMENTS IS ',I6)
220 CONTINUE
C
C
C* SPLIT ITERATION
```

FILE: ISODAT

```
C
C
225 DO 225 I=1,INCAT
    PTR(I)=I
    ISPLT=0
    DO 260 K=1,INCAT
C*
C* FIND MAXIMUM STANDARD DEVIATION PER CLUSTER
C*
    SGMA(K) = 0.
    DO 250 J = 1,NOFEAT
    SDUM = STDEV(J,K)*SUNCOR(J)
    IF (SDUM - SGMA(K)) 250,240,240
240   ISGMA(K) = J
    SGMA(K) = SDUM
250 CONTINUE
    IF (SGMA(K).GE.STDMAX) ISPLT=ISPLT+1
260 CONTINUE
    IF (2*LNCAT.GT.MAXCL) CALDESCEN(SGMA,LNCAT,ISGMA,PTR)
    TEST=FLOAT(ISPLT)/FLOAT(LNCAT)
    IF (TEST.LE.PERCEN) SPLFIN=1
    IF (KKT.GT. ISTOP) SPLFIN = 1
    IF (SPLFIN.EQ. 0) GO TO 270
    IF (MOD(KKT,KRN) .EQ. 0) WRITE(6,503)
503 FORMAT( /)
    IF (MOD(KKT,KRN).EQ.0)WRITE(6,502)
502 FORMAT(1X,'USER INPUT*SPLIT-COMBINE SEQUENCE OF ITERATIONS')
    ISFO=ISEQ+1
    IF (SEQUEN(ISFO).EQ.SS) GO TO 270
    IF (SEQUFN(ISEQ).EQ.CC) GO TO 410
C
C IS SPLITTING REQUIRED
C
270 K=1
    NCAT=INCAT
280 IF (K-NCAT) 290,290,500
290 IF (STDMAX-SGMA(K)) 300,300,310
300 IF (N(K)-(NMIN*NMIN+2)) 310,310,320
310 K=K+1
    GO TO 280
C
C SPLIT CLUSTER K
C
320 TRIG1=1
    DEL=.TRUE.
    KX=ISGMA(K)
330 INCAT=INCAT+1
    LLNCAT=LLNCAT+1
    IF (LLNCAT.LE.MAXCLS) GO TO 350
    IF (MOD(KKT,KRN).EQ.0)WRITE(6,340)KKT
340 FORMAT(/' MAXIMUM CLUSTERS ON ITERATION',I4/' SPLITTING REQUIRED B
    *UT NOT PERFORMED'//)
    LNCAT = MAXCL
    LLNCAT=MAXCLS
    GO TO 500
350 INC=INCAT
    LL=PTR(K)
360 DO 370 I=1,KDIM
370   AMN(I,INC)=AMN(I,LL)
380   AMN(KX,LL)=AMN(KX,LL)+SEP*SGMA(K)
    AMN(KX,INC)=AMN(KX,INC)-SEP*SGMA(K)
    SGMA(K)=0.0
    IF (MOD(KKT,KRN)) 400,401,400
401 WRITE(6,390)LL,KX,INC
390 FORMAT('0 CLUSTER ',I2,' IS SPLIT IN THE ',I2,'TH PARAMETER INTO C
    LUSTER ',I2)
400 CONTINUE
    K=K+1
    GO TO 280
C
C EVEN ITERATION
C
C ARE CLUSTERS TO BE COMBINED
C
410 CONTINUE
    DO 405 L=1,LNCAT
405   PTR(L)=I
C
C NOCOMB=0
```

FILE: ISODAT

```
      NOCLST=LNCAT-1
      L=-1
406  L=L+2
      IF (L.GT.NOCLST) GO TO 480
      NOCLTR = LNCAT - 1
      KK=0
      DMIN=DMIN
      DO 430 I=1,NOCLTR
C
      IF (PTR(I).EQ.0) GO TO 430
      II=I+1
      DO 425 J=II,LNCAT
      IF (PTR(J).EQ.0) GO TO 425
      SDIJ = 0.0
      DO 420 JJ=1,KDIM
      SDIJ=SDIJ+((AMN(JJ,I)-AMN(JJ,J))*2/(STDEV(JJ,I)*STDEV(JJ,J)))
420  CONTINUE
      DIJ=SQRT(SDIJ)
C
      IF (DIJ.GT.DMIN) GO TO 425
      DMIN=DIJ
      KKK=I
      KKK=J
425  CONTINUE
430  CONTINUE
C
      IF (KK.EQ.0) GO TO 480
      PTR(KK)=0
C
      COMBINE CLUSTERS KK AND KKK
C
      DEL=.TRUE.
      RND=1.0 /FLOAT(N(KK)+N(KKK))
C
      DO 460 K=1,KDIM
      AMN(K,KK)=(N(KK)*AMN(K,KK)+N(KKK)*AMN(K,KKK))*RND
460  CONTINUE
C
      RETF=3
      LK=KKK
      GO TO 570
461  KKK=LK
      IF (KKK.EQ.(LNCAT+1)) GO TO 435
C
      MOVE POINTERS UP
C
      DO 175 K=KKK,LNCAT
      PTR(K) = PTR(K+1)
175  CONTINUE
C
      IF (MOD(KKT,KRN))440,441,440
441  WRITE (6,490)KK,KKK,KK
440  IF (L.LT.NOCLST) GO TO 406
C
490  CONTINUE
490  FORMAT(' CLUSTERS ',I2,' AND ',I2,' HAVE BEEN COMBINED INTO CLUST
      ZER ',I2)
C*
C*  REINITIALIZE
C*
500  CONTINUE
      DO 510 J=1,MAXCLS
      SGMA(J)=0.0
      ISGMA(J)=0
      DO 510 K=1,KDIM
      AVP(K,J)=0.0
      STDEV(K,J)=0.0
      MFANS(K,J)=AMN(K,J)
      AMN(K,J)=0.0
510  CONTINUE
      KKT=KKT+1
      OFL=.FALSE.
      GO TO 10
C
530  IF (KKT.NF.2) GO TO 550
      WRITE (6,440)
540  FORMAT(' THE ORIGINAL CLUSTER WAS NOT SPLIT - EXAMINE THE INPUT VA
      *LINE FOR STDMAX')
      KKT=1
      ISTOP=0
      GO TO 10
```

IS002380  
IS002390  
IS002400  
IS002410  
IS002420  
IS002430  
IS002440  
IS002450  
IS002460  
IS002470  
IS002480  
IS002490  
IS002500  
IS002510  
IS002520  
IS002530  
IS002540  
IS002550  
IS002560  
IS002570  
IS002580  
IS002590  
IS002600  
IS002610  
IS002620  
IS002630  
IS002640  
IS002650  
IS002660  
IS002670  
IS002680  
IS002690  
IS002700  
IS002710  
IS002720  
IS002730  
IS002740  
IS002750  
IS002760  
IS002770  
IS002780  
IS002790  
IS002800  
IS002810  
IS002820  
IS002830  
IS002840  
IS002850  
IS002860  
IS002870  
IS002880  
IS002890  
IS002900  
IS002910  
IS002920  
IS002930  
IS002940  
IS002950  
IS002960  
IS002970  
IS002980  
IS002990  
IS003000  
IS003010  
IS003020  
IS003030  
IS003040  
IS003050  
IS003060  
IS003070  
IS003080  
IS003090  
IS003100  
IS003110  
IS003120  
IS003130  
IS003140  
IS003150  
IS003160

FILE: ISODAT

```
550 WRITE (6,560)KKT
560 FORMAT(///,AFTER '.I4.', ITERATIONS ALL DATA HAS BEEN ASSIGNED TO 0
      *NE CLUSTER:/)
      KKT=1
      ISTOP=0
      GO TO 10
570 CONTINUE
C*
C* ROUTINE TO DELETE A CLUSTER
      INCAT=INCAT-1
      LLNCAT=LLNCAT-1
      IF (LK.EQ.(INCAT+1).AND.DODU.EQ.0) GO TO (171,201,461),RETF
      DO 561 J=LK,LLNCAT
      DO 552 L=1,KDIM
      AMN(L,J)=AMN(L,J+1)
      MEANS(L,J)=MEANS(L,J+1)
      MEAN(L,J)=MEAN(L,J+1)
552 STDEV(L,J)=STDEV(L,J+1)
      N(J)=N(J+1)
561 CONTINUE
      GO TO (171,201,461),RETF
      END
```

5003170  
5003180  
5003190  
5003200  
5003210  
5003220  
5003230  
5003240  
5003250  
5003260  
5003270  
5003280  
5003290  
5003300  
5003310  
5003320  
5003330  
5003340  
5003350  
5003360  
5003370  
5003380  
5003390







FILE: R0DATA

```
C* THIS SUBROUTINE COORDINATES THE ROUTINES TO READ FIELDS OF DATA FROM THE IMAGE TAPE AND STORE IT ON A DRUM FILE FOR THE ISOCLS ROUTINES.
C*
C* SUBROUTINE R0DATA (ARRAY, TOP, IDATA, IDIM, LAST)
C* IMPLICIT INTEGER (A-Z)
C* DIMENSION ARRAY (TOP), FLINDEF (6), IDATA (IDIM), FL (12), LSTAT (3)
C* INCLUDE COMMONS.LIST
C* INCLUDE COMMONS.LIST
C* COMMON /PASS/STOP, LNCAT, NMIN, KRN, STDMAX, DLMIN, SEP,
C* MAP, SPTIG, IRU, KPTS, NOPTS, PUNCH,
C* ICHN, CHNTHS, ICHAIN (62), NPOS, I-REGIN, REGINI,
C* REGIN2, REGIN3, CLSNUM, NOFLD, IPT, TOTWRD, TOTPTS,
C* NCLASS, NOCLS, TOTSUM, TOTFLD, TOTVRT, NOCL, NVRT
C* NITCLS, NOFEAT, MAXCLS, FEATV (30), SYMTRA (62)
C* VANSIZ, STATKY, ISOKEY, MAPENT, MAPKEY, SEQLEN (20), PERCEN, SIMERP
C* IOWFR, IOWITT, INFILE, INITM, PMIN, SUBVEC (62), NOSUB2, CMNVC (30)
C* NOCHAI, FRCONP, NOSEQ, MEAND, MEANDU,
C* SYMCL, SYMCL, ITRIG, ITRIGU, NOFLAG,
C* DIFLAG, DDDU, STDOTS (60), NSDOTS, SINCOR (30), LLNCAT,
C* DVERT (25, 2), DPECT (60, 2), DVPT (11, 2), IDCNT (2), NDOU (2)
C* MAXFT1, MAXPOP
C* LOCAL SINCOR
C* COMMON /GLOBAL/HEAD (63), MAPTAP, DATAPE, SAVTAP, HMFILE, HMKEY,
C* HISFIL, HISKEY, TRFORM, EXPTP, EMPKEY, MAPUNT, NOFILE,
C* DRIMAD, DRWDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
C* MSHSTN, MSHSTJ, SCTRUN, MAPFIL
C* DOTINT, DOTFIL, NCHPAS, TRNSFL, HMTREL, MISTFL, PCHUNT,
C* CROUT, PRINT, RANDIO
C* SEND
C* PRINT = 30
C* DIMENSION CARD (20)
C* EQUIVALENCE (FLINDEF (1), LINSTR), (FLINDEF (4), SAMSTR),
C* (FLINDEF (2), LINEND), (FLINDEF (5), SAMEND),
C* (FLINDEF (3), LININC), (FLINDEF (6), SAMINC)
C* DATA LPWV (1)
C* DIMENSION L (2), LDOU (11, 2), ICH (35), ICH (35), MDINT (11, 2),
C* DINT (12, 2), DPINT (12, 2), DIN (70)
C* DATA DNAME /'OHE' /
C* DATA DNAME /'UNID' /
C* DIMENSION FLOSAV (4, 10), VERTEX (220)
C* RESERVE 2000 LOCATIONS OF 'ARRAY' FOR FIELD DEFINITION INFORMATION.
C* THE REMAINDER OF 'ARRAY' IS USED FOR I/O BUFFERS.
C* CLASS AND FIELD INFORMATION STORED AS FOLLOWS
C*
C* ARRAY (1) = CLASS NAME
C* ARRAY (2) = RESERVED FOR INDEX POINTER TO NEXT CLASS NAME
C* ARRAY (3) = RESERVED FOR NO. OF CLUSTERS IN THIS CLASS
C* ARRAY (4) = NO. OF FIELDS FOR THIS CLASS
C* ARRAY (5) = FIRST FIELD NAME FOR THIS CLASS
C* (6) = NO. OF VERTICES FOR THIS FIELD (NV)
C* (7) - (7+NV*2) = ACTUAL VERTEX NUMBERS
C* (8+NV*2) = TOTAL PIXELS IN THIS FIELD
C* (9+NV*2) - (10+NV*2) = FLDINFL BLOCK FOR THIS FIELD
C* CALL TAPHDR (DATAPE, DATFIL)
1 CONTINUE
RESERV=2000
ADDRS=REGINI
IWRD=1
IWRD=0
NVRT=0
LAST=0
TOTWRD=0
IDP = 0
DOFLAG = 0
DIFLAG = 0
DDDU = 0
NDOU (1) = 0
NDOU (2) = 0
2 REINDX=-RESERV+1
NRUES=3
MAXDIM=TOP-RESERV
HIESIZ=MAXDIM*(NRUES+DOFEAT) * NOFEAT
IF (HIESIZ.GT. 100) GO TO 3
RESERV=RESERV-100
IF (RESERV.GT. 30) GO TO 2
R0D00010
R0D00020
R0D00030
R0D00040
R0D00050
R0D00060
R0D00070
R0D00080
R0D00090
R0D00100
R0D00110
R0D00120
R0D00130
R0D00140
R0D00150
R0D00160
R0D00170
R0D00180
R0D00190
R0D00200
R0D00210
R0D00220
R0D00230
R0D00240
R0D00250
R0D00260
R0D00270
R0D00280
R0D00290
R0D00300
R0D00310
R0D00320
R0D00330
R0D00340
R0D00350
R0D00360
R0D00370
R0D00380
R0D00390
R0D00400
R0D00410
R0D00420
R0D00430
R0D00440
R0D00450
R0D00460
R0D00470
R0D00480
R0D00490
R0D00500
R0D00510
R0D00520
R0D00530
R0D00540
R0D00550
R0D00560
R0D00570
R0D00580
R0D00590
R0D00600
R0D00610
R0D00620
R0D00630
R0D00640
R0D00650
R0D00660
R0D00670
R0D00680
R0D00690
R0D00700
R0D00710
R0D00720
R0D00730
R0D00740
R0D00750
R0D00760
R0D00770
R0D00780
R0D00790
```

FILE: RDDATA

```
GO TO 70
3 CONTINUE
NOFLD=0
IPT=1
TOTVT2=0
IF (NOCL.EQ.0) GO TO 5
4 ARRAY(IPT)=NXTCLS
IPT=IPT+4
WRITE(6,HEAD)
WRITE(6,500)NXTCLS
C*
C* READ A FIELD DESCRIPTION FROM CARDS.
C*
5 ICK = LARFAD(ARRAY(IPT),ARRAY(IPT+2),FLD1,ARRAY(IPT+1) )
IF (ICK.NE.-3) GO TO 1000
WRITE (6,143)
READ (RUNIT,150) (CARD(I), I=1,20)
WRITE (6,160) (CARD(I), I=1,20)
150 FORMAT(20A4)
160 FORMAT(1Y,20A4)
REWIND RUNIT
IDP=IDP+1
IDCNT(IDP)=0
DVPNT(1,IDP)=1
READ(30,100) DNAME
REWIND 30
IF (DNAME.EQ.DONAME) ITRIG0=1
IF (DNAME.EQ.DUNAME) ITRIGU=1
IF (DNAME.EQ.DONAME) IS=2
IF (DNAME.EQ.DUNAME) IS=1
INDV=1
INDP=1
GO TO 5
1000 IF (ICK.LE.0.OP.IDP.LE.0) GO TO 1030
IF (IDCNT(IDP).LT.10) GO TO 1025
WRITE(6,170)
170 FORMAT(// ' TOO MANY DO OR DU FIELDS THESE IGNORED')
GO TO 5
1025 CONTINUE
READ (RUNIT,150) (CARD(I), I=1,20)
WRITE (6,160) (CARD(I), I=1,20)
REWIND RUNIT
DVEPT(INDV,IDP) = ARRAY(IPT + 1)
IOLIM = DVEPT(INDV,IDP)*2
DO 1010 I=1,IOLIM
INDV=INDV+1
VERTEX(TOTVT2+I)=ARRAY(IPT+I+1)
1010 DVEPT(INDV,IDP) = ARRAY(IPT + I + 1)
INDV = INDV + 1
TOTVT2=TOTVT2+IOLIM
DO 1020 I=1,6
DVEPT(INDP,IDP)=FLDINF(I)
1020 INDP=INDP+1
IDCNT(IDP)=IDCNT(IDP)+1
IDUM=IDCNT(IDP)+1
DVPNT(IDUM,IDP)=INDV
GO TO 5
C FINISHED WITH DO/DU FIELD PROCESSING
C
1030 CONTINUE
IDP=0
IDPP=ITRIG0+ITRIGU
IF (ICK.LT.0) GO TO 20
IF (ICK.EQ.0) GO TO 30
IF (NOCL.GT.0) GO TO 6
WRITE (6,800)
CALL CMERR
6 CONTINUE
NV=ARRAY(IPT+1)
NVRT=NVRT+NV
NOFLD=NOFLD+1
NSAMP=(SAMEND-SAMSTR)/SAMINC+1
FLDSAM=0
IR=IPT+2
NQ=NV-1
NQ=NQ-5
IF (NQ.GT.5) NQ=5
IF (IR+NQ*2 = 1)
WRITE (6,600)NOFLD,ARRAY(IPT),SAMINC,LININC,
* (LPRN,ARRAY(1),ARRAY(1+1),I=IR,IE,2)
RDD00800
RDD00810
RDD00820
RDD00830
RDD00840
RDD00850
RDD00860
RDD00870
RDD00880
RDD00890
RDD00900
RDD00910
RDD00920
RDD00930
RDD00940
RDD00950
RDD00960
RDD00970
RDD00980
RDD00990
RDD01000
RDD01010
RDD01020
RDD01030
RDD01040
RDD01050
RDD01060
RDD01070
RDD01080
RDD01090
RDD01100
RDD01110
RDD01120
RDD01130
RDD01140
RDD01150
RDD01160
RDD01170
RDD01180
RDD01190
RDD01200
RDD01210
RDD01220
RDD01230
RDD01240
RDD01250
RDD01260
RDD01270
RDD01280
RDD01290
RDD01300
RDD01310
RDD01320
RDD01330
RDD01340
RDD01350
RDD01360
RDD01370
RDD01380
RDD01390
RDD01400
RDD01410
RDD01420
RDD01430
RDD01440
RDD01450
RDD01460
RDD01470
RDD01480
RDD01490
RDD01500
RDD01510
RDD01520
RDD01530
RDD01540
RDD01550
RDD01560
RDD01570
RDD01580
```

FILE: RDDATA

```
IF(NR.LE.0)GO TO 7
IR=IF+1
IF=IR+NR*2 - 1
WRITE (6,650) (LPRN,ARRAY(I),ARRAY(I+1),I=IB,IE,2)
7 CONTINUE
IF(NSAMP*NOFEAT.GT.IDIM)GO TO 90
C*
C* POSITION TAPE FOR THIS FIELD
C*
CALL FLDINT(FLDINF,FETVEC,NOFEAT)
FLDSAM=0
DO 10 LINE=LINSTR,LINEND,LININC
LND(1)=0
LND(2)=0
IDBR = 2
IDEE=1
IF (IDPP.EQ.0) GO TO 1095
C BOTH DO AND DU TRIGGERS OFF --- SKIP AROUND
DO 1040 I=1,IDPP
IDLIM=IDCNT(INO)
DO 1050 I=1,IDLIM
IDUM=(I-1)*5
LONSTR=IDRECT(IDUM+1,IND)
LOEND=IDRECT(IDUM+2,IND)
LOINC=IDRECT(IDUM+3,IND)
DO 1040 II = LONSTR,LOEND,LOINC
IF(II.NE.(LINE)) GO TO 1040
LND(IND)=LND(IND)+1
IDUM=LND(IND)
LDOU(IDUM,IND) = I
1040 CONTINUE
1050 CONTINUE
1050 CONTINUE
IF (LND(1).EQ.0.AND.LND(IDPP).EQ.0) GO TO 1095
C NO DO OR DU FOR THIS LINE
IF(LND(1).GT.0)IDBR=1
IF(IDPP.EQ.2.AND.LND(2).GT.0) IDEE=2
DO 1090 I=1,IDBR*IDEE
IDLIM=LND(IND)
IF (IDLIM.EQ.0) GO TO 1090
IDUM=0
DPINT(1,1) = 1
DPINT(1,2) = 1
DO 1080 I=1,IDLIM
IDF=LDOU(I,IND)
DVP=DVPNT(IDF,IND)
CALL FLDINT(DVERT(DVP+1,IND),DVERT(DVP,IND),FL,LINE,SAMPS,NI)
DPINT(I,IND)=NI
IF (NI.EQ.0) GO TO 1080
DO 1070 II=1,NI
1070 DPINT(II+IDUM,IND)=FL(II)
IDUM=IDUM+NI
DPINT(I+1,IND) = IDUM + 1
1080 CONTINUE
1090 CONTINUE
1095 CONTINUE
CALL LINEPD(IDATA,ENDTAP)
IF(ENDTAP.EQ.-1)GO TO 80
C*
C* FIND SAMPLE INTERSECTS FOR THIS LINE - NI=NO. OF INTERSECTS
C*
CALL FLDINT(ARRAY(IPT+2),NV,FL,LINE,SAMPS,NI)
C*
C* STORE DATA ON THIS LINE INTO OUTPUT BUFFER
C*
REAL RWRD
EQUVALNCF(0,0,0,ITGWRD)
MODSS=MOD(SAMSTR,SAMINC)
DO 60 I=1,NI,2
IR=(FL(I)-SAMSTR)/SAMINC+1
IF=(FL(I+1)-SAMSTR)/SAMINC+1
IF(MODSS.NE.MOD(FL(I),SAMINC)) IR=IR+1
IF(IR.GT.IE) GOTO 60
IF(IDPP.EQ.0) GOTO 2055
IF(LND(IDPP).EQ.0.AND.LND(IDEE).EQ.0) GOTO 2055
DO 2050 I=1,IDBR*IDEE
IDLIM=LND(IND)
IF (IDLIM.EQ.0) GOTO 2050
IOSIT=1
RDD01590
RDD01600
RDD01610
RDD01620
RDD01630
RDD01640
RDD01650
RDD01660
RDD01670
RDD01680
RDD01690
RDD01700
RDD01710
RDD01720
RDD01730
RDD01740
RDD01750
RDD01760
RDD01770
RDD01780
RDD01790
RDD01800
RDD01810
RDD01820
RDD01830
RDD01840
RDD01850
RDD01860
RDD01870
RDD01880
RDD01890
RDD01900
RDD01910
RDD01920
RDD01930
RDD01940
RDD01950
RDD01960
RDD01970
RDD01980
RDD01990
RDD02000
RDD02010
RDD02020
RDD02030
RDD02040
RDD02050
RDD02060
RDD02070
RDD02080
RDD02090
RDD02100
RDD02110
RDD02120
RDD02130
RDD02140
RDD02150
RDD02160
RDD02170
RDD02180
RDD02190
RDD02200
RDD02210
RDD02220
RDD02230
RDD02240
RDD02250
RDD02260
RDD02270
RDD02280
RDD02290
RDD02300
RDD02310
RDD02320
RDD02330
RDD02340
RDD02350
RDD02360
RDD02370
```

9-27  
68

FILE: RDDATA

```
MEANDD=MEANDD
IF (IDRR.EQ.IDFE) GOTO 2003
IF (IND.EQ.2) IDSIT=2
IF (IND.EQ.2) MEANDD=MEANDD
GOTO 2009
2003 IF (IDRR.EQ.1.AND.IDPP.EQ.2) GOTO 2009
IF (ITRIGU.EQ.0) GOTO 2009
IDSIT=2
MEANDD=MEANDD
2009 CONTINUE
DO 2040 K=1.IDLIM
NDIN=NDINT(K.IND)
IF (NDIN.EQ.0) GOTO 2040
OPIN=OPINT(K.IND)
DO 2010 KK=1.NDIN
2010 DIN(KK)=DINT(OPIN+KK-1.IND)
IDUM=0
DO 2020 KK=1.NDIN*2
IDUM=IDUM+1
IDB(IDUM)=(DIN(KK)-SAMSTR)/SAMINC+1
IDE(IDUM)=(DIN(KK+1)-SAMSTR)/SAMINC+1
IF (MODSS.ME.MOD(DIN(KK).SAMINC)) IDB(IDUM)=IDB(IDUM)+1
2020 CONTINUE
DO 2030 KK=1.IDUM
IDS=IDR(KK)
IDF=IDF(KK)
IF (IDS.GT.IE.OP.IH.GT.IDF) GOTO 2030
IF (IDS.LE.IH) IDS=IH
IF (IDF.GT.IE) IDF=IE
IF (IDS.GT.IDF) GOTO 2030
DO 2025 KKK=IDS.IDF
NDOU(IND)=NDOU(IND)+1
DO 2023 KKKK=1.NOFEAT
DUMMY1=KKK+NSAMP*(KKKK-1)
2023 IDATA(DUMMY1)=MEANDD
2025 CONTINUE
2030 CONTINUE
2040 CONTINUE
IF (IDSIT.EQ.1.AND.NDOU(IND).GT.0) DOFLAG=1
IF (IDSIT.EQ.2.AND.NDOU(IND).GT.0) DUFLAG=1
2050 CONTINUE
2055 DODU=DOFLAG+DUFLAG
DO 50 J=IH.IF
FLDSAM=FLDSAM+1
DO 50 K=1.NOFEAT
IWRD=IWRD+1
DUMMY2=J+NSAMP*(K-1)
RWRD=IDATA(DUMMY2)
DUMMY3=I*INDX-1+IWRD+BUFSIZ*(IBUF-1)
APRAY(DUMMY3)=ITGWRD
IF (IWRD.LT.BUFSIZ) GOTO 50
TOTWRD=TOTWRD+IWRD
IF (TOTWRD.GT.MADS) GOTO 35
DUMMY4=I*INDX+BUFSIZ*(IBUF-1)
CALL RWRITE(ADDRS.ARRAY(DUMMY4).BUFSIZ.LSTAT(IBUF))
IBUF=IBUF+1
ADDRS=ADDRS+BUFSIZ
IF (IBUF.GT.IBUFS) IBUF=1
40 IF (LSTAT(IBUF).EQ.1) GOTO 40
IWRD=0
50 CONTINUE
60 CONTINUE
10 CONTINUE
IPT = IPT + NV*2 + 2
APRAY(IPT)=FLDSAM
DO 15 I=1.6
IPT=IPT+1
15 ARRAY(IPT)=FLDINF(I)
IPT=IPT+1
IF (IPT+30 .GT. RESERV) GO TO 70
GO TO 5

C*
C* CLASS NAME CARD ENCOUNTERED - RERFAD PREVIOUS CARD TO GET NAME
C*
20 NOCL=NOCL+1
IF (NOCL.GT.1) GO TO 25
READ(30.100)NATCLS
REWIND 30
GO TO 4
```

9-18  
6-9  
ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: RDATA

```
25 CLSNAM=NXTCLS                                RDD03170
   WFAD(30,100)NXTCLS                          RDD03180
   GO TO 31                                     RDD03190
C* EMPTY LAST HUFFFR AND RETURN TO PROCESS DATA FOR THIS CLASS. RDD03200
30 CLSNAM=NXTCLS                                RDD03210
   LAST=1                                       RDD03220
C*                                               RDD03230
31 TOTWWD=TOTWRD+IWRD                          RDD03240
   IF (TOTWWD.GT.NWDS) GOTO 35                 RDD03250
   DUMMY=HFINDX+RUF51Z*(IHUF-1)               RDD03260
   CALL RWRITE(A0)RES,ARRAY(DUMMY),IWRD,LSTAT(IBUF) RDD03270
   TOTPTS=TOTWRD/NOFEAT                        RDD03280
   IF (TOTWWD+TOTPTS.LE.NWDS)RETURN            RDD03300
   WRITE(A,200)NWDS                             RDD03310
   RETURN                                       RDD03320
35 WRITE(A,200)NWDS                             RDD03330
   CALL CMERP                                  RDD03340
70 WRITE(A,300)RESERV                          RDD03350
   CALL CMERP                                  RDD03360
80 WRITE(A,400)                                RDD03370
   CALL CMERP                                  RDD03380
90 WRITE(A,700)IDIM                            RDD03390
   CALL CMERP                                  RDD03400
100 FORMAT(10X,A4)                              RDD03410
140 FORMAT(/// ' DESIGNATED OTHER OR UNIDENTIFIABLE FIELDS INPUT'//) RDD03420
200 FORMAT(' TOO MUCH DATA REQUESTED--PIXELS*(CHANNELS+1) CANNOT EXCEED RDD03430
   * )',I10)                                    RDD03440
300 FORMAT(' STORAGE REQUIRED FOR FIELD DEFINITION INFORMATION EXCEEDS RDD03450
   * THE DIMENSION LIMIT OF ',I5)              RDD03460
400 FORMAT(' END-OF-TAPE REACHED BEFORE END OF FIELD'//) RDD03470
500 FORMAT(//40X,' FIELDS TO BE CLUSTERED FOR CLASS ',I1,A4//) RDD03480
   * T36,'SAMPLE',T45,'LINE'/T20,'FIELD NAME',T36,'INC.', RDD03490
   * T45,'INC.',T77,'VERTICES (SAMPLE,LINE)'//) RDD03500
600 FORMAT(1X,T16,13,T22,A4,T36,14,T45,14,T60, RDD03510
   * 5(A),I4,'.',I4,'.')',I1) RDD03520
650 FORMAT(1X,T60,5(A),I4,'.',I4,'.')',I1) RDD03530
700 FORMAT(' NO OF PIXELS TO BE UNPACKED PER SCAN EXCEEDS THE DIMENS RDD03540
   * ON LIMIT OF ',I5) RDD03550
800 FORMAT(// 'INPUT ERROR - A CLASSNAME CAPD MUST BE INPUT BEFORE A GRDD03560
   *ROUP OF FIELDS'//) RDD03570
   RETURN                                       RDD03580
   END                                         RDD03590
```

## 10. SELECT PROCESSOR

### FILE SELECT

```

SUBROUTINE SELECT (ARRAY, TOP)
IMPLICIT INTEGER (A-H, O-Z)
-----
CALL..    CALL SELECT (ARRAY, TOP)
ARGS..    ARRAY - SFE 'MONTOR'
           TOP  - SFE 'MONTOR'

PURPOSE.. COORDINATES THE VARIOUS ROUTINES
           FOR 'FEATURE SELECTION' STEP

RETURNS.. NONE
-----

INCLUDE COMBK1, LIST
-----

INCLUDE COMRK6, LIST
DIMENSION ARRAY (1), SUBRAY (12000)
DATA SUBSIZ / 12000 /
C* THE ARRAY SUBRAY IS USED IN SELECT FOR VARIABLE DIMENSIONING
C* INCLUDE COMRK7, LIST
COMMON / INFORM / NOCLS2, NOSUB2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
* AVAR2, COVAR2, CLSID2, SUBNO2, SUBDS2, FLDSV2, VERTX2,
* FETVC2 (30), SUBVC2 (75), SUBPTR (75), CLSVC2 (60),
* KEPPTS (60), NOGRP, GRPNAM (60), GRPDEX (61),
* GRPCHK (61), GROUPS (124)
COMMON / GLOBAL / HEAD (63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
* HISFIL, HISKEY, TRFORM, ERIPTP, ERPKEY, MAPUNT, NOFILE,
* DRUMAD, DRMWDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
* NHSTUN, NHSTFI, SCTRUN, MAPFIL
* DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
* CRDUNT, PRTUNT, RNDPIO
COMMON / FSL / CFAC, TOTMSR, SEPMSR, PRCKEY, CRIKEY, INCFET,
* INCVEC (30), ICOUNT, SETWGT, EVALBF (100), FETVC4 (30)
* NOFET4, VARSZ4, COMBAS, DTAB4, WGH514, RESTVC (10), DIVSIZ
* STATKY, ADRESO, ADRESP, ADRESF, ADRSH1, ADRSH2
INTEGER ADRESO, ADRESP, ADRESF, ADRSH1, ADRSH2, STATKY
DOUBLE PRECISION CFAC, TOTMSR, SEPMSR

CSEND

COMMON BLOCK FSL IS USED ONLY BY THE 'SELECT' PROCESSOR

DEFINITIONS
PRCKEY - KEY INDICATING WHICH PROCEDURE TO EXECUTE
1 - EXHAUSTIVE SEARCH
2 - WITHOUT REPLACEMENT
3 - DAVIDON
4 - EVALUATE A USER INPUT B-MATRIX
5 - EVALUATE SPECIFIC CHANNELS INPUT BY USER
CRIKEY - KEY INDICATING WHICH CRITERIA IS TO BE USED
FOR MEASURING SEPARABILITY.
1 - WEIGHTED AV. DIVERGENCE
2 - WEIGHTED AV. TRANSFORMED DIVERGENCE
3 - WEIGHTED AV. KHATTACHARYYA DISTANCE
INCFET - NO. OF CHANNELS TO INCLUDE IN THE 'BEST' SET. THIS IS
USER INPUT ON THE 'INCLUDE' CONTROL CARD.
INCVEC - VECTOR CONTAINING THE CHANNELS TO BE INCLUDED. INPUT
ON THE 'INCLUDE' CARD. MEANINGFUL ONLY IF WITHOUT
REPLACEMENT PROCEDURE IS EXECUTED.
ICOUNT - MAX. NO. OF ITERATIONS IN DAVIDON PROCEDURE.

```

```

SEL00010
SEL00020
SEL00030
SEL00040
SEL00050
SEL00060
SEL00070
SEL00080
SEL00090
SEL00100
SEL00110
SEL00120
SEL00130
SEL00140
SEL00150
SEL00160
SEL00170
SEL00180
SEL00190
SEL00200
SEL00210
SEL00220
SEL00230
SEL00240
SEL00250
SEL00260
SEL00270
SEL00280
SEL00290
SEL00300
SEL00310
SEL00320
SEL00330
SEL00340
SEL00350
SEL00360
SEL00370
SEL00380
SEL00390
SEL00400
SEL00410
SEL00420
SEL00430
SEL00440
SEL00450
SEL00460
SEL00470
SEL00480
SEL00490
SEL00500
SEL00510
SEL00520
SEL00530
SEL00540
SEL00550
SEL00560
SEL00570
SEL00580
SEL00590
SEL00600
SEL00610
SEL00620
SEL00630
SEL00640
SEL00650
SEL00660
SEL00670
SEL00680
SEL00690
SEL00700
SEL00710
SEL00720
SEL00730
SEL00740
SEL00750
SEL00760

```

FILE SELECT

```

C*          SETWGT - TRIGGER INDICATING WHETHER OR NOT DEFAULT WEIGHTS      SEL00770
C*          ARE TO BE SET.                                                  SEL00780
C*          EVALBF - BUFFER CONTAINING ALL USER 'EVALUATE' REQUESTS.        SEL00790
C*          EVALRF (1) = NO. OF CHANNELS FOR FIRST REQUEST                  SEL00800
C*          (2-N) = A SET OF CHANNELS TO BE EVALUATED                     SEL00810
C*          (N+1) = NO. OF CHANNELS ON SECOND REQUEST                       SEL00820
C*          -                                                                 SEL00830
IEY0331 COMMENTS DELETED *****
COMMON/RESTKN/ KPPPTS(60), IPRIOR, KBEST, NCPASS                          SEL01190
DIMENSION ILABLX(13), ILARLY(13)                                         SEL01200
DIMENSION FETVEC(30)                                                       SEL01210
DIMENSION PER(6), INDPER(6), PERM(248)                                    SEL01220
DATA INDPER/1,13,58,138,213,249/                                         SEL01230
DATA PERM/1,2,1,3,1,4,1,5,1,6,1,7,                                     SEL01240
* 1,2,3,1,2,4,1,2,5,1,2,6,1,2,7,1,3,4,1,3,5,1,3,6,1,3,7,             SEL01250
* 1,4,5,1,4,6,1,4,7,1,5,6,1,5,7,1,6,7,                                 SEL01260
* 1,2,3,4,1,2,3,5,1,2,3,6,1,2,3,7,1,2,4,5,1,2,4,6,                 SEL01270
* 1,2,4,7,1,2,5,6,1,2,5,7,1,2,6,7,1,3,4,5,1,3,4,6,                 SEL01280
* 1,3,4,7,1,3,5,6,1,3,5,7,1,3,6,7,1,4,5,6,                           SEL01290
* 1,4,5,7,1,4,6,7,1,5,6,7,                                             SEL01300
* 1,2,3,4,5,1,2,3,4,6,1,2,3,4,7,1,2,3,5,6,1,2,3,5,7,             SEL01310
* 1,2,3,6,7,1,2,4,5,6,1,2,4,5,7,1,2,4,6,7,1,2,5,6,7,             SEL01320
* 1,3,4,5,6,1,3,4,5,7,1,3,4,6,7,1,3,5,6,7,1,4,5,6,7,             SEL01330
* 1,2,3,4,5,6,1,2,3,4,5,7,1,2,3,4,6,7,1,2,3,5,6,7,             SEL01340
* 1,2,4,5,6,7,1,3,4,5,6,7/                                           SEL01350
DIMENSION IPSCHK(8)                                                       SEL01360
DIMENSION FETSAV(30), FTSABV(30)                                         SEL01370
DOUBLE PRECISION SMALL                                                    SEL01380
SMALL = 2**35                                                            SEL01390
JTIME = 1                                                                  SEL01400
NFSAVE = 0                                                                SEL01410
10 CONTINUE                                                                SEL01420
CALL SETUP4 (ARRAY, TOP, STOPFG, JTIME, SUBRAY, SUBSIZ)                   SEL01430
IF (STOPFG.NE.0) GO TO 9                                                  SEL01440
IF (PRCKEY.NE.6) GO TO 9                                                  SEL01450
DO 6 I=1,8                                                                SEL01460
6  IPSCHK(I) = 0                                                           SEL01470
DO 7 I=1,NOFET2                                                           SEL01480
7  K=(FETVC2(I)-1)/NCPASS +1                                             SEL01490
IPSCHK(K) = 1                                                             SEL01500
NPASS = 0                                                                  SEL01510
DO 8 I=1,8                                                                SEL01520
8  NPASS = NPASS + IPSCHK(I)                                             SEL01530
NFPSS = NOFET2/NPASS                                                     SEL01540
IDUM = NFPSS*NPASS                                                       SEL01550
IF (NOFET2.NE.IDUM) GO TO 90                                              SEL01560
IF (KREST.LT.2.OR.KREST.GE.NPASS) GO TO 90                              SEL01570
NOFET4 = NFPSS*KREST                                                     SEL01580
NFSAVE = NOFET4                                                           SEL01590
KPASS = KBEST                                                             SEL01600
9  CONTINUE                                                                SEL01610
JTIME = JTIME+1                                                           SEL01620
IF (STOPFG.EQ.0) GO TO 5                                                 SEL01630
C* SET FETVC2 FOR CLASSIFY                                               SEL01640
IF (NFSAVE.NE.0) NOFET4=NFSAVE                                           SEL01650
DO 4 I=1,NOFET4                                                           SEL01660
4  FETVC2(I)=FETVC4(I)                                                   SEL01670
CALL ORDER(FETVC2,NOFET4)                                               SEL01680
NOFET2 = NOFET4                                                           SEL01690
RETURN                                                                    SEL01700
5  CONTINUE                                                                SEL01710
C* SET ADRESD FOR RANDOM ACCESS DRUM FILE                                SEL01720
C* ADRESD=DRUMAD                                                         SEL01730
C* JREST = 0                                                             SEL01740
C* PRELIM--TAKE CARE OF PRELIMINARIES                                   SEL01750
C* COMPUTE SEPARABILITY MEASURE AND INTERCLASS MEASURES USING ALL       SEL01760
C* FEATURES, AND SAVE ON SCRATCH FILE FOR LATER PRINTING. IF CRIKEY=1   SEL01770
C* SET DEFAULT WEIGHTS IF WEIGHTS NOT INPUT, ALSO COMPUTE 'S' MATRIX.   SEL01780
C*                                                                       SEL01790
SRASE=1                                                                    SEL01800
S1=SRASE                                                                    SEL01810
SRASE=S1+NOCLS2*VARSZ2+1                                                 SEL01820
IF (CRIKEY.NE.1) SRASE=1                                                 SEL01830
SLEFT=SURSZ-SRASE                                                         SEL01840
SLEFT=SURSZ-SRASE                                                         SEL01850
SLEFT=SURSZ-SRASE                                                         SEL01860

```



FILE SELECT

```

CALL PRELIM(ARRAY(COVAR2),ARRAY(AVAR2),ARRAY(DTAB4),
* ARRAY(WGHS14),SUBRAY(S1),SUBRAY(SBASE),SLEFT)
IF(PRCKEY.EQ.5)GO TO 60
IF(PRCKEY.EQ.4)GO TO 12
IF(PRCKEY.EQ.6)GO TO 15
11 JRFST=JBEST+1
NFSAVE=NOFET4
NOFET4=BESTVC(JBEST)
IF(NOFET4.LE.0)GO TO 60
C*
C* IF DAVIDON PROCEDURE INDICATED, FIND BEST SET OF FEATURES BY
C* WITHOUT REPLACEMENT, IF FIRST GUESS B-MATRIX WAS NOT INPUT.
C*
12 IF(PRCKEY.NE.3)GO TO 15
C*
C* SET ADDRESSES FOR RANDOM ACCESS DRUM FILE
C*
ADRESF=ADRESD+DIVSIZ*2
ADRESF=ADRESF+NOFET4*NOFET2*2
ADRSH1=ADRESF+NOFET4*NOFET2*2
ADRSH2=ADRSH1+(NOFET4*NOFET2*2)**2
C*
C* WAS FIRST GUESS B-MATRIX INPUT
C*
IF(WMKEY.EQ.1)GO TO 15
SAVPRC=3
PRCKEY=2
C*
C* COMPUTE BASES FOR ARRAYS OF 'BEST' SET OF FEATURES
C* TRANSFORMED COVARIANCES AND MEANS STORED IN DOUBLE PRECISION
C*
15 VAPSZ4=NOFET4*(NOFET4+1)/2
COVAR4=CORRAS
AVAR4=COVAR4 + NOCLS2*VARSZ4*2
CORRSS = AVAR4 + NOCLS2 * NOFET4 *2
IF(CORRSS .LE. TOP) GO TO 20
WRITE(6,200) CORRSS
CALL CMERR
C*
C* SUBRAY STORAGE - STORE 'S' ARRAYS ONLY IF CRIKEY=1, STORE PARTIALS
C* ONLY IF PRCKEY=3, STORE B-MATRIX IF PRCKEY=3 OR 4.
C*
20 S2=S1 + NOCLS2*VARSZ2
R1=S2 + NOCLS2*VARSZ4*2
IF(CRIKEY.NE.1)H1=1
P1=H1 + NOFET4*NOFET2*2
SBASE=P1 + NOFET4*NOFET2*2
IF(PRCKEY.NE.3)SBASE=P1
IF(PRCKEY.LT.3)SBASE=B1
SLFFT=SUHSIZ-SBASE
IF(SBASE.LE.SUHSIZ)GO TO 25
WRITE(6,100)SBASE
CALL CMERR
C*
C* PERFORM THE OPTIMIZATION PROCEDURE INDICATED BY PRCKEY
C*
25 GO TO (30,35,40,45,85,87),PRCKEY
C*
C* EXHAUSTIVE SEARCH PROCEDURE
C*
30 CALL EXSRCH(ARRAY(COVAR2),ARRAY(AVAR2),ARRAY(DTAB4),ARRAY(WGHS14)
* ,ARRAY(COVAR4),ARRAY(AVAR4),SUBRAY(S1),SUBRAY(S2),
* SUBRAY(SBASE),SLEFT)
GO TO 50
C*
C* WITHOUT REPLACEMENT PROCEDURE
C*
35 CALL WHRPLC(ARRAY(COVAR2),ARRAY(AVAR2),ARRAY(DTAB4),ARRAY(WGHS14)
* ,ARRAY(COVAR4),ARRAY(AVAR4),SUBRAY(S1),SUBRAY(S2),
* SUBRAY(SBASE),SLEFT)
GO TO 50
C*
C* DAVIDON PROCEDURE
C*
40 CALL DAVIDN(ARRAY(COVAR2),ARRAY(AVAR2),ARRAY(DTAB4),ARRAY(WGHS14)
* ,ARRAY(COVAR4),ARRAY(AVAR4),SUBRAY(S1),SUBRAY(S2)

```

SEL01870  
SEL01880  
SEL01890  
SEL01900  
SEL01910  
SEL01920  
SEL01930  
SEL01940  
SEL01950  
SEL01960  
SEL01970  
SEL01980  
SEL01990  
SEL02000  
SEL02010  
SEL02020  
SEL02030  
SEL02040  
SEL02050  
SEL02060  
SEL02070  
SEL02080  
SEL02090  
SEL02100  
SEL02110  
SEL02120  
SEL02130  
SEL02140  
SEL02150  
SEL02160  
SEL02170  
SEL02180  
SEL02190  
SEL02200  
SEL02210  
SEL02220  
SEL02230  
SEL02240  
SEL02250  
SEL02260  
SEL02270  
SEL02280  
SEL02290  
SEL02300  
SEL02310  
SEL02320  
SEL02330  
SEL02340  
SEL02350  
SEL02360  
SEL02370  
SEL02380  
SEL02390  
SEL02400  
SEL02410  
SEL02420  
SEL02430  
SEL02440  
SEL02450  
SEL02460  
SEL02470  
SEL02480  
SEL02490  
SEL02500  
SEL02510  
SEL02520  
SEL02530  
SEL02540  
SEL02550  
SEL02560  
SEL02570  
SEL02580  
SEL02590  
SEL02600  
SEL02610  
SEL02620

FILE SELECT

```

      * ,SURRAY (B1) ,SUBRAY (P1) ,SUHRAY (SRASE) ,SLEFT)
      CALL WRBTMT (SUBRAY (H1) ,NOFFT4 ,NOFET2 ,FETVC2)
      GO TO 50
C*
C*
C* USER INPUT B-MATRIX
45 CALL USERIN (ARRAY (COVAR2) ,ARRAY (AVAR2) ,ARRAY (DTAB4) ,ARRAY (WGHS14)
      * ,ARRAY (COVAR4) ,ARRAY (AVAR4) ,SUBRAY (S1) ,SUBRAY (S2)
      * ,SURRAY (B1) ,SUBRAY (SBASE) ,SLEFT)
C*
C*
C* GENERATE REPORTS
50 CALL GENRPT (ARRAY (CLSID2) ,ARRAY (WGHS14) ,ARRAY (DTAB4) ,
      * SURRAY (SBASE) ,SLEFT ,FETVEC)
      CALL PLOT (SUBRAY (SRASE) ,ARRAY (DTAB4) ,DIVSIZ ,MAXX ,ILABLX ,ILABLY ,
      * ICODE ,IOPT)
C*
      IF (SAVPHC .NE. 3) GO TO 11
      SAVPRC = 0
      PRCKEY = 3
      GO TO 20
C*
C*
C* PERFORM EVALUATE REQUEST
60 IV = 1
      ISAVE = PRCKEY
      PRCKEY = 5
70 NOFFT4 = EVALHF (IV)
      IF (NOFFT4 .GT. 0) GO TO 75
      PRCKEY = ISAVE
      GO TO 10
75 DO 90 I = 1 ,NOFFT4
      IV = IV + 1
80 FETVEC (I) = EVALBF (IV)
C
C
C RENUMBERING CHANNELS IN REFERENCE TO SUBSET OF CHANNELS
      DO 82 I = 1 ,NOFFT2
      DO 82 J = 1 ,NOFFT4
      IF (FETVEC (J) .NE. FETVC2 (I)) GO TO 82
      FETVC4 (J) = I
82 CONTINUE
      CALL ORDER (FETVC4 ,NOFFT4)
C*
C*
C* GO COMPUTE BASE ADDRESSES FOR REDUCED ARRAYS
      GO TO 15
85 CONTINUE
      CALL EVLFET (ARRAY (COVAR2) ,ARRAY (AVAR2) ,ARRAY (DTAB4) ,ARRAY (WGHS14)
      * ,ARRAY (COVAR4) ,ARRAY (AVAR4) ,SUBRAY (S1) ,SUBRAY (S2)
      * ,SURRAY (SBASE) ,SLEFT)
      CALL GENRPT (ARRAY (CLSID2) ,ARRAY (WGHS14) ,ARRAY (DTAB4)
      * ,SUBRAY (SBASE) ,SLEFT ,FETVEC)
      CALL PLOT (SUBRAY (SRASE) ,ARRAY (DTAB4) ,DIVSIZ ,MAXX ,ILABLX ,ILABLY ,
      * ICODE ,IOPT)
      IV = IV + 1
      GO TO 70
C
C BEST K OF N PASSES
87 CONTINUE
91 NCNT = NPASS + 1 - KPASS
      IDUM = INDPER (KPASS - 1)
      NCNTP = (INDPER (KPASS) - IDUM) / KPASS
      DO 99 I = 1 ,NCNT
      DO 98 II = 1 ,NCNTP
      DO 92 III = 1 ,KPASS
      IDM = IDUM + (II - 1) * KPASS + III - 1
      PER (III) = PER (IDM) + I - 1
      IF (PER (III) .GT. NPASS) GO TO 97
92 CONTINUE
      DO 94 III = 1 ,KPASS
      IDM = PER (III)
      DO 93 IIII = 1 ,NFPPS
      IDMA = NFPPS * (III - 1) + IIII
      IDMB = NFPPS * (IDM - 1) + IIII
      FETVC4 (IDMA) = IDMB
93 FETVC4 (IDMA) = FETVC2 (IDMB)
94 CONTINUE

```

SEL 02630  
 SEL 02640  
 SEL 02650  
 SEL 02660  
 SEL 02670  
 SEL 02680  
 SEL 02690  
 SEL 02700  
 SEL 02710  
 SEL 02720  
 SEL 02730  
 SEL 02740  
 SEL 02750  
 SEL 02760  
 SEL 02770  
 SEL 02780  
 SEL 02790  
 SEL 02800  
 SEL 02810  
 SEL 02820  
 SEL 02830  
 SEL 02840  
 SEL 02850  
 SEL 02860  
 SEL 02870  
 SEL 02880  
 SEL 02890  
 SEL 02900  
 SEL 02910  
 SEL 02920  
 SEL 02930  
 SEL 02940  
 SEL 02950  
 SEL 02960  
 SEL 02970  
 SEL 02980  
 SEL 02990  
 SEL 03000  
 SEL 03010  
 SEL 03020  
 SEL 03030  
 SEL 03040  
 SEL 03050  
 SEL 03060  
 SEL 03070  
 SEL 03080  
 SEL 03090  
 SEL 03100  
 SEL 03110  
 SEL 03120  
 SEL 03130  
 SEL 03140  
 SEL 03150  
 SEL 03160  
 SEL 03170  
 SEL 03180  
 SEL 03190  
 SEL 03200  
 SEL 03210  
 SEL 03220  
 SEL 03230  
 SEL 03240  
 SEL 03250  
 SEL 03260  
 SEL 03270  
 SEL 03280  
 SEL 03290  
 SEL 03300  
 SEL 03310  
 SEL 03320  
 SEL 03330  
 SEL 03340  
 SEL 03350  
 SEL 03360  
 SEL 03370  
 SEL 03380

FILE SELECT

```

      CALL EVLFET (ARRAY (COVAR2), ARRAY (AVAR2),
    * ARRAY (DTAH4), ARRAY (WGHS14), ARRAY (COVAR4), ARRAY (AVAR4),
    * SUBRAY (S1), SUBRAY (S2), SUBRAY (SBASE), SLEFT)
      IF (SMALL.LT.SEPMSR) GO TO 96
      DO 95 III = 1, NOFET4
      FTSAV (III) = FETVEC (III)
95    FETVC4 (III) = FETVC4 (III)
      SMALL = SEPMSR
96    CONTINUE
      CALL GENRPT (ARRAY (CLS102), ARRAY (WGHS14),
    * ARRAY (DTAR4), SUBRAY (SBASE), SLEFT, FETVEC)
97    CONTINUE
98    CONTINUE
99    CONTINUE
      DO 101 I=1, NOFET4
      FETVEC (I) = FTSAV (I)
101   FETVC4 (I) = FTSAV (I)
      WRITE (6,1010)
1010  FORMAT (1H1, 'BEST SEPARABILITY MEASURE', //)
      WRITE (6,1000) SMALL
      WRITE (6,1020)
1020  FORMAT (// ' CORRESPONDING FEATURES', //)
      WRITE (6,1000) (FETVEC (I), I=1, NOFET4)
1000  FORMAT ( )
      GO TO 10
90    WRITE (6,250) NOFET4, (FETVC2 (I), I=1, NOFET2)
250   FORMAT (' ERROR IN INPUT CHANNELS' //, 1X, I2, 30 (1X, I2))
      CALL CMERR
100   FORMAT (' CORE OVERFLOW IN SURRAY-', I6, ' STORAGE LOCATIONS NECESSAR
    *Y FOR THIS PROBLEM')
200   FORMAT (' CORE OVERFLOW IN ARRAY-', I6, ' STORAGE LOCATIONS NEEDED FO
    *R THIS PROBLEM')
      END

```

SFL03390  
 SFL03400  
 SFL03410  
 SFL03420  
 SFL03430  
 SFL03440  
 SFL03450  
 SFL03460  
 SFL03470  
 SFL03480  
 SFL03490  
 SFL03500  
 SFL03510  
 SFL03520  
 SFL03530  
 SFL03540  
 SFL03550  
 SFL03560  
 SFL03570  
 SFL03580  
 SFL03590  
 SFL03600  
 SFL03610  
 SFL03620  
 SFL03630  
 SFL03640  
 SFL03650  
 SFL03660  
 SFL03670  
 SFL03680  
 SFL03690  
 SFL03700  
 SFL03710



FILE: AVEDIV

```
DO 25 K=1, KK                                AVE00810
L=K-1                                        AVE00820
25 WRKY(IW1+L)=WRKY(IW2+L)-WRKY(IW)+L)      AVE00830
CALL MT2(WRKY(ICV),WRKY(IW),WRKY(IW2),NOFET4,NOFET2) AVE00840
DO 25 K=1, KK                                AVE00850
L=K-1                                        AVE00860
26 PARTLS(K)=WRKY(IW2+L)+PARTLS(K)          AVE00870
30 CONTINUE                                  AVE00880
SMSR = -(CFAC*SMSR/2. - NF)                  AVE00890
IF (IPART.LT.0) RETURN                       AVE00900
DO 40 K=1, KK                                AVE00910
40 PARTLS(K)=-CFAC*PARTLS(K)                AVE00920
RETURN                                        AVE00930
100 FORMAT(' REDUCED COVARIANCE MATRIX FOR CLASS',I3,' IS NOT POSITIVE AVE00940
* OFFINITE')                                  AVE00950
200 FORMAT(' MORE STORAGE NEEDED IN SUB. AVEDIV FOR WORK ARRAY--WORK SAVE AVE00960
* IZ=',I7)                                    AVE00970
END                                            AVE00980
```



FILE: HMTCHR

```
IF (IFULL.EQ.1) GO TO 8
WRKRY(ICV2+IK-1)=COVMT2(IK,J)
WRKRY(IW1+IK-1)=COVMT2(IK,J)+COVMT2(IK,I)
GO TO 10
A WRKRY(ICV2+IK-1)=COVMTX(IK,J)
WRKRY(IW1+IK-1)=COVMTX(IK,J)+COVMTX(IK,I)
10 CONTINUE
CALL COLINV(WRKRY(ICV2),NF,IERR,3,DET2)
NOF2=1./DET2
IF (IERR.EQ.0) GO TO 15
WRITE(A,100)J
GO TO 50
15 CALL COLINV(WRKRY(IW1),NF,IERR,3,DET3)
DET3=1./DET3
IF (IERR.EQ.0) GO TO 16
WRITE(A,200)I,J
GO TO 50
16 IF (IFULL.NE.1) GO TO 1A
DO 17 IK=1,NOFET2
17 T(IK)=AVEMTX(IK,I)-AVEMTX(IK,J)
GO TO 25
1A DO 20 IK=1,NOFET4
20 T(IK)=AVEMT2(IK,I)-AVEMT2(IK,J)
C MULTIPLY T TRANSPOSE TIMES WRKRY
25 CONTINUE
CALL MT4(WRKRY(IW1),T,WRKRY(IW2),NF,NF,1,1)
RNUM=0.0
DO 24 IK=1,NF
24 RNUM=RNUM+WRKRY(IW2+IK-1)*T(IK)
DIVTAB(NM)=DFIP(-.25*RNUM-.5*DLG(DET3/(2.*NF
*DSQRT(DET1*DET2))))
SMSR=SMSR+WEIGHT(NM)*DIVTAB(NM)
IF (IPART.LT.0) GO TO 50
C*
C* COMPUTE PARTIALS
C*
CALL MT4(WRKRY(IW2),BMAT,WRKRY(IW3),1,NOFET4,NOFET2,0)
DO 30 IK=1,VAR522
30 WRKRY(IK+IW4-1)=COVMTX(IK,J)+COVMTX(IK,I)
CALL MT4(WRKRY(IW4),WRKRY(IW3),WRKRY(IW2),NOFET2,NOFET2,1,1)
DO 35 IK=1,NOFET2
35 WRKRY(IW3+IK-1)=AVEMTX(IK,I)-AVEMTX(IK,J)-WRKRY(IW2+IK-1)
CALL MT4(T,WRKRY(IW3),WRKRY(IW5),NOFET4,1,NOFET2,0)
CALL MT4(WRKRY(IW1),WRKRY(IW5),WRKRY(IW6),NOFET4,NOFET4,NOFET2,1)
DO 40 IK=1,10
40 WRKRY(IW6+IK-1)=WRKRY(IW6+IK-1)/2
M=T
IC=ICV1
41 DO 45 IK=1,VAR524
L=IK-1
45 WRKRY(IW4+L)=WRKRY(IW4+L)-WRKRY(IC+L)/2
CALL MT4(WRKRY(IW4),BMAT,WRKRY(IW7),NOFET4,NOFET4,NOFET2,1)
CALL MT4(WRKRY(IW7),COVMTX(1,M),WRKRY(IW5),NOFET4,NOFET2)
DO 43 IK=1,10
L=IK-1
43 WRKRY(IW4+L)=WRKRY(IW4+L)+WRKRY(IW5+L)
IF (M.EQ.0) GO TO 46
M=J
IC=ICV2
GO TO 41
46 DO 47 IK=1,10
L=IK-1
PARTLS(IK)=PARTLS(IK)-WEIGHT(NM)*DIVTAB(NM)*WRKRY(IW6+L)/NOCLS2
47 CONTINUE
50 CONTINUE
60 CONTINUE
SMSR=SMSR/NOCLS2
DETERM
100 FORMAT(' COVAR FOR CLASS',I3,' IS NOT POSITIVE DEFINITE')
200 FORMAT(' COVAR FOR SUM OF CLASSES',I4,' IS NOT POSITIVE DEF.')
300 FORMAT(' NOT ENOUGH WORK AREA AVAILABLE IN HMTCHR -- IWRK52=',I5)
END
```

FILE: BSTCHK

```

SUBROUTINE BSTCHK(NOBEST)
C
C-----
IMPLICIT INTEGER (A-M,0-Z)
C-----
CALL.. CALL BSTCHK(NOBEST)
ARGS.. NOBEST - NO OF FEATURES-TUPLES TO ANALYZE
REQUIRES.. COMMON /INFORM/
PURPOSE.. CHECKS VALIDITY OF REQUESTED FEATURE-TUPLES
RETURNS.. CORRECT FEATURE-TUPLE QUE
C-----
CALL.. CALL SHWCHK(COMBUF,CPTR)
ARGS.. COMBUF - SHOW REQUEST QUE
PURPOSE.. CHECKS VALIDITY OF SHOW REQUESTS
RETURNS.. CORRECT SHOW REQUEST QUE
C-----
DIMENSION COMBUF(1)
INCLUDE COMPK1,LIST
C-----
DIMENSION INVERT(30)
INCLUDE COMPK7,LIST
COMMON/INFORM/NOCLS?,NOVAR?,NOFET?,VAR5Z?,TOTVT?,NOFLD?,
* AVAR?,COVAR?,CLSID?,SUBNO?,SURDS?,FLDSY?,VERTX?,
* FFTVC?(30),SURVC?(75),SUBPTR(75),CLVC?(60),
* KPPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),
* GHPCHK(6),GRUPS(124)
COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CHIKEY,INCFET,
* INCVEC(30),ICOUNT,SETLGT,EVALHF(100),FFTV(4),
* NOFFT?,VAR574,COMBAS,DTAR4,WGMS14,BESTVC(10),D:VSIZ
* ,STATKY,ADRES?,ADRESF,ADRESF,ADMSH1,ADMSH2
INTEGER ADRES?,ADRESF,ADRESF,ADMSH1,ADMSH2,STATKY
DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
CSEMD
II = 0
DO 30 I=1,NOBEST
J = RESTVC(I)
IF (J.GT.NOFFT?)GO TO 10
II = II+1
RESTVC(II) = J
GO TO 30
10 WRITE (A,201.)
20 FORMAT('A07A0 'HTEST'?,13.' IS GREATER THAN OR EQUAL TO NO. OF F
* FEATURES IN GIVEN DATA...[IGNORED]')
30 CONTINUE
NOBEST = II
C
C-----
RETURN
C-----
ENTRY EVLCHK(COMBUF,CPTH)
DO 32 I=1,NOFFT?
INVERT(I)=0
DO 33 I=1,NOFFT?
K = FFTVC(I)
33 INVERT(K)=I
K = 0

```

HST00010  
HST00020  
HST00030  
HST00040  
HST00050  
HST00060  
HST00070  
HST00080  
HST00090  
HST00100  
HST00110  
HST00120  
HST00130  
HST00140  
HST00150  
HST00160  
HST00170  
HST00180  
HST00190  
HST00200  
HST00210  
HST00220  
HST00230  
HST00240  
HST00250  
HST00260  
HST00270  
HST00280  
HST00290  
HST00300  
HST00310  
HST00320  
HST00330  
HST00340  
HST00350  
HST00360  
HST00370  
HST00380  
HST00390  
HST00400  
HST00410  
HST00420  
HST00430  
HST00440  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00060  
HST00520  
HST00530  
HST00540  
HST00550  
HST00560  
HST00570  
HST00580  
HST00590  
HST00600  
HST00610  
HST00620  
HST00630  
HST00640  
HST00650  
HST00660  
HST00670  
HST00680  
HST00690  
HST00700  
HST00710  
HST00720  
HST00730  
HST00740  
HST00750  
HST00760  
HST00770  
HST00780  
HST00790  
HST00800



FILE: BSTCHK

I = 1	PST00810
40 IF (I.GE.CPTR) GO TO 100	PST00820
N = COMBUF(I)	PST00830
J = I+N	PST00840
I = I+1	PST00850
LAST = 0	PST00860
DO 50 L=I,J	PST00870
LL = COMBUF(L)	PST00880
IF (LL.LE.LAST.OR.INVERT(LL).EQ.0.OR.LL.GT.FETVC2(NOFET2)) GO TO 80	PST00890
10	PST00900
50 LAST = LL	PST00910
K = K+1	PST00920
COMBUF(K) = N	PST00930
DO 60 L=I,J	PST00940
K = K+1	PST00950
60 COMBUF(K) = COMBUF(L)	PST00960
70 I = J+1	PST00970
GO TO 40	PST00980
C	PST00990
80 WRITE (6,90) (COMBUF(LL),LL=I,J)	PST01000
90 FORMAT(' INVALID EVALUATE REQUEST.....',I4)	PST01010
GO TO 70	PST01020
C	PST01030
100 CPTR = K	PST01040
C	PST01050
RETURN	PST01060
END	PST01070

FILE: COLINV

SUBROUTINE COLINV(S,N,IERR,IND,DET)  
DOUBLE PRECISION S,SUR,DET

PURPOSE

INVERT A GIVEN SYMMETRIC POSITIVE DEFINITE MATRIX (S) BY  
COMPUTING A TRIANGULAR FACTORIZATION (R), INVERTING R TO  
OBTAIN A, AND THEN FINDING THE INVERSE OF S.  
S=R\*(TRANSPOSE OF R), A=(INVERSE OF R),  
(INVERSE OF S)=(TRANSPOSE OF A)\*A .

ARGUMENTS

S - LOWER TRIANGULAR PART OF THE GIVEN SYMMETRIC MATRIX  
STORED ROWWISE IN N\*(N+1)/2 SUCCESSIVE STORAGE LOCATIONS.  
ON RETURN S CONTAINS THE LOWER TRIANGULAR MATRIX R,  
THE LOWER TRIANGULAR MATRIX A, OR THE LOWER TRIANGULAR  
PART OF THE INVERSE OF S, DEPENDING ON VALUE  
ASSIGNED TO IND.  
N - THE NUMBER OF ROWS OR COLUMNS IN GIVEN MATRIX.  
IERR - RESULTING ERROR PARAMETER CODED AS FOLLOWS  
IERR=0 -NO ERROR  
IERR=-1 -MATRIX S IS NOT POSITIVE DEFINITE  
IND -PARAMETER INDICATING WHICH MATRIX IS RETURNED --  
IND=1 -MATRIX R  
IND=2 -MATRIX A  
IND=3 -MATRIX (INVERSE OF S)  
DET -DETERMINANT OF RETURNED MATRIX.

REFERENCE

'INVERSION OF SYMMETRIC POSITIVE DEFINITE MATRICES,' BY  
J.K. BRYAN AND D.L. TERBE, E.E. DEPT., UNIV. OF MISSOURI, 1971.

DIMENSION S(1)  
IERR = 0  
LS = 0  
N2 = 1  
DET = 1.0  
DO 70 I=1,N  
N2 = N2 + I - 1  
M = I\*(I+1)/2  
L = N2  
DO 40 J=I,N  
L = L + J - 1  
IF( I .EQ. 1 ) GO TO 20  
M1 = J\*(J-1)/2 + 1  
L1 = L - 1  
N1 = N2  
DO 10 K=M1,L1  
S(L) = S(L) - S(N1)\*S(K)  
10 N1 = N1 + 1  
20 IF( J .EQ. 1 ) GO TO 30  
S(L) = S(L)/S(M)  
GO TO 40  
30 IF( S(L) .LE. 0.0 ) GO TO 100  
S(L) = DSORT(S(L))  
DET = DET\*S(L)  
40 CONTINUE  
IF( IND.EQ.1 ) GO TO 70  
S(M) = 1./S(M)  
IF( I .EQ. 1 ) GO TO 70  
I1 = I - 1  
DO 60 J=1,I1  
K1 = I - J  
L1 = LS  
LS = LS + 1  
M1 = J\*(J-1)/2 + 1  
J1 = J - 1  
SUR = 0.0  
DO 50 K=1,K1  
L1 = L1 + 1  
J1 = J1 + 1  
M1 = M1 + J1 - 1  
50 SUR = SUR + S(L1)\*S(M1)  
60 S(LS) = -S(M)\*SUR  
70 LS = LS + 1  
IF( IND.EQ.2 .OR. IND.EQ.3 ) DET=1./DET  
IF( IND.EQ.1 .OR. IND.EQ.2 ) RETURN  
DO 90 I=1,N  
M = I\*(I-1)/2 + 1  
L = M

COL00010  
COL00020  
COL00030  
COL00040  
COL00050  
COL00060  
COL00070  
COL00080  
COL00090  
COL00100  
COL00110  
COL00120  
COL00130  
COL00140  
COL00150  
COL00160  
COL00170  
COL00180  
COL00190  
COL00200  
COL00210  
COL00220  
COL00230  
COL00240  
COL00250  
COL00260  
COL00270  
COL00280  
COL00290  
COL00300  
COL00310  
COL00320  
COL00330  
COL00340  
COL00350  
COL00360  
COL00370  
COL00380  
COL00390  
COL00400  
COL00410  
COL00420  
COL00430  
COL00440  
COL00450  
COL00460  
COL00470  
COL00480  
COL00490  
COL00500  
COL00510  
COL00520  
COL00530  
COL00540  
COL00550  
COL00560  
COL00570  
COL00580  
COL00590  
COL00600  
COL00610  
COL00620  
COL00630  
COL00640  
COL00650  
COL00660  
COL00670  
COL00680  
COL00690  
COL00700  
COL00710  
COL00720  
COL00730  
COL00740  
COL00750  
COL00760  
COL00770  
COL00780  
COL00790

FILE: COLINV

```
DO 90 J=I,N
L2 = L
L = L + J - 1
SUM = 0.0
DO 80 K=J,N
L2 = L2 + K - 1
M1 = L2 + J - 1
80 SUM = SUB + S(L2)*S(M1)
90 S(L) = SUM
DET = DET*DET
RETURN
100 IERR = -1
RETURN
END
```

COL00800  
COL00810  
COL00820  
COL00830  
COL00840  
COL00850  
COL00860  
COL00870  
COL00880  
COL00890  
COL00900  
COL00910  
COL00920  
COL00930

FILE: CONVRT

	SUBROUTINE CONVRT(NOSPAC,IFLG)	CON00010
C		CON00020
C	THE PURPOSE OF THIS ROUTINE IS TO CONVERT EBCDIC CHARS TO COMPUTATIONAL	CON00030
C	OR TO CONVERT COMPUTATIONAL CHARS TO EBCDIC CHARS	CON00040
C	LOGICAL*1 LTEST(4),LSPAC(4)	CON00050
C	EQUIVALENCE (ITEST,LTEST(4)), (ICHAR,LSPAC(4))	CON00060
C	DATA BCD0/ZF0/,BCD9/ZF9/	CON00070
C	ITEST = 0	CON00080
C	ICHAR = NOSPAC	CON00090
C	LTEST(4) = LSPAC(4)	CON00100
C		CON00110
C	CK FOR FLAG NOT 0, IMPLYING CHAR IS EBCDIC	CON00120
C	IF (IFLG .NE. 0) GO TO 1060	CON00130
C		CON00140
C	CK FOR COMPUTATIONAL NUMBER & CONVERT TO EBCDIC	CON00150
C	IF (ITEST .LT. 0 .OR. ITEST .GT. 9) GO TO 9999	CON00160
C	ITEST = ITEST + BCD0	CON00170
C	LSPAC(4) = LTEST(4)	CON00180
C	NOSPAC = ICHAR	CON00190
C	RETURN	CON00200
C		CON00210
C	CK FOR EBCDIC NUMBER TO CONVERT TO COMPUTATIONAL NUMBER	CON00220
C	1060 IF (ITEST .LT. BCD0 .OR. ITEST .GT. BCD9) GO TO 9999	CON00230
C	ITEST = ITEST - BCD0	CON00240
C	LSPAC(4) = LTEST(4)	CON00250
C	NOSPAC = ICHAR	CON00260
C	RETURN	CON00270
C		CON00280
C	ERROR	CON00290
C	9999 WRITE (6,9999)	CON00300
C	9999 FORMAT (1X,'ERROR - NON-NUMERIC CHARACTER INPUT')	CON00310
C	RETURN	CON00320
C	END	CON00330
		CON00340
		CON00350

FILE: CURIC

```
SUBROUTINE CURIC (XX,YY,XMIN2)
DOUBLE PRECISION
1 XX(4) ,YY(4) ,ARRAY(4,4) ,COEFF(4) ,CONST(4)
2 TEMP ,ADD ,F ,XMIN2
DO 14 I = 1,4
  ARRAY(I,1) = 1.000
  ARRAY(I,2) = XX(I)
  ARRAY(I,3) = XX(I)**2
  ARRAY(I,4) = XX(I)**3
14 CONST(I) = YY(I)
15 DO 6 K2 = 1,3
  KK = K2 + 1
  L = K2
  DO 7 I = KK,4
    IF (DABS(ARRAY(I,K2)) - DABS(ARRAY(L,K2))) > 7.7,20
20 CONTINUE
  L = I
  7 CONTINUE
  IF (L - K2) < 8.8,21
21 CONTINUE
  DO 9 J = K2,4
    TEMP = ARRAY(K2,J)
    ARRAY(K2,J) = ARRAY(L,J)
  9 ARRAY(L,J) = TEMP
  TEMP = CONST(K2)
  CONST(K2) = CONST(L)
  CONST(L) = TEMP
  A DO 6 I = KK,4
    F = ARRAY(I,K2)/ARRAY(K2,K2)
    ARRAY(I,K2) = 0.000
  DO 10 J = KK,4
10 ARRAY(I,J) = ARRAY(I,J) - F*ARRAY(K2,J)
  6 CONST(I) = CONST(I) - F*CONST(K2)
  COEFF(4) = CONST(4)/ARRAY(4,4)
  I = 3
11 II = I + 1
  ADD = 0.000
  DO 12 J = II,4
12 ADD = ADD + ARRAY(I,J)*COEFF(J)
  COEFF(I) = (CONST(I) - ADD)/ARRAY(I,I)
  I = I - 1
  IF (I) < 22.11
22 CONTINUE
  F = COEFF(3)**2 - 3.000*COEFF(2)*COEFF(4)
  IF (F) < 16.23,23
23 CONTINUE
  IF (DABS(COEFF(4)) - 1.00-15) < 17.17,24
24 CONTINUE
  XMIN2 = (-COEFF(3) + DSQRT(F))/(3.000*COEFF(4))
  RETURN
17 IF (DABS(COEFF(3)) - 1.00-15) < 16.16,25
25 CONTINUE
  XMIN2 = -COEFF(2)/(COEFF(3) + DSQRT(F))
  RETURN
16 XMIN2 = 0.000
  RETURN
END
```

CUB00010  
CUB00020  
CUB00030  
CUR00040  
CUR00050  
CUR00060  
CUR00070  
CUR00080  
CUR00090  
CUB00100  
CUB00110  
CUB00120  
CUB00130  
CUR00140  
CUR00150  
CUB00160  
CUR00170  
CUR00180  
CUR00190  
CUR00200  
CUR00210  
CUR00220  
CUR00230  
CUR00240  
CUR00250  
CUR00260  
CUR00270  
CUR00280  
CUR00290  
CUR00300  
CUR00310  
CUR00320  
CUR00330  
CUR00340  
CUR00350  
CUR00360  
CUR00370  
CUR00380  
CUR00390  
CUR00400  
CUR00410  
CUR00420  
CUR00430  
CUR00440  
CUR00450  
CUR00460  
CUR00470  
CUR00480  
CUR00490  
CUR00500  
CUR00510  
CUR00520  
CUR00530  
CUR00540  
CUR00550  
CUR00560  
CUR00570



FILE: DAVDN2

```
SUBROUTINE DAVDN2(XBAR,XBARS,P,WRK,IWKSZ,*)
DOUBLE PRECISION XPAR(N),XBARS(N),P(N),WRK(IWKSZ)
DOUBLE PRECISION FHAT(4),CAY(4),X(10),Y(10)
* CAYMIN,CCAY,DFDK,EPSCF,EPSCG,EX,F,FII,
* G1,G2,TEMP,XMIN1,XMIN2,XNF
COMMON/DVNHK/DFDK,CAYMIN,FII,CCAY,IID,IIDMEN,ITT,ICNT,N
DFDK***** MINUS THE NORM OF THE GRADIENT WITH RESPECT TO
C ***** THE H-MATRIX SQUARED
C ***** INITIAL VALUE OF FUNCTION TO BE MINIMIZED
C ***** SFT TO -1 SO PARTIALS WILL NOT BE COMPUTED
C ***** DURING SEARCH
C ***** NUMBER OF VARIABLES
C ***** ITT DENOTING THE NUMBER OF EVALUATIONS
C ***** OF THE FUNCTION TO BE MINIMIZED
C ***** VECTOR CORRESPONDING TO SEARCH DIRECTION
C ***** (H-MATRIX TIMES THE GRADIENT)
C ***** NOMINAL VECTOR OF CONTROL PARAMETERS AT
C ***** THE START OF EACH CYCLE
C ***** PERTURBED VECTOR OF CONTROL PARAMETERS IN
C ***** ACCORDANCE WITH THE DAVIDON ALGORITHM AND
C ***** SEARCH PROCEDURES
G1 = 1.500 - DSQRT(1.2500)
G2 = 1.000 - G1
C
C THIS STARTS ONE DIMENSIONAL SEARCH ON K
C
C INITIALIZE 1D SEARCH VARIABLES
C EPSCF IS RELATIVE EPSILON FOR CUBIC FIT.
C EPSCG IS RELATIVE EPSILON FOR GOLDEN SECTION.
C KUBIC = 1 CURVIC FIT METHOD
C KUBIC = 0 GOLDEN SECTION METHOD
C KUBIC = -1 GOLDEN SECTION AND CUBIC FIT METHODS COMBINED
C
9038 IPART=-1
Y(10) = FII
JSW = -1
KUBIC=1
EPSCF=1.0D-3
CAY(1)=0.0D+0
FHAT(1) = FII
CAY(2)=CCAY
DO 9039 I=1,N
9039 XBARS(I)=XBAR(I)+CAY(2)*P(I)
CALL FIN2(FHAT(2),IPART,XBARS,WRK,IWKSZ,69999)
Y(9) = FHAT(2)
X(9)=CAY(2)
ITT=ITT+1
Y(1)=2.0D+0*(FHAT(2)-FHAT(1)-DFDK*CAY(2))/(CAY(2)*CAY(2))
IF(Y(1) - 1.00D-15)9040,1,1
1 CONTINUE
XMIN1=-DFDK/Y(1)
CAY(3)=XMIN1
DO 9041 I=1,N
9041 XBARS(I)=XBAR(I)+CAY(3)*P(I)
CALL FIN2(FHAT(3),IPART,XBARS,WRK,IWKSZ,69999)
ITT=ITT+1
GO TO 9043
9040 CAY(3)=CCAY*1.20D+0
XMIN1=CAY(3)
GO TO 9042
9043 Y(1)=FHAT(2)-FHAT(1)-DFDK*CAY(2)
Y(2)=FHAT(3)-FHAT(1)-DFDK*CAY(3)
X(1)=CAY(2)*CAY(2)/2.0D+0
X(2)=X(1)*CAY(2)/3.0D+0
X(3)=CAY(3)*CAY(3)/2.0D+0
X(4)=X(3)*CAY(3)/3.0D+0
Y(3)=X(3)*X(2)-X(1)*X(4)
Y(4)=Y(3)
IF(Y(3) - 1.0D+0)Y(4)=-Y(3)
XNF = Y(4)
Y(4)=(X(2)*Y(2)-X(4)*Y(1))/Y(3)
Y(5)=(X(3)*Y(1)-X(1)*Y(2))/Y(3)
Y(6)=Y(4)*Y(4)-2.0D+0*DFDK*Y(5)
IF(Y(6))9044,2,2
2 CONTINUE
XMIN2 = -2.000*DFDK/(Y(4) + DSQRT(Y(6)))
IF(XMIN2 - 1.00D-15)9044,3,3
3 CONTINUE
IF(DABS(XMIN2-XMIN1) - EPSCF*XMIN1)4,4,9045
DAV00010
DAV00020
DAV00030
DAV00040
DAV00050
DAV00060
DAV00070
DAV00080
DAV00090
DAV00100
DAV00110
DAV00120
DAV00130
DAV00140
DAV00150
DAV00160
DAV00170
DAV00180
DAV00190
DAV00200
DAV00210
DAV00220
DAV00230
DAV00240
DAV00250
DAV00260
DAV00270
DAV00280
DAV00290
DAV00300
DAV00310
DAV00320
DAV00330
DAV00340
DAV00350
DAV00360
DAV00370
DAV00380
DAV00390
DAV00400
DAV00410
DAV00420
DAV00430
DAV00440
DAV00450
DAV00460
DAV00470
DAV00480
DAV00490
DAV00500
DAV00510
DAV00520
DAV00530
DAV00540
DAV00550
DAV00560
DAV00570
DAV00580
DAV00590
DAV00600
DAV00610
DAV00620
DAV00630
DAV00640
DAV00650
DAV00660
DAV00670
DAV00680
DAV00690
DAV00700
DAV00710
DAV00720
DAV00730
DAV00740
DAV00750
DAV00760
DAV00770
DAV00780
DAV00790
```

FILE: DAVDN2

```
4 CONTINUE
  CAYMIN=XMIN2
  GO TO 9045
9045 CAY(4)=XMIN2
  GO TO 9048
9044 CAY(4)=CAY(3)*1.50+0
9048 DO 9047 I=1,N
9047 XBAR(I)=XBAR(I)+CAY(4)*P(I)
  CALL FINT2(FHAT(4),IPART,XBAR,WRK,IWKSZ,&9999)
  ITT=ITT+1
  DO 9050 J=1,3
  DO 9050 I=1,3
  IF(CAY(I) - CAY(I+1))9050,5,5
5 CONTINUE
  TEMP = CAY(I)
  CAY(I) = CAY(I+1)
  CAY(I+1) = TEMP
  TEMP = FHAT(I)
  FHAT(I) = FHAT(I+1)
  FHAT(I+1) = TEMP
9050 CONTINUE
  GO TO 542
9066 FII = Y(10)
  CAY(4)=X(9)
  FHAT(4)=Y(9)
  IJDMEN = -1
  R01 IPART=-1
  KURJC = 1
  IF(I10)888,888,6
6 CONTINUE
  KURIC = 0
  R88 XMIN1 = 0.000
  CAY(1)=0.0+0
  FHAT(1) = FII
  IF(DFDK .GT. -1.0-12)GO TO 340
  IF(DFDK .GT. -1.0-9) GO TO 330
  IF(DFDK .GT. -1.0-5) GO TO 320
  IF(DFDK .GT. -1.0+1)GO TO 350
  EPSGS=1.0-2
  FPSCF=1.0-2
  GO TO 410
350 FPSCF=1.0-3
  EPSGS=1.0-3
  GO TO 410
320 FPSCF=1.0-5
  EPSGS=1.0-5
  GO TO 410
330 EPSCF = 1.0-2
  FPSCF = 1.0-2
  GO TO 410
340 EPSCF = 1.0-1
  FPSCF = 1.0-1
C
C
C PART 1
  ESTABLISH GOLDEN SECTION IN WHICH FUNCTION IS UNIMODAL
410 IF(I10)7,7,9067
7 CONTINUE
  CAY(4)=CCAY
  DO 430 I=1,N
430 XBAR(I)=XBAR(I)+CAY(4)*P(I)
  CALL FINT2(FHAT(4),IPART,XBAR,WRK,IWKSZ,&9999)
  ITT = ITT + 1
9067 CONTINUE
  IF(FHAT(4) - FHAT(1))R,490,490
  R CONTINUE
440 CAY(2)=CAY(4)
  FHAT(2)=FHAT(4)
  CAY(4) = CAY(2)/G1
  DO 450 I=1,N
450 XBAR(I)=XBAR(I)+CAY(4)*P(I)
  CALL FINT2(FHAT(4),IPART,XBAR,WRK,IWKSZ,&9999)
  ITT = ITT + 1
  IF(FHAT(4) - FHAT(2))440,9,9
9 CONTINUE
460 CAY(3) = G2*(CAY(4) - CAY(1)) + CAY(1)
  DO 470 I=1,N
470 XBAR(I)=XBAR(I)+CAY(3)*P(I)
  CALL FINT2(FHAT(3),IPART,XBAR,WRK,IWKSZ,&9999)
```

DAV00800  
DAV00810  
DAV00820  
DAV00830  
DAV00840  
DAV00850  
DAV00860  
DAV00870  
DAV00880  
DAV00890  
DAV00900  
DAV00910  
DAV00920  
DAV00930  
DAV00940  
DAV00950  
DAV00960  
DAV00970  
DAV00980  
DAV00990  
DAV01000  
DAV01010  
DAV01020  
DAV01030  
DAV01040  
DAV01050  
DAV01060  
DAV01070  
DAV01080  
DAV01090  
DAV01100  
DAV01110  
DAV01120  
DAV01130  
DAV01140  
DAV01150  
DAV01160  
DAV01170  
DAV01180  
DAV01190  
DAV01200  
DAV01210  
DAV01220  
DAV01230  
DAV01240  
DAV01250  
DAV01260  
DAV01270  
DAV01280  
DAV01290  
DAV01300  
DAV01310  
DAV01320  
DAV01330  
DAV01340  
DAV01350  
DAV01360  
DAV01370  
DAV01380  
DAV01390  
DAV01400  
DAV01410  
DAV01420  
DAV01430  
DAV01440  
DAV01450  
DAV01460  
DAV01470  
DAV01480  
DAV01490  
DAV01500  
DAV01510  
DAV01520  
DAV01530  
DAV01540  
DAV01550  
DAV01560  
DAV01570  
DAV01580



FILE: DAVON2

```
      ITT = ITT + 1
      IF (FHAT(4) - FHAT(3))10,10,510
10 CONTINUE
475 FHAT(4)=FHAT(3)
      CAY(4)=CAY(3)
      CAY(3)=CAY(2)
      FHAT(3)=FHAT(2)
      CAY(2)=CAY(4) - CAY(3) + CAY(1)
      DO 480 I=1,N
480 XBARS(I)=XBAR(I)+CAY(2)*P(I)
      CALL FINT2(FHAT(2),IPART,XBARS,WKK,IWKSZ,&9999)
      ITT = ITT + 1
      IF (FHAT(2) - FHAT(1))510,11,11
11 CONTINUE
485 FHAT(4) = FHAT(2)
      CAY(4) = CAY(2)
490 IF (IDAVN.EQ.2) GO TO R007
      CAY(2) = G1*(CAY(4) - CAY(1)) + CAY(1)
      IF (CAY(2) - 1.0D-15)12,12,495
12 CONTINUE
R007 CAYMIN=0.0+0
      WRITE (6,492)
492 FORMAT(1X,46HMINIMUM IS AT ORIGIN - PROGRAM CANNOT CONTINUE)
      RETURN
495 DO 500 I=1,N
500 XBARS(I)=XBAR(I)+CAY(2)*P(I)
      CALL FINT2(FHAT(2),IPART,XBARS,WKK,IWKSZ,&9999)
      ITT = ITT + 1
      IF (FHAT(2) .LT. FHAT(1))GO TO 460
      GO TO 485

CCCC
PART 2
      SHPINK THE GOLDEN SECTION CONTAINING THE MINIMUM
510 IF (I1DMFN)13,13,9066
13 CONTINUE
514 IF (FHAT(3) - FHAT(2))14,525,520
14 CONTINUE
      CAY(1)=CAY(2)
      FHAT(1)=FHAT(2)
      CAY(2)=CAY(3)
      FHAT(2)=FHAT(3)
      CAY(3)=CAY(1)+CAY(4)-CAY(2)
      FX = CAY(3)
      JSW = 1
      GO TO 612
520 CAY(4) = CAY(3)
      CAY(4)=CAY(3)
      FHAT(4)=FHAT(3)
      CAY(3)=CAY(2)
      FHAT(3)=FHAT(2)
      CAY(2)=CAY(1)+CAY(4)-CAY(3)
      FX = CAY(2)
      JSW = -1
      GO TO 612
525 FHAT(1) = FHAT(2)
      FHAT(1)=FHAT(2)
      CAY(4)=CAY(3)
      FHAT(4)=FHAT(3)
      CAY(3) = 3.0D+0*CAY(2) - CAY(3) - CAY(1)
      CAY(1)=CAY(2)
      CAY(2) = CAY(4) + CAY(1) - CAY(3)
      FX = CAY(2)
      JSW = 0
      GO TO 612
526 FHAT(3)=F

CCCC
CHECK ON THE UNIMODALITY OF THE FUNCTION IN THE NEW INTERVAL
530 IF (FHAT(2) - FHAT(1))15,15,485
15 CONTINUE
      IF (FHAT(3) - FHAT(4))16,16,475
16 CONTINUE

CCCC
PART 3
      FIND MINIMUM BY EITHER GOLDEN SECTION OR CUBIC FIT TECHNIQUE
540 IF (CAY(3) - CAY(2)*(1.00D+0*EPSGS))585,17,17
17 CONTINUE
```

DAV01590  
DAV01600  
DAV01610  
DAV01620  
DAV01630  
DAV01640  
DAV01650  
DAV01660  
DAV01670  
DAV01680  
DAV01690  
DAV01700  
DAV01710  
DAV01720  
DAV01730  
DAV01740  
DAV01750  
DAV01760  
DAV01770  
DAV01780  
DAV01790  
DAV01800  
DAV01810  
DAV01820  
DAV01830  
DAV01840  
DAV01850  
DAV01860  
DAV01870  
DAV01880  
DAV01890  
DAV01900  
DAV01910  
DAV01920  
DAV01930  
DAV01940  
DAV01950  
DAV01960  
DAV01970  
DAV01980  
DAV01990  
DAV02000  
DAV02010  
DAV02020  
DAV02030  
DAV02040  
DAV02050  
DAV02060  
DAV02070  
DAV02080  
DAV02090  
DAV02100  
DAV02110  
DAV02120  
DAV02130  
DAV02140  
DAV02150  
DAV02160  
DAV02170  
DAV02180  
DAV02190  
DAV02200  
DAV02210  
DAV02220  
DAV02230  
DAV02240  
DAV02250  
DAV02260  
DAV02270  
DAV02280  
DAV02290  
DAV02300  
DAV02310  
DAV02320  
DAV02330  
DAV02340  
DAV02350  
DAV02360  
DAV02370

FILE: DAV0N2

```
IF (KURIC.EQ.0) GO TO 514
IF (CAY(4) - 1.00-5) 510,542,542
542 CALL CURIC(CAY,FHAT,XMIN2)
IF (XMIN2) 510,510,18
18 CONTINUE
IF (KURIC.GT.0) GO TO 2000
IF (XMIN1) 580,580,19
19 CONTINUE
IF (XMIN2*(1.000-EPSCF) - XMIN1) 20,20,580
20 CONTINUE
545 CAYMIN = (XMIN2 + XMIN1)/2.00+0
GO TO 9999
540 XMIN1 = XMIN2
GO TO 510
545 CAYMIN=(CAY(3)+CAY(2))/2.0+0
GO TO 9999
C
612 DO 613 I=1,N
613 XHARS(I) = XHAR(I) + EX*P(I)
CALL FINT2(F,IPART,XHARS,WRK,IWKSZ,69999)
ITT = ITT + 1
IF (JSW) 21,21,536
21 CONTINUE
FHAT(2) = F
IF (JSW) 540,22,22
22 CONTINUE
FX = CAY(3)
JSW = 1
GO TO 612
C
REFARRANGE Y(I) SO THEY ARE IN ASCENDING ORDER
C
2000 DO 2020 I = 1,4
X(I) = CAY(I)
2020 Y(I) = FHAT(I)
DO 2040 J = 1,3
DO 2040 I = 1,3
IF (Y(I) - Y(I+1)) 2040,23,23
23 CONTINUE
TEMP = Y(I)
Y(I) = Y(I+1)
Y(I+1) = TEMP
TEMP = X(I)
X(I) = X(I+1)
X(I+1) = TEMP
2040 CONTINUE
3000 DO 3001 I=1,N
3001 XHARS(I) = XHAR(I) + XMIN2 * P(I)
CALL FINT2(F,IPART,XHARS,WRK,IWKSZ,69999)
ITT = ITT + 1
3005 DO 3010 K = 1,4
J = K
IF (F.LT.Y(K)) GO TO 3020
3010 CONTINUE
GO TO 4000
3020 I = 4
3025 IF (I.EQ.0) GO TO 3030
Y(I) = Y(I-1)
X(I) = X(I-1)
I = I-1
GO TO 3025
3030 Y(I) = F
X(I) = XMIN2
3045 CALL CURIC(X,Y,XMIN2)
3050 IF (XMIN2) 24,4000,24
24 CONTINUE
IF (DARS(XMIN2-X(I)) - EPSCF*XMIN2) 25,25,3000
25 CONTINUE
CAYMIN = XMIN2
GO TO 9999
4000 IF (I10MEN) 26,26,9066
26 CONTINUE
KURIC = -1
XMIN1 = 0.00+0
XMIN2 = 0.00+0
GO TO 514
9999 CAY=CAYMIN
I10MEN=110
C
```

DAV02380  
DAV02390  
DAV02400  
DAV02410  
DAV02420  
DAV02430  
DAV02440  
DAV02450  
DAV02460  
DAV02470  
DAV02480  
DAV02490  
DAV02500  
DAV02510  
DAV02520  
DAV02530  
DAV02540  
DAV02550  
DAV02560  
DAV02570  
DAV02580  
DAV02590  
DAV02600  
DAV02610  
DAV02620  
DAV02630  
DAV02640  
DAV02650  
DAV02660  
DAV02670  
DAV02680  
DAV02690  
DAV02700  
DAV02710  
DAV02720  
DAV02730  
DAV02740  
DAV02750  
DAV02760  
DAV02770  
DAV02780  
DAV02790  
DAV02800  
DAV02810  
DAV02820  
DAV02830  
DAV02840  
DAV02850  
DAV02860  
DAV02870  
DAV02880  
DAV02890  
DAV02900  
DAV02910  
DAV02920  
DAV02930  
DAV02940  
DAV02950  
DAV02960  
DAV02970  
DAV02980  
DAV02990  
DAV03000  
DAV03010  
DAV03020  
DAV03030  
DAV03040  
DAV03050  
DAV03060  
DAV03070  
DAV03080  
DAV03090  
DAV03100  
DAV03110  
DAV03120  
DAV03130  
DAV03140  
DAV03150  
DAV03160

FILE: DAVDN2

```
C THIS ENDS ONE DIMENSIONAL SEARCH ON K  
C  
30 DO 31 I=1,N  
P(I)=CAYMIN*P(I)  
XRAR(I)=XBAR(I)*P(I)  
31 CONTINUE  
RETURN  
END
```

DAV03170  
DAV03180  
DAV03190  
DAV03200  
DAV03210  
DAV03220  
DAV03230  
DAV03240

FILE: DAVDN3

```

C* SURROUTINE DAVDN3(FX,FX1,P,H,MY)
C* DAVDN3 - UPDATES THE H MATRIX AND THE P MATRIX FOR THE NEXT
C* CYCLE THROUGH THE SEARCH.
C* - THE H MATRIX IS BROUGHT IN TO CORE ONE ROW AT A TIME.
C* SCRATCH FILES 7 AND 28 ARE USED TO STORE AND UPDATE
C* THIS MATRIX.
C* N***** - NUMBER OF VARIABLES
C* FX***** - GRADIENT OF FUNCTION BEING MINIMIZED
C* (PARTIAL DER. WITH RESPECT TO NEW H-MATRIX)
C* FX1*** - VALUE OF GRADIENT AT THE BEGINNING OF THE CYCLE.
C* (PARTIAL DER. WITH RESPECT TO OLD H-MATRIX)
C* H***** - THE H MATRIX APPROXIMATES THE INVERSE MATRIX OF
C* PARTIAL DERIVATIVES.
C* HY***** - VECTOR USED IN UPDATING THE H-MATRIX
C* P***** - VECTOR CORRESPONDING TO SEARCH DIRECTION
C* (H-MATRIX TIMES THE GRADIENT)
C*
C* INCLUDE COMP7.LIST
C* COMMON/FS1/CFAC,TOTMSR,SEPMSR,PRCKEY,CRKIFY,INCFET,
C* INCVFC(20),ICOUNT,SFTWGT,EVALRF(100),FETVC4(30)
C* ,NOFET4,VARS74,CORHAS,OTAR4,NGHS14,RESTVC(10),DIVSIZ
C* ,STATKY,ADRES0,ADRES1,ADRESF,ADRS1,ADRS2
C* ,INTEGER ADRES0,ADRES1,ADRESF,ADRS1,ADRS2,STATKY
C* DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C*
C* CSENO
C* DOUBLE PRECISION DFDK,CAYMIN,CCAY,FII
C* DOUBLE PRECISION SIGYI,YHY,DELXFX
C* DOUBLE PRECISION FX(N),FX1(N),P(N),H(N),MY(N)
C* COMMON/DVNHK/DFDK,CAYMIN,FII,CCAY,IID,IIDMEN,ITT,ICNT,N
C* N2=N*2
C* KAD=ADRS1
C* LAD=ADRS2
C* IF(MOD(ICNT,2).EQ.1)GO TO 10
C* KAD=ADRS2
C* LAD=ADRS1
C* 10 CONTINUE
C* CALL RRFAD(ADRESF,FX1,N2,ISTAT)
C* UNIVAC CHECK DRUM STATUS
C* DELXFX=0.0
C* JAD=KAD
C* DO 40 I=1,N
C* DELXFX=DELXFX + P(I)*FX(I)/CAYMIN
C* HY(I)=0.0
C* READ ONE ROW OF THE H MATRIX
C* CALL RRFAD(JAD,H,N2,ISTAT)
C* UNIVAC CHECK DRUM STATUS
C* JAD=JAD+N2
C* 20 DO 30 J=1,N
C* 30 HY(I)=HY(I) + H(J)*(FX(J)-FX1(J))
C* 40 CONTINUE
C* YHY=0.0
C* SIGYI=0.0
C* DO 50 I=1,N
C* SIGYI=SIGYI + P(I)*(FX(I)-FX1(I))
C* 50 YHY=YHY + (FX(I)-FX1(I))*HY(I)
C* IF(SIGYI.LT.1.0-36 .OR. YHY .LT. 1.0-36)GO TO 80
C* DO 65 I=1,N
C* FX1(I)=0.0
C* CALL RRFAD(KAD,H,N2,ISTAT)
C* UNIVAC CHECK DRUM STATUS
C* KAD=KAD+N2
C* UPDATE THIS ROW OF THE H MATRIX
C* 55 DO 60 J=1,N
C* H(J) = H(J) + P(I)*P(J)/SIGYI - HY(I)*HY(J)/YHY
C* 60 CONTINUE
C* DO 61 J=1,N
C* USE FX1 STORAGE TO UPDATE P ARRAY
C* FX1(I)=FX1(I) - H(J)*FX(J)
C* 61 CONTINUE
C* WRITE UPDATED H
C* CALL RWRITE(LAD,H,N2,ISTAT)
C* LAD=LAD+12
C* UNIVAC CHECK DRUM STATUS
C* 65 CONTINUE
C*
C* DAV00010
C* DAV00020
C* DAV00030
C* DAV00040
C* DAV00050
C* DAV00060
C* DAV00070
C* DAV00080
C* DAV00090
C* DAV00100
C* DAV00110
C* DAV00120
C* DAV00130
C* DAV00140
C* DAV00150
C* DAV00160
C* DAV00170
C* DAV00180
C* DAV00190
C* DAV00200
C* DAV00210
C* DAV00220
C* DAV00230
C* DAV00240
C* DAV00250
C* DAV00260
C* DAV00270
C* DAV00280
C* DAV00290
C* DAV00300
C* DAV00310
C* DAV00320
C* DAV00330
C* DAV00340
C* DAV00350
C* DAV00360
C* DAV00370
C* DAV00380
C* DAV00390
C* DAV00400
C* DAV00410
C* DAV00420
C* DAV00430
C* DAV00440
C* DAV00450
C* DAV00460
C* DAV00470
C* DAV00480
C* DAV00490
C* DAV00500
C* DAV00510
C* DAV00520
C* DAV00530
C* DAV00540
C* DAV00550
C* DAV00560
C* DAV00570
C* DAV00580
C* DAV00590
C* DAV00600
C* DAV00610
C* DAV00620
C* DAV00630
C* DAV00640
C* DAV00650
C* DAV00660
C* DAV00670
C* DAV00680
C* DAV00690
C* DAV00700
C* DAV00710
C* DAV00720
C* DAV00730
C* DAV00740
C* DAV00750
C* DAV00760
C* DAV00770
C* DAV00780
C* DAV00790

```

FILE: DAVDN3

```

      DFDK=0.0
      DO 70 I=1,N
      P(I)=FY(I)
C* 70 DFDK=DFDK+FX(I)*P(I)
C*
C* SAVE LATEST PARTIALS ON SCRATCH FILE
C*
C* CALL RWRITE(ADRESF,FX,N2,ISTAT)
C UNIVAC CHECK DRUM STATUS
C RETURN
C 80 WRITE(A,300)SIGYI,YHY
C RETURN
C*
C* 300 FORMAT(' *DAVDN3*--EITHER SIGYI OR YHY TOO CLOSE TO ZERO TO UPDATE
C * M MATRIX--SIGYI=*,E15.7,* YHY=*,E15.7)
C UNIVAC CHECK DRUM STATUS
C END
```

DAV00800  
DAV00810  
DAV00820  
DAV00830  
DAV00840  
DAV00850  
DAV00860  
DAV00870  
DAV00880  
DAV00890  
DAV00900  
DAV00910  
DAV00920  
DAV00930  
DAV00940  
DAV00950  
DAV00960

FILE: DAVIDN

```

      SURROUTINE DAVIDN(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,AVEMT2,S,S2, DAV00010
      HMAT,PARTLS,WKRY,IWKSZ) DAV00020
..... DAV00030
C..... DAV00040
C..... DAV00050
C..... DAV00060
      DAVIDN IS THE DRIVER ROUTINE FOR THE DAVIDON-FEETCHER-POWELL PRO DAV00070
      INCLUDE COMK1.LIST DAV00080
      INCLUDE COM-K7.LIST DAV00090
      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2, DAV00100
      AVA22,COVAR2,CLSI02,SUMN02,SURDS2,FLUSV2,VENTX2, DAV00110
      FETVC2(30),SUMVC2(75),SURPTH(75),CLSVC2(60), DAV00120
      KFPTS(60),NOGRP,GRPNAM(60),GRPDEX(61), DAV00130
      GRPCHK(61),GROUPS(124) DAV00140
      COMMON/FSL/CFAC,TOTMSH,SEPM5H,PKKEY,CRIKEY,INCFET, DAV00150
      INVEC(30),ICOUNT,SETWGT,EVALHF(100),FETVC4(30) DAV00160
      NOFET4,VARSZ4,COHBS,UTAH4,WGHS14,HESTVC(10),DIVSIZ DAV00170
      ,STATKY,ADHESD,ADHESP,ADHESF,ADHSH1,ADHSH2 DAV00180
      INTEGER ADHESD,ADHESP,ADHESF,ADHSH1,ADHSH2,STATKY DAV00190
      DOUBLE PRECISION CFAC,TOTMSH,SEPM5H DAV00200
CSEND DAV00210
      EQUIVALENCE (IRLOCK(2),PKKEY) DAV00220
      DOUBLE PRECISION HMAT(NOFET4,NOFET2) DAV00230
      DIMENSION WKRY(1) DAV00240
      DIMENSION COVMTX(1), AVEMTX(1), WEIGHT(1), S(1) DAV00250
      INTEGER FETVC2,FETVC4,BMKEY,CRIKEY,PKKEY DAV00260
      INTEGER SETWGT DAV00270
      COMMON/DVNRK/DFDK,CAYMIN,FII,CCAY,I10,I10MEN,ITT,ICNT,N DAV00280
      DOUBLE PRECISION DELF,FII,CAYMIN,DFDK,CCAY,PARTLS(1),SMSR DAV00290
      DOUBLE PRECISION DIVTAB(1),COVMT2(1),AVEMT2(1),S2(1) DAV00300
      CCAY=1.D-6 DAV00310
      I10=1 DAV00320
      I10MEN=1 DAV00330
C* SET CONVERGENCE TOLERANCE DAV00340
      DELF=1.D-6 DAV00350
      IF(CRIKEY.EQ.1.AND.SETWGT.EQ.2) DELF=1.D-1 DAV00360
      IF(CRIKEY.EQ.3) DELF=1.D-5 DAV00370
      ICNT=0 DAV00380
C* DAV00390
C* INITIALIZE H-MATRIX (XBAR) DAV00400
C* DAV00410
C* IF (BMKEY.LE.0) GO TO 10 DAV00420
C* READ H-MATRIX IN SINGLE PRECISION, THEN STORE IN DOUBLE PRECISION DAV00430
C* DAV00440
      CALL HMFIL(WKRY,NOFET4,NOFET2,FETVC2,2) DAV00450
      IK=0 DAV00460
      DO 5 I=1,NOFET2 DAV00470
      DO 5 J=1,NOFET4 DAV00480
      IK=IK+1 DAV00490
      5 HMAT(J,I)=WKRY(IK) DAV00500
      GO TO 20 DAV00510
C* DAV00520
C* INITIALIZE H-MATRIX FROM 'HEST' SET FOUND IN WITHOUT REPLACEMENT P DAV00530
C* DAV00540
      10 CALL ORDER(FETVC4,NOFET4) DAV00550
      DO 15 I=1,NOFET4 DAV00560
      DO 15 J=1,NOFET2 DAV00570
      HMAT(I,J)=0.0 DAV00580
      15 IF(FETVC4(I).EQ.FETVC2(J)) HMAT(I,J)=1.0 DAV00590
C* DAV00600
C* INITIALIZE ROUTINE FOR EVALUATING PARTIALS AND SEP. MEASURE DAV00610
C* DAV00620
      20 CALL FINI(COVMTX,AVEMTX,DIVTAB,WEIGHT,S,S2,COVMT2,AVEMT2,PARTLS) DAV00630
      ITT=0 DAV00640
C* DAV00650
C* COMPUTE PARTIALS FOR INITIAL H-MATRIX DAV00660
C* DAV00670
      IPART=1 DAV00680
      CALL FINI2(SMSR,IPART,HMAT,WKRY,IWKSZ,N2) DAV00690
      CONTINUE DAV00700
      25 ITT=ITT+1 DAV00710
      FII=SMSR DAV00720
      N=NOFET2*NOFET4 DAV00730
      LEFT=IWKSZ-N*2 DAV00740
C* DAV00750
C* IP=BASE ADDRESS FOR P ARRAY DAV00760
C* IF=BASE ADDRESS FOR H ARRAY (ONLY ONE ROW OF H IN CORE AT A TIME) DAV00770
C* IHY=BASE ADDRESS FOR HY ARRAY DAV00780
C* DAV00790

```

FILE: DAVIDN

```
IP=1
IH=IP*N2
IHY=IH*N2
IF1=IHY*N2
ITEST=IF1*N2
IF(ITEST.LE.IWRKSZ)GO TO 30
WRITE(6,100)IWRKSZ
CALL CMERR
30 CONTINUE
C*
C* INITIALIZE H AND P ARRAYS - SAVE H AND PARTLS ON SCRATCH FILE
C* ALSO COMPUTE INITIAL OFDK
C* CALL DAVD1(PARTLS,WRKRY(IP),WRKRY(IH))
C* PARTLS STORAGE CAN NOW BE USED IN DAVIDN2 FOR TEMP. STORAGE OF XRA
C* DAVD2 RETURNS A NEW H-MATRIX AND P ARRAY AND CAYMIN
40 CALL DAVD2(BMAT,PARTLS,WRKRY(IP),WRKRY(IH),LEFT,650)
N2=2*N
C* SAVE P ARRAY ON SCRATCH FILE
CALL RWRITE(ADRES,WRKRY(IP),N2,LSTAT)
C* WAIT FOR I/O
41 IF(LSTAT.EQ.1)GO TO 41
IF(LSTAT.NE.0)GO TO 70
C* COMPUTE NEW PARTIALS
CALL FINTZ(SMSR,IPART,BMAT,WRKRY,IWRKSZ,645)
45 CONTINUE
ITT=ITT+1
ICNT=ICNT+1
SFPMSR=SMSR
IF(DARS(SMSR-FII).LT.DELF)GO TO 50
FII=SMSR
IF(ITT.GE.ICOUNT)GO TO 50
SMSR=SFPMSR
C* READ BACK P ARRAY
CALL PREAD(ADRES,WRKRY(IP),N2,LSTAT)
46 IF(LSTAT.EQ.1)GO TO 46
IF(LSTAT.NE.0)GO TO 70
C*
C* SUBROUTINE DAVD3 UPDATES THE H AND P ARRAYS AND SAVES NEW
C* H AND NEW PARTIALS ON SCRATCH FILE
C*
CALL DAVD3(PARTLS,WRKRY(IF1),WRKRY(IP),WRKRY(IH),WRKRY(IHY))
GO TO 40
C*
C* IF CRITERIA = AV. DIVERGENCE - COMPUTE INTERCLASS DIVERGENCES
C*
50 IF(CPIKEY.NE.1)GO TO 60
CALL DIVRG1(COVRT,VARSZ4,AVEMT2,DIVTAB,NOCLS2,NOFET4,
WRKRY,IWRKSZ)
60 CONTINUE
C*
C* STORE IN SINGLE PRECISION ARRAY AND
C* WRITE H-MATRIX ON FILE AND PUNCH ON CARDS IF REQUESTED
C*
IK=0
DO 65 I=1,NOFET2
DO 65 J=1,NOFET4
IK=IK+1
WRKRY(IK)=HMAT(J,I)
65 CONTINUE
HMKEY=1
CALL HMFIL(WRKRY,NOFET4,NOFET2,FETVC2,5)
C
IF(PCHKEY.EQ.1)CALL HMFIL(WRKRY,NOFET4,NOFET2,FETVC2,4)
RETURN
70 WRITE(6,200)LSTAT
100 FORMAT(' NOT ENOUGH WORK AREA AVAILABLE IN DAVIDN--IWRKSZ=',I6)
200 FORMAT(' ERROR ON DRUM FILE - SUBROUTINE DAVIDN---LSTAT=',I3)
END
```

DAV00800  
DAV00810  
DAV00820  
DAV00830  
DAV00840  
DAV00850  
DAV00860  
DAV00870  
DAV00880  
DAV00890  
DAV00900  
DAV00910  
DAV00920  
DAV00930  
DAV00940  
DAV00950  
DAV00960  
DAV00970  
DAV00980  
DAV00990  
DAV01000  
DAV01010  
DAV01020  
DAV01030  
DAV01040  
DAV01050  
DAV01060  
DAV01070  
DAV01080  
DAV01090  
DAV01100  
DAV01110  
DAV01120  
DAV01130  
DAV01140  
DAV01150  
DAV01160  
DAV01170  
DAV01180  
DAV01190  
DAV01200  
DAV01210  
DAV01220  
DAV01230  
DAV01240  
DAV01250  
DAV01260  
DAV01270  
DAV01280  
DAV01290  
DAV01300  
DAV01310  
DAV01320  
DAV01330  
DAV01340  
DAV01350  
DAV01360  
DAV01370  
DAV01380  
DAV01390  
DAV01400  
DAV01410  
DAV01420  
DAV01430  
DAV01440  
DAV01450  
DAV01460

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: DIVERG

```
      SUBROUTINE DIVERG(COVMTX,VARSIZ,AVEMTX,DIVTAB,NOCLS,
      * NOFET,WRKRY,IWRKSZ)
C*
C* SURROUTINE TO COMPUTE INTERCLASS DIVERGENCES
C*
      INTEGER VARSIZ
      DOUBLE PRECISION WRKRY(1),T
      DOUBLE PRECISION DIVTAB,DET,TRACE
      DIMENSION COVMTX(VARSIZ,NOCLS), AVEMTX(NOFET,NOCLS),
      * DIVTAB(1),T(30)
      INP=1
      GO TO 3
      ENTRY DIVERG(COVMT2,VARSIZ,AVEMT2,DIVTAB,NOCLS,NOFET,WRKRY,IWRKSZ)
      DOUBLE PRECISION COVMT2(VARSIZ,NOCLS),AVEMT2(NOFET,NOCLS)
      INP=0
3 CONTINUE
      ICV1=1
      ICV2=ICV1+VARSIZ
      IW1=ICV2+VARSIZ
      IW2=IW1+VARSIZ
      IF(IWRKSZ/2.GE.IW2+VARSIZ)GO TO 4
      WRITE(6,200)IWRKSZ
      CALL CMEWR
4 CONTINUE
      MN=0
      IC=NOCLS-1
      DO 30 I=1,IC
C* FIND INVERSE FOR CLASS I COVAR. MATRIX
      DO 1 IK=1,VARSIZ
      IF(INP.EQ.1)WRKRY(IK)=COVMTX(IK,I)
      IF(INP.EQ.0)WRKRY(IK)=COVMT2(IK,I)
1 CONTINUE
      CALL COLINV(WRKRY(ICV1),NOFET,IERR,3,DET)
      IF(IERR.EQ.0)GO TO 2
      WRITE(6,100) I
      GO TO 30
2 IM=I+1
      DO 20 J=IM,NOCLS
      MN=MN+1
      DO 5 II=1,NOFET
      IF(INP.EQ.1)T(II)=AVEMTX(II,I)-AVEMTX(II,J)
      IF(INP.EQ.0)T(II)=AVEMT2(II,I)-AVEMT2(II,J)
5 CONTINUE
      K=0
      DO 10 II=1,NOFET
      DO 10 JJ=1,II
      K=K+1
      IF(INP.EQ.1)GO TO 6
      WRKRY(ICV2+K-1)=COVMT2(K,J)
      WRKRY(IW1+K-1) =COVMT2(K,J) + T(II)*T(JJ)
      WRKRY(IW2+K-1) =COVMT2(K,I) + T(II)*T(JJ)
      GO TO 10
6 CONTINUE
      WRKRY(ICV2+K-1)=COVMTX(K,J)
      WRKRY(IW1+K-1) =COVMTX(K,J) + T(II)*T(JJ)
      WRKRY(IW2+K-1)=COVMTX(K,I)+T(II)*T(JJ)
10 CONTINUE
C* FIND INVERSE FOR CLASS J COVAR. MATRIX
      CALL COLINV(WRKRY(ICV2),NOFET,IERR,3,DET)
      IF(IERR.EQ.0)GO TO 15
      WRITE(6,100)J
      GO TO 20
15 DIVTAB(MN)=.5*(TRACE(WRKRY(ICV1),WRKRY(IW1),NOFET)
      * TRACE(WRKRY(ICV2),WRKRY(IW2),NOFET)) - NOFET
20 CONTINUE
30 CONTINUE
      RETURN
100 FORMAT(' COVAR FOR CLASS',I4,' IS NOT POSITIVE DEFINITE')
200 FORMAT(' NOT ENOUGH WORK AREA AVAILABLE IN DIVERG -- IWRKSZ=',I5)
      END
```



FILE: EVALSP

```
      SUBROUTINE EVALSP( SMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,
      DIVTAB,WEIGHT,IPART,PARTLS,BMAT,WKRY,IWKSZ)
C*
C* THIS SUBROUTINE COORDINATES THE ROUTINES FOR COMPUTING THE
C* SEPARABILITY MEASURE FOR A PARTICULAR LINEAR COMBINATION OR
C* SET OF FEATURES
C*
      DOUBLE PRECISION SMSR,COVMT2(1),AVEMT2(1),S2(1),DIVTAB(1)
      PARTLS(1),HMAT(1)
      INTEGER CRIKEY
      INCLUDE COMRK1,LIST
      INCLUDE COMRK7,LIST
      COMMON/INFORM/NOCLS2,NOSUR2,NOFFI2,VAR52,TOTVT2,NOFLD2,
      AVAR2,COVAR2,CLSID2,SURNO2,SURDS2,FLUSV2,VENTX2,
      FETVC2(30),SUNVC2(75),SUBPTH(75),CLSV2(60),
      KFPPTS(60),NOGMP,GMPNAM(60),GRPDEX(61),
      GRPCHK(61),GROUPS(124)
      COMMON/FSL/CFAC,TOTMSR,SEPMSP,PRCKEY,CRIKEY,INCFET,
      INCVEC(30),ICOUNT,SFTWGT,EVALRF(100),FETVC4(30)
      NOFFI4,VAR574,COH45,DTA4,WGHS14,BESTVC(10),DIVSIZ
      STATKY,ADRES0,ADRES1,ADRESF,ADRSH1,ADRSH2
      INTEGER ADRES0,ADRES1,ADRESF,ADRSH1,ADRSH2,STATKY
      DOUBLE PRECISION CFAC,TOTMSR,SEPMSP
CSEND
      DIMENSION COVMTX(1), AVEMTX(1), S(1),
      WEIGHT(1),IPART(1),WKRY(1)
      IFULL=0
      GO TO(10,20,30,40),CRIKEY
C* WEIGHTED AVERAGE DIVERGENCE
      10 CALL AVEDIV(SMSR,COVMTX,S,COVMT2,S2,WKRY,IWKSZ,
      IPART,PARTLS,BMAT,IFULL)
      RETURN
C*
C* WEIGHTED AVERAGE TRANSFORMED DIVERGENCE
C*
      20 CALL TRNDIV(SMSR,COVMTX,AVEMTX,COVMT2,AVEMT2,
      WEIGHT,DIVTAB,WKRY,
      IWKSZ,IPART,PARTLS,BMAT,IFULL)
      RETURN
C*
C* WEIGHTED AVERAGE BHATTACHARYYA DISTANCE
C*
      30 CALL BHCHP(SMSR,COVMTX,AVEMTX,WEIGHT,DIVTAB,
      COVMT2,AVEMT2,WKRY,
      IWKSZ,IPART,PARTLS,BMAT,IFULL)
      40 CONTINUE
      RETURN
      END
```

EVA00010  
EVA00020  
EVA00030  
EVA00040  
EVA00050  
EVA00060  
EVA00070  
EVA00080  
EVA00090  
EVA00100  
EVA00110  
EVA00120  
EVA00130  
EVA00140  
EVA00150  
EVA00160  
EVA00170  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00060  
EVA00250  
EVA00260  
EVA00270  
EVA00280  
EVA00290  
EVA00300  
EVA00310  
EVA00320  
EVA00330  
EVA00340  
EVA00350  
EVA00360  
EVA00370  
EVA00380  
EVA00390  
EVA00400  
EVA00410  
EVA00420  
EVA00430  
EVA00440  
EVA00450  
EVA00460  
EVA00470  
EVA00480  
EVA00490

FILE: EVLFET

```
      SURROUTINE EVLFET(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,AVEMT2,
      S,S2,WRKRY,IWRKSZ)
C      INCLUDE COMRK1.LIST
C      INCLUDE COMRK7.LIST
      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
      AVAR2,COVAR2,CLSIO2,SURNO2,SURDS2,FLDSV2,VERTX2,
      FETVC2(30),SURVC2(75),SURPTR(75),CLSVC2(60),
      KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
      GRPCHK(41),GROUPS(124)
      COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
      INCVEC(30),ICOUNT,SETWGT,EVALRF(100),FETVC4(30)
      NOFET4,VARS74,CORHAS,DTAH4,WGHS14,BESTVC(10),DIVSZ
      STATKY,ADRESD,ADRESP,ADRESF,ADRSH1,ADRSH2
      INTEGER ADRESD,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY
      DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
CSEND
      DOUBLE PRECISION COVMT2(1),AVEMT2(1),S2(1),DUM(1),DIVTAB(1)
      INTEGER VARSZ4,FETVC4
      INTEGER CRIKEY
      DIMENSION AVEMTX(1),COVMTX(1), WEIGHT(1),
      S(1), WRKRY(1)
      IPART=-1
      CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,FETVC4,
      DUM,WRKRY,IWRKSZ)
      CALL EVALSP(SEPMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,
      WEIGHT,IPART,DUM,DUM,WRKRY,IWRKSZ)
      IF(CRIKEY.NE.1)RETURN
      CALL DIVRG1(COVMT2,VARSZ4,AVEMT2,DIVTAB,NOCLS2,NOFET4,WRKRY,
      IWRKSZ)
      RETURN
      END
```

EVL00010  
EVL00020  
EVL00030  
EVL00040  
EVL00050  
EVL00060  
EVL00070  
EVL00080  
EVL00090  
EVL00100  
EVL00110  
EVL00120  
EVL00130  
EVL00140  
EVL00150  
EVL00160  
EVL00170  
EVL00180  
EVL00190  
EVL00200  
EVL00210  
EVL00220  
EVL00230  
EVL00240  
EVL00250  
EVL00260  
EVL00270  
EVL00280  
EVL00290  
EVL00300  
EVL00310

FILE: EXSRCH

```

      SURROUTINE EXSRCH(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,
      AVEMT2,S,S2,WRKRY,IWRKSZ)
C*
C* THIS SURROUTINE USES THE EXHAUSTIVE SEARCH PROCEDURE TO FIND
C* THE BEST 'NOFET4' OUT OF 'NOFET2' FEATURES, BY MAXIMIZING THE
C* SEPARABILITY MEASURE INDICATED BY 'CRIKEY'.
C*
C* INCLUDE COMRK1.LIST
C* INCLUDE COMRK7.LIST
C* COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
C*   AVAR2,COVAR2,CLS102,SURN02,SURDS2,FLDSV2,VERTX2,
C*   FFTVC2(30),SUBVC2(75),SURPTR(75),CLSVC2(60),
C*   KFPPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),
C*   GRPCHK(61),GROUPS(124)
C* COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
C*   INCVEC(30),ICOUNT,SETWGT,EVALBF(100),FETVC4(30)
C*   ,NOFET4,VARSZ4,CORBAS,DTAB4,WGHS14,RESTVC(10),DIVSIZ
C*   ,STATKY,ADRES0,ADRES1,ADRESF,ADRSH1,ADRSH2
C*   INTEGER ADRES0,ADRES1,ADRESF,ADRSH1,ADRSH2,STATKY
C*   DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
CSEND
C*
C* INTEGER FETVC2,FETVC4,TVEC,VARSZ2,VARSZ4
C* INTEGER DIVSIZ,SVFC(30),CRIKEY
C* DOUBLE PRECISION TMSR,DIVTAB(DIVSIZ),DUM(1),DM
C* DOUBLE PRECISION COVMT2(VARSZ4,1),AVEMT2(NOFET4,1),S2(VARSZ4,1)
C* DIMENSION COVMTX(VARSZ2,NOCLS2) ;
C*   AVEMTX(NOFET2,NOCLS2) ;
C*   S(VARSZ2,NOCLS2)
C* DIMENSION TVEC(30)
C* DIMENSION WEIGHT(1),WRKRY(1)
C*
C* INITIALIZE TVEC FOR GETSET ROUTINE
C*
C* DO 1 I=1,NOFET4
C*   TVFC(I)=I
C*   TVFC(NOFET4)=NOFET4-1
C*   SEPMSR=1.E+35
C*   GET NEXT SET OF FEATURES
C*
C* 4 CALL GETSET(TVEC,NOFET4,NOFET2,LAST)
C*   IF(LAST.EQ.0)GO TO 10
C*
C* GET SURSET OF STATISTICS FOR THIS SET OF FEATURES
C*
C* CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,TVEC,DM,WRKRY,IWRKSZ)
C*
C* EVALUATE SEPARABILITY MEASURE FOR THIS SET OF FEATURES.
C* SET IPART SO PARTIAL DERIATIVES WILL NOT BE CALCULATED.
C*
C* IPART=-1
C* CALL EVALSP(TMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,
C*   WEIGHT,IPART,DUM,DUM,WRKRY,IWRKSZ)
C* IF (SEPMSR.LT.TMSR)GO TO 4
C* DO 5 I=1,NOFET4
C*   K=TVFC(I)
C*   SVFC(I)=K
C* 5 FETVC4(I)=FFTVC2(K)
C*   SEPMSR=TMSR
C* GO TO 4
C* FINISHED
C*
C* COMPUTE INTERCLASS MEASURES FOR FEATURES CHOSEN
C*
C* 10 CONTINUE
C* CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,SVFC,DM,WRKRY,IWRKSZ)
C* CALL EVALSP(SEPMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,
C*   WEIGHT,IPART,DUM,DUM,WRKRY,IWRKSZ)
C* IF (CRIKEY.NE.1)RETURN
C* CALL DIVRG(COVMT2,VARSZ4,AVEMT2,DIVTAB,NOCLS2,
C*   NOFET4,WRKRY,IWRKSZ)
C* 20 RETURN
C* END

```

EXS00010  
EXS00020  
EXS00030  
EXS00040  
EXS00050  
EXS00060  
EXS00070  
EXS00080  
EXS00090  
EXS00100  
EXS00110  
EXS00120  
EXS00130  
EXS00140  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00060  
EXS00220  
EXS00230  
EXS00240  
EXS00250  
EXS00260  
EXS00270  
EXS00280  
EXS00290  
EXS00300  
EXS00310  
EXS00320  
EXS00330  
EXS00340  
EXS00350  
EXS00360  
EXS00370  
EXS00380  
EXS00390  
EXS00400  
EXS00410  
EXS00420  
EXS00430  
EXS00440  
EXS00450  
EXS00460  
EXS00470  
EXS00480  
EXS00490  
EXS00500  
EXS00510  
EXS00520  
EXS00530  
EXS00540  
EXS00550  
EXS00560  
EXS00570  
EXS00580  
EXS00590  
EXS00600  
EXS00610  
EXS00620  
EXS00630  
EXS00640  
EXS00650  
EXS00660  
EXS00670  
EXS00680  
EXS00690  
EXS00700  
EXS00710  
EXS00720  
EXS00730  
EXS00740



FILE: FINT1

```
C*      WRKRY- WORKING STORAGE                                F|N00800
C*      IWRKZ- SIZE OF WRKRY IN COMPUTER WORDS              F|N00810
C*      .....                                              F|N00820
C*      DIMENSION BMAT(1),WRKRY(1)                          F|N00830
C*      DOUBLE PRECISION BMAT,SMSR                          F|N00840
C*      GET TRANSFORMED STATISTICS FOR THIS B-MATRIX       F|N00850
C*      CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DUM,BMAT,WRKRY,   F|N00860
C*      IWRKZ)                                               F|N00870
C*      CALL FVALSP(SPSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,   F|N00880
C*      WEIGHT,IPART,PARTLS,BMAT,WRKRY,IWRKZ)               F|N00890
C*      SMSR=NARS(SPSR)                                       F|N00900
C*      ITT=ITT+1                                             F|N00910
C*      ICNT=ICNT+1                                           F|N00920
C*      IF(ICNT.LE.1)GO TO 50                                F|N00930
C*      IF(ICNT.LE.30)GO TO 50                               F|N00940
C*      WRITE(4,700)                                          F|N00950
C*      ICNT=0                                                F|N00960
C*      RETURN 1                                              F|N00970
50 CONTINUE                                                  F|N00980
C*      IF(IPART.LT.0)RETURN                                  F|N00990
C*      ICNT=0                                                F|N01000
C*      ICYCLE=ICYCLE+1                                       F|N01010
C*      IF(CRIKEY.EQ.1)RATIO=SMSR/TOTMSR                     F|N01020
C*      IF(CRIKEY.NE.1)RATIO=TOTMSR/SMSR                     F|N01030
C*      WRITE(6,600)ICYCLE,ITT,SMSR,RATIO                    F|N01040
C*      RETURN                                                F|N01050
100 FORMAT(///25X,'CONVERGENCE CHARACTERISTIC SUMMARY FOR THE DAVIDN-   F|N01060
C*      LETCHER-POWELL PROCEDURE'//25X,76('-'//)           F|N01070
200 FORMAT(//35X,'NUMBER OF LINEAR COMBINATIONS',10X,'=',I10/   F|N01080
C*      35X,'DESIGN NO. OF FUNCTIONAL EVAL.',8X,'=',I10)    F|N01090
300 FORMAT(35X,'MAX. WEIGHTED AVERAGE DIVERGENCE (D)',3X,'=',E10.5) F|N01100
400 FORMAT(35X,'MIN. WEIGHTED AV.TRANS. DIVERGENCE (T)',3X,'=',E10.5) F|N01110
500 FORMAT(35X,'MIN. WEIGHTED AV.BHATTACHARYYA DIS. (H)',3X,'=',E10.5) F|N01120
600 FORMAT(20X,I4,T37,I4,T61,F12.7,T90,E12.7)               F|N01130
700 FORMAT(' MAX. ITERATIONS PER CYCLE - BEGIN NEW CYCLE')   F|N01140
END                                                            F|N01150
F|N01160
F|N01170
F|N01180
```

FILE: GENRPT

```
SUBROUTINE GENRPT(CLSNAM,WEIGHT,DIVTAB,WRKRY,IWRKSZ,FETVEC) GEN00010
DIMENSION CLSNAM(NOCLS2),WEIGHT(1) GEN00020
C INCLUDE COMRK6.LIST GEN00030
DOUBLE PRECISION RATIO GEN00040
DOUBLE PRECISION WRKRY(1),DIVTAB(1) GEN00050
C DIMENSION FETVEC(30) GEN00060
C INCLUDE COMRK1.LIST GEN00070
C INCLUDE COMRK7.LIST GEN00080
COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2, GEN00090
AVAR2,COVAR2,CLS1D2,SURNO2,SURDS2,FLDSV2,VERTX2, GEN00100
FETVC2(30),SUHVC2(75),SURPTR(75),CLSVC2(60), GEN00110
KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61), GEN00120
GRPCHK(61),GROUPS(124) GEN00130
COMMON/GLORAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY, GEN00140
HISFIL,HISKEY,TRFORM,FRIPTR,ERPKEY,MAPUNT,NOFILF, GEN00150
ORIMAD,ORMWD5,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVL GEN00160
,NHSTUN,NHSTFI,SCRTRN,MAPFIL GEN00170
,DOTUNT,NOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, GEN00180
CRDUNT,PRUNT,RANDIO GEN00190
COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET, GEN00200
INVC4(30),ICOUNT,SETWGT,EVALBF(100),FETVC4(30) GEN00210
,NOFET4,VAR74,CORBAS,DTAR4,WGH514,RESTVC(10),DIVSIZ GEN00220
,STATKY,ADRESD,ADRESF,ADRESH,ADRS2 GEN00230
INTEGER ADPFSO,ADRESF,ADRESH,ADRS1,ADRS2,STATKY GEN00240
DOUBLE PRECISION CFAC,TOTMSR,SEPMSR GEN00250
CSEND GEN00260
INTEGER CRIKEY,PRCKEY GEN00270
DIMENSION PRC(6,6),CRI(13,4) GEN00280
DATA PRC/ EXHA,USTI,VF S,EARC,H , , , GEN00290
WITH,OUT,REPL,ACEM,ENT , , , GEN00300
DAVI,DON,FLET,CHER,-POW,ELL , , , GEN00310
EVAL, B-,MATR,IX R,EQUE,ST , , , GEN00320
EVAL, FE,ATUR,ES R,EQUE,ST , , , GEN00330
PFST, K, OF, N, PA, SSES, / GEN00340
DATA CRI/ WEIG,HTED, AVE, PAGE, DIV, ERGE, NCE , , GEN00350
WEIG,HTED, AV, TRAI,NSFO, RMED, DIV, , , GEN00360
ERGE, NCE , , , , , , , , , , , , , GEN00370
WEIG,HTED, AV, BHA,TTAC,HARY, YA D, , , GEN00380
ISTA, NCE , , , , , , , , , , , , , GEN00390
PROB, ABIL, ITY, OF M, ISCL, ASSI, FICA, , , GEN00400
TION, OF, TRAI, NING, DAT, A , / GEN00410
INTEGER DIVSIZ,PAGSIZ GEN00420
INTEGER FETVC4 GEN00430
IPCNT=27 GEN00440
WRITE(6,HEAD) GEN00450
WRITE(6,100)(PRC(I,PRCKEY),I=1,6),(CRI(I,CRIKEY),I=1,13) GEN00460
IR=1 GEN00470
IE=15 GEN00480
5 IF(IF.GT.NOFET2)IF=NOFET2 GEN00490
WRITE(6,105)(FETVC2(I),I=IR,IE) GEN00500
IF(IF.EQ.NOFET2)GO TO 10 GEN00510
IR=IE+1 GEN00520
IE=IF+15 GEN00530
GO TO 5 GEN00540
10 CONTINUE GEN00550
WRITE(6,125) GEN00560
IF(PRCKEY.EQ.3.OR.PRCKEY.EQ.4)WRITE(6,110)NOFET4 GEN00570
IF(PRCKEY.EQ.3.OR.PRCKEY.EQ.4)GO TO 1A GEN00580
IR=1 GEN00590
IE=14 GEN00600
12 IF(IF.GT.NOFET4)IF=NOFET4 GEN00610
IF(PRCKEY.LT.3)WRITE(6,120)(FETVC4(I),I=IR,IE) GEN00620
IF(PRCKEY.GE.5)WRITE(6,115)(FETVEC(I),I=IR,IE) GEN00630
IF(IE.EQ.NOFET4)GO TO 1B GEN00640
IR=IF+1 GEN00650
IE=IF+14 GEN00660
GO TO 12 GEN00670
1A CONTINUE GEN00680
IF(CRIKEY.EQ.1)SEPMSR=DAHS(SEPMSK) GEN00690
IF(PRCKEY.EQ.3)WRITE(6,130)SEPMSR GEN00700
IF(PRCKEY.EQ.4)WRITE(6,130)SEPMSR GEN00710
IF(PRCKEY.GE.5)WRITE(6,145)SEPMSR GEN00720
IF(PRCKEY.LT.3)WRITE(6,140)SEPMSR GEN00730
IF(CRIKEY.EQ.1)RATIO=SEPMSR/TOTMSR GEN00740
IF(CRIKEY.NE.1)RATIO=TOTMSR/SEPMSR GEN00750
IF(CRIKEY.NE.1)WRITE(6,150)TOTMSR,RATIO GEN00760
IF(CRIKEY.EQ.1)WRITE(6,155)TOTMSR,RATIO GEN00770
WRITE(6,160) GEN00780
GEN00790
```

FILE: GENRPT

```
IF (CRIKEY.EQ.1) WRITE(6,170)
IF (CRIKEY.EQ.2) WRITE(6,175)
IF (CRIKEY.EQ.3) WRITE(6,180)
IF (PRCKEY.LE.2) WRITE(6,190)
IF (PRCKEY.EQ.3) WRITE(6,200)
IF (PRCKEY.EQ.4) WRITE(6,200)
IF (PRCKEY.GE.5) WRITE(6,205)
WRITE(6,125)

C*
C*
C*
READ INTERCLASS MEASURE FOR ALL FEATURES - COMPUTED AND SAVED IN
PRELIM.

NW=DIVSIZ*2
CALL WREAD(ADRESO,WRKRY,NW,ISTAT)
19 IF (ISTAT.EQ.1) GO TO 19
NC=NOCLS2-1
IK=0
DO 20 I=1,NC
K=I+1
DO 20 J=K,NOCLS2
IK=IK+1
IF (CRIKEY.EQ.1) RATIO=DIVTAB(IK)/WRKRY(IK)
IF (CRIKEY.NE.1) RATIO=WRKRY(IK)/DIVTAB(IK)
WRITE(6,210) CLSNAM(I),CLSNAM(J),
* WEIGHT(IK),DIVTAB(IK),WRKRY(IK),RATIO

IPCNT=IPCNT+1
IF (IPCNT.LT.PAGSIZ) GO TO 20
IPCNT=17
IF (IK.EQ.DIVSIZ) GO TO 20
WRITE(6,HEAD)
WRITE(6,160)
IF (CRIKEY.EQ.1) WRITE(6,170)
IF (CRIKEY.EQ.2) WRITE(6,175)
IF (CRIKEY.EQ.3) WRITE(6,180)
IF (PRCKEY.LE.2) WRITE(6,190)
IF (PRCKEY.EQ.3) WRITE(6,200)
IF (PRCKEY.EQ.4) WRITE(6,200)
IF (PRCKEY.GE.5) WRITE(6,205)
WRITE(6,125)
20 CONTINUE
IF (CRIKEY.NE.2) RETURN

C*
C*
C*
GET DIVERGENCE BACK FOR PLOTS

DO 30 I=1,DIVSIZ
WRKRY(I)=-14.*DLOG(WRKRY(I))
DIVTAB(I)=-14.*DLOG(DIVTAB(I))
30 CONTINUE
100 FORMAT(1X,T35,'RESULTS FOR CHANNEL SELECTION ACTIVITY USING: //
* 1X,T40,'OPTIMIZATION PROCEDURE - ',F44.7
* 1X,T40,'SEPARABILITY MEASURE - ',F134.4)
105 FORMAT(1X,T40,'CHANNELS CONSIDERED - ',F15(12,' '))
110 FORMAT(1X,T35,'NO. OF LINEAR COMBINATIONS - ',I3)
115 FORMAT(1X,T35,'CHANNELS EVALUATED - ',I3)
* 16(12,' '))
120 FORMAT(1X,T35,'CHANNELS SELECTED - ',I3)
* 12,15(' ',I2))
125 FORMAT(1X,T35,'SEPARABILITY MEASURE FOR LINEAR COMB. - ',F14.8)
140 FORMAT(1X,T35,'SEPARABILITY MEASURE FOR SELECTED CHANNELS - ',F14.8)
145 FORMAT(1X,T35,'SEPARABILITY MEASURE FOR EVALUATE REQUEST - ',F14.8)
* E14.8)
150 FORMAT(1X,T35,'MINIMUM SEP.MEASURE (USING ALL CHANNELS) - ',F14.8)
* E14.8/1X,T35,'RATIO',T78,' - ',F14.8)
155 FORMAT(1X,T35,'MAXIMUM SEP.MEASURE (USING ALL CHANNELS) - ',F14.8)
* F14.8/1X,T35,'RATIO',T78,' - ',F14.8)
160 FORMAT(/// 1X,T50,'INTERCLASS SEPARABILITY TABLE' //)
170 FORMAT(1X,T73,'INTERCLASS DIVERGENCE')
175 FORMAT(1X,T65,'INTERCLASS TRANSFORMED DIVERGENCE')
180 FORMAT(1X,T65,'INTERCLASS BHATTACHARYYA DISTANCE')
190 FORMAT(1X,T22,'SUBCLASS PAIR',T41,'WEIGHT',T58,'SELECTED CHANNELS',
* T40,'ALL CHANNELS',T99,'RATIO')
200 FORMAT(1X,T22,'SUBCLASS PAIR',T41,'WEIGHT',T58,'LINEAR COMBINATION',
* T40,'ALL CHANNELS',T99,'RATIO')
205 FORMAT(1X,T24,'SUBCLASS PAIR',T41,'WEIGHT',T58,'EVALUATED CHANNELS',
* T40,'ALL CHANNELS',T99,'RATIO')
210 FORMAT(20X,A4,7X,A4,4X,E8.3,T60,D14.8,T78,D14.8,T96,D14.8)

GEN00800
GEN00810
GEN00820
GEN00830
GEN00840
GEN00850
GEN00860
GEN00870
GEN00880
GEN00890
GEN00900
GEN00910
GEN00920
GEN00930
GEN00940
GEN00950
GEN00960
GEN00970
GEN00980
GEN00990
GEN01000
GEN01010
GEN01020
GEN01030
GEN01040
GEN01050
GEN01060
GEN01070
GEN01080
GEN01090
GEN01100
GEN01110
GEN01120
GEN01130
GEN01140
GEN01150
GEN01160
GEN01170
GEN01180
GEN01190
GEN01200
GEN01210
GEN01220
GEN01230
GEN01240
GEN01250
GEN01260
GEN01270
GEN01280
GEN01290
GEN01300
GEN01310
GEN01320
GEN01330
GEN01340
GEN01350
GEN01360
GEN01370
GEN01380
GEN01390
GEN01400
GEN01410
GEN01420
GEN01430
GEN01440
GEN01450
GEN01460
GEN01470
GEN01480
GEN01490
GEN01500
GEN01510
GEN01520
GEN01530
GEN01540
GEN01550
GEN01560
GEN01570
GEN01580
```

FILE: GENRPT

RETURN  
END

GEN01590  
GEN01600

~~10-34~~

104





FILE: GTSTAT

```
      SUBROUTINE GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,      GTS00010
      VEC,BMAT,WRKRY,IWRKSZ)                                GTS00020
C*                                                         GTS00030
C* THIS SUBROUTINE SELECTS THE SUBSETS OF THE STATISTICAL  GTS00040
C* PARAMETERS COVMTX,AVEMTX AND S DEFINED BY VEC OR BMAT AND GTS00050
C* STORES THE SUBSETS INTO COVMT2,AVEMT2, AND S2 RESPECTIVELY. GTS00060
C*                                                         GTS00070
C* INCLUDE COMCHK,LIST                                     GTS00080
C* INCLUDE COMCHK7,LIST                                   GTS00090
C* COMMON/INFOVM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,  GTS00100
C*   AVAR2,COVAR2,CLSID2,SUMNO2,SUBNS2,FLDSV2,VERTX2,      GTS00110
C*   FETVC2(30),SUHVC2(75),SUMPTH(75),CLSV2(60),          GTS00120
C*   KEPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),              GTS00130
C*   GRPCHK(61),GROUPS(124)                               GTS00140
C* COMMON/FSL/CFAC,TOTMSR,SEPMSH,PRCKEY,CRIKEY,INCFET,    COM00010
C*   INCVEC(30),ICOUNT,SETWGT,EVALRF(100),FETVC4(30)      COM00020
C*   ,NOFET4,VARSZ4,CUMHAS,DIARA,WGMS14,RESTVC(10),DIVSIZ COM00030
C*   ,STATKY,ADHESD,ADRESF,ADRESF,ADRSH1,ADRSH2          COM00040
C*   INTEGER ADHESD,ADRESF,ADRESF,ADRSH1,ADRSH2          COM00050
C*   DOUBLE PRECISION CFAC,TOTMSR,SEPMSR                  COM00060
CSEND                                                    GTS00220
C* INTEGER PRCKEY,CRIKEY,VEC,VARSZ2,VARSZ4               GTS00230
C* DOUBLE PRECISION COVMT2(VARSZ4,1),AVEMT2(NOFET4,1),   GTS00240
C*   S2(VARSZ4,1),BMAT(NOFET4,1),WRKRY(1),SUM            GTS00250
C* DIMENSION COVMTX(VARSZ2,NOCLS2),                      GTS00260
C*   AVEMTX(NOFET2,NOCLS2),                               GTS00270
C*   S(VARSZ2,NOCLS2),VEC(1)                              GTS00280
C*                                                         GTS00290
C* IF WITHOUT REPLACEMENT OR EX. SEARCH PROCEDURE, SELECT ELEMENTS GTS00300
C* DETERMINED BY 'VEC'.                                    GTS00310
C* GO TO (5,5,20,20,5,5),PRCKEY                           GTS00320
5 CONTINUE                                               GTS00330
  DO 10 I=1,NOCLS2                                       GTS00340
    IK=0                                                  GTS00350
    DO 10 J=1,NOFET4                                       GTS00360
      K=VEC(J)                                             GTS00370
      LOC=K*(K-1)/2                                         GTS00380
      AVEMT2(J,1)=AVEMTX(K,I)                             GTS00390
      DO 10 L=1,J                                           GTS00400
        IK=IK+1                                             GTS00410
        IW=VEC(L)*LOC                                       GTS00420
        COVMT2(IK,1)=COVMTX(IW,I)                         GTS00430
        IF(CHIKFY.NE.1)GO TO 10                             GTS00440
      S2(IK,1)=S(IW,I)                                     GTS00450
10 CONTINUE                                               GTS00460
  RETURN                                                  GTS00470
C*                                                         GTS00480
C* DAVIDON OR USER INPUT PROCEDURE, MULTIPLY R-MATRIX   GTS00490
C*                                                         GTS00500
20 CONTINUE                                               GTS00510
  IW1=1                                                  GTS00520
  ITFST=IW1*NOFET2*NOFET4                                GTS00530
  IF(IWKSZ/2.GE.ITFST)GO TO 30                            GTS00540
  WRITE(6,100)IWRKSZ                                       GTS00550
  CALL CMEMR                                              GTS00560
30 CONTINUE                                               GTS00570
  DO 40 I=1,NOCLS2                                       GTS00580
    DO 40 J=1,NOFET4                                       GTS00590
      SUM=0.0                                              GTS00600
      DO 40 K=1,NOFET2                                       GTS00610
        SUM=SUM+AVEMTX(K,I)*BMAT(J,K)                    GTS00620
      AVEMT2(I,1)=SUM                                       GTS00630
40 CONTINUE                                               GTS00640
  CALL TRANSR(COVMTX,COVMT2,WRKRY(IW1),BMAT)            GTS00650
  IF(CHIKFY.EQ.1)CALL TRANSR(S,S2,WRKRY(IW1),BMAT)      GTS00660
  RETURN                                                  GTS00670
100 FORMAT(' NOT ENOUGH WORK AREA IN GTSTAT -- IWRKSZ=',I5) GTS00680
  END                                                    GTS00690
```

FILE: MT1

C  
C  
C  
C  
C  
C  
C

MT1 - MATRIX B IS STORED IN SYMMETRIC NOTATION AND SINGLE PRE.

C

```
SUBROUTINE MT1(A,R,C,M,N)
DOUBLE PRECISION A(M,N),C(M,N),SUM
DIMENSION M(1)
MATRIX R IS STORED IN SYMMETRIC NOTATION
DO 60 II=1,M
DO 60 JJ=1,N
SUM=0.0
DO 55 KK=1,N
IF(KK.GT.JJ)IK=KK*(KK-1)/2 + JJ
IF(KK.LT.JJ)IK=JJ*(JJ-1)/2 + KK
55 SUM = SUM + A(II,KK)*R(II,IK)
60 C(II,JJ)=SUM
RETURN
END
```

FILE: MT2

```
C*  SURROUTINE TO FORM PRODUCT OF MATRICES A*B AND STORE IN C.
C*
C*  MT2 - MATRIX A IS STORED IN SYMMETRIC NOTATION
C*
SUBROUTINE MT2(A,R,C,M,N)
DOUBLE PRECISION A(1),R(M,N),C(M,N),SUM1
DO 90 II=1,M
DO 90 JJ=1,N
SUM1=0.0
DO 85 IJ=1,M
IF (IJ.GE.II) IK=IJ*(IJ-1)/2+II
IF (IJ.LE.II) IK=II*(II-1)/2+IJ
85 SUM1=SUM1 + A(IK)*R(IJ,JJ)
90 C(II,JJ)=SUM1
RETURN
END
```

FILE: MT3

```
C*
C*  MATRICES A AND/OR B MAY BE STORED IN SYMMETRIC NOTATION
C*  L=M IF A IS SYMMETRIC ---- N=M IF B IS SYMMETRIC
SUBROUTINE MT3(A,B,C,L,M,N,ISYMA,ISYMB)
DOUBLE PRECISION A(1),B(1),C(L,N),SUM
DO 70 II=1,L
DO 70 JJ=1,N
SUM=0.0
DO 65 KK=1,M
IF (ISYMA.EQ.1) GO TO 61
IK=L*(KK-1) + II
GO TO 62
61 IF (KK.GE.II) IK=KK*(KK-1)/2 + II
IF (KK.LT.II) IK=II*(II-1)/2 + KK
62 IF (ISYMB.EQ.1) GO TO 63
JK=M*(JJ-1) + KK
GO TO 65
63 IF (KK.GE.JJ) JK=KK*(KK-1)/2 + JJ
IF (KK.LT.JJ) JK=JJ*(JJ-1)/2 + KK
65 SUM=SUM + A(IK)*B(JK)
70 C(II,JJ)=SUM
RETURN
END
```

FILE: MT4

```
C*
C* SUPROUTINE MT4(A,R,C,L,M,N,ISYM)
C*
C* MATRIX A CAN BE STORED FULL OR IN SYMMETRIC NOTATION
C*
C* ISYM=1 IF A IS SYMMETRIC
C* ISYM=0 IF A IS FULL
C*
DOUBLE PRECISION A(1),B(M,N),C(L,N),SUM
DO 95 II=1,L
DO 95 JJ=1,N
SUM=0.0
DO 99 IJ=1,M
IF (ISYM.EQ.1) GO TO 85
IK=L*(IJ-1) + II
GO TO 99
85 IF (IJ.GE.II) IK=IJ*(IJ-1)/2 + II
IF (IJ.LT.II) IK=II*(II-1)/2 + IJ
99 SUM = SUM + A(IK)*B(IJ,JJ)
95 C(II,JJ)=SUM
RETURN
END
```

FILE: PLOT

```
.....
SURROUTINE PLOT(X,Y,NOX,MAXX,ILABX,ILABY,ICODE,IOPT)
.....
THIS SUBROUTINE WAS WRITTEN BY J.K.DALY OF TRW FOR THE ASTEP
PROGRAM. IT WAS MODIFIED SLIGHTLY FOR USE IN THIS PROGRAM BY
R. MINTER
--ICODE AND IOPT ARE SET TO 0, AND ILABX,ILABY ARE BLANK.
WRITE STATEMENTS WERE ADDED FOR LABELING.
.....
PROGRAMMER - J.K. DALY
DATE - FEBRUARY, 1973
MODIFIED FOR IRM 370-148 R HANSEN,C HORTON DEC,1977
INPUT VARIABLES
X - ARRAY CONTAINING X COORDINATES (TYPE REAL)
Y - ARRAY CONTAINING Y COORDINATES (TYPE REAL)
NOX - NO. OF X COORDINATES INPUT (MAX. 39)
MAXX - MAXIMUM VALUE OF POINT ON SCALE TO BE INPUT (INTEGER)
ILABX - LABEL OF THE X-AXIS (MAX. OF 78 CHARACTERS ALLOWED)
ILABY - LABEL OF THE Y-AXIS (MAX. OF 78 CHARACTERS ALLOWED)
ICODE - 1 = RUN IS MADE ON TSS,INVALID 12-77
0 = RUN IS MADE ON 1108
IOPT - 0 = 45 DEGREE ANGLE LINE TO BE DRAWN ON PLOT
1 = NO 45 DEG ANGLE LINE,INVALID 12-77
INTERNAL VARIABLES
FMDX - WHICH WORD OF THE FORMAT ARRAY(FMTARY) TO START
LOADING THE LABELS.
FMTARY - ARRAY TO BE BUILT FOR EACH LINE OF THE PLOT
I - INDEX FOR INNER LOOP FOR DETERMINING THE LOCATION
OF THE POINT ON THE X-AXIS
INDEX - MAXIMUM NO. OF WORDS OF FMTARY TO BE WRITTEN (TSS=11,
1108=15)
INITLN - INITIAL LINE TO BEGIN PRINTING THE LABELS
INUM - NUMBER OF POINTS TO FALL IN SAME PLACE ON PLOT.
J - INDEX FOR LOOP TO FIND NEXT HIGHEST Y-COORDINATE.
K - INDEX FOR OUTER LOOP BUILDING AND PRINTING EACH
LINE OF PLOT.
LARNDX - INDEX FOR LABARY
LASTLN - LAST LINE TO PRINT LABEL ON PLOT
M - INDEX OF SCLARY
MHORIZ - NO. OF CHARACTERS/PLOT
MLNCT - NO. OF LINES/PLOT
NDX - INDEX TO 'WICLAR'
NOXPT - NO. OF POINTS TO FALL BETWEEN LABELED POINTS
ON X-AXIS
NOYPT - NO. OF POINTS TO FALL BETWEEN LABELED POINTS
ON Y-AXIS
SCLARY - ARRAY CONTAINING POINTS TO BE PRINTED AS LABELS
ON THE X- AND Y-AXIS
WICLAR - ARRAY CONTAINING LINE NO.S OF BOTH KINDS OF PLOTS.
DEPENDING ON THE LINE NO. DETERMINES WHICH LABEL
WILL BE PRINTED.
XLNVLU - VALUE OF EACH POINT ON THE X-AXIS. USED TO CALCULATE
EACH POINT OF XSCLAR (X SCALE ARRAY) IN SURROUTINE
SCALEF.
XSCLAR - ACTUAL VALUE OF EACH POINT ON X-AXIS (TYPE REAL)
YLNVLU - VALUE OF EACH POINT ON Y-AXIS. USED TO CALCULATE EACH
POINT OF YSCLAR (Y SCALE ARRAY)
* * * * *
INCLUDE COMR6.LIST
INCLUDE COMR7.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,RMKEY,
HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILF,
* DRUMAD,DRMADS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* .NHSTUN,NHSTFL,SCTRUN,MAPFIL
* .DOTUNT,DOTFIL,NCHPAS,TPNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,KANDIO
COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRKEY,INCFET,
INCVEC(30),ICOUNT,SFTWGT,FVALRF(100),FFTVC4(30)
* .NOFET4,VARS74,CORHAS,DTA94,WGHS14,HESTVC(10),DIVSI7
* .STATKY,ADRESO,ADRESO,ADRESF,ADRSH1,ADRSH2
INTEGER ADRESO,ADRESO,ADRESF,ADRSH1,ADRSH2,STATKY
INTEGER PCO9,PCOX,PCORLK,PCOSTR
DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
* * * * *
CSEND
C DIMENSION FMTARY(23)
C
```

FILE: PLOT

```
REAL                                YSCLAR(57),XSCLAR(85)
DOUBLE PRECISION X(NOX),Y(NOX)
REAL SCLARY(N)
INTEGER CRKEY,PRKEY
INTEGER LABARY(40),LARNOX,WICLAR(10),FMDX
INTEGER ILAPLX(20),ILABLY(20),FMTARY,ASTRIC,0
C
C LOGICAL*1 FMTA(92),LTEST(4),LSTAR,LOGI
C
DATA LOGI/ZF1/,LRLANK/Z40/,LSTAR/Z5C/
DATA FMTARY/23*4H /,ASTRIC/4H***//,IBLANK/4H /
DATA IPOINT/4H /
DATA RCDRLK/Z40/,RCDSTH/Z5C/,HCD9/ZF9/,BCDX/ZE7/
DATA WICLAR /55.54.52.50.49.42.41.40.39.38/
C
C EQUIVALENCE (FMTA(1),FMTARY(1)),(ITEST,LTEST(1))
C
C BLANK X AND Y LABELS
DO 100 I=1,20
  ILAPLX(I)=IBLANK
  ILABLY(I)=IBLANK
C
C* FIND MAXX
MAXX=1
AMAX=0.
IF(CRKEY.EQ.3)GO TO 900
IKCT=0
C
C COUNT NUMBER OF X(I) GREATER THAN 700. SET AMAX = LARGEST X(I)
C SET MAXX=700 UNLESS MORE THAN 1/5 OF ELEMS GRTR 700.THEN MAXX=1400
DO 800 I=1,NOX
  IF(X(I).GT.700)IKCT=IKCT+1
  800 CONTINUE
  MAXX=700
  IF(FLOAT(IKCT)/FLOAT(NOX) .GT. .2)MAXX=1400
  900 CONTINUE
C
C WRITE MESSAGE 'SEPARABILITY TO BE GAINED MAP'
CALL SETMRG(66,2,64)
WRITE(6,1695)
1695 FORMAT(50X,'SEPARABILITY-TO-BE-GAINED MAP')
C
C BLANK LABARY
DO 1000 II=1,40
  LABARY(II) = IBLANK
  1000 CONTINUE
C
C BLANK FMTARY
DO 1001 III=1,23
  FMTARY(III) = IBLANK
  1001 CONTINUE
C
C STORE X- AND Y-AXIS LABELS INTO LABEL ARRAY(LABARY)
DO 1020 II=1,20
  IF (ILABLY(II).EQ.0) GO TO 1020
  LABARY(II) = ILABLY(II)
  IF (ILAPLX(II).EQ.0) GO TO 1020
  LABARY(II+20) = ILAPLX(II)
  1020 CONTINUE
C
C* * * SET VARIABLES FOR PRINT PLOT ON 1108
LARNOX = 1
N = 7
MLNCT = 56
MPORTZ = 45
NOYPT = 12
NOYPT = 4
NOSPAC = 46
INITLN = 55
LASTLN = 44
NOX = 1
FMDX = 3
INDEX = 23
C
C CALL SCALE TO FOR SCALE FOR PLOT
1050 CONTINUE
C
C
```

PL000800  
PL000810  
PL000820  
PL000830  
PL000840  
PL000850  
PL000860  
PL000870  
PL000880  
PL000890  
PL000900  
PL000910  
PL000920  
PL000930  
PL000940  
PL000950  
PL000960  
PL000970  
PL000980  
PL000990  
PL001000  
PL001010  
PL001020  
PL001030  
PL001040  
PL001050  
PL001060  
PL001070  
PL001080  
PL001090  
PL001100  
PL001110  
PL001120  
PL001130  
PL001140  
PL001150  
PL001160  
PL001170  
PL001180  
PL001190  
PL001200  
PL001210  
PL001220  
PL001230  
PL001240  
PL001250  
PL001260  
PL001270  
PL001280  
PL001290  
PL001300  
PL001310  
PL001320  
PL001330  
PL001340  
PL001350  
PL001360  
PL001370  
PL001380  
PL001390  
PL001400  
PL001410  
PL001420  
PL001430  
PL001440  
PL001450  
PL001460  
PL001470  
PL001480  
PL001490  
PL001500  
PL001510  
PL001520  
PL001530  
PL001540  
PL001550  
PL001560  
PL001570  
PL001580



FILE: PLOT

```
      CALL SCALE(MAXX,MLNCT,INCR,YSCLAR,XSCLAR,SCLARY,XLNVLU,YLNVLU,
      *      MHORIZ,NOXPT,NOYPT)
      XLNVLU = XLNVLU/2.0
      YLNVLU = YLNVLU/2.0
1000 CONTINUE
C
C      FIND Y COORDINATE
      DO 1400 I1400 = 1,MLNCT
      K = MLNCT - I1400 + 1
      DO 1100 J=1,NOX
      YY = Y(J)
      IF(YY.GT.YSCLAR(K))GO TO 1180
      IF(YY.LE.YSCLAR(K-1))GO TO 1180
      IF(YY.GT.FLOAT(MAXX)) GO TO 1180
1100 CONTINUE
C
C      * * * Y COORDINATE FOUND. NOW FIND POS. OF X COOD.
      DO 1160 I1160 = 1,MHORIZ
      I = MHORIZ - I1160 + 1
C      CHECK FOR X VALUE IN RANGE
      XX = X(J)
      IF(XX.GT.XSCLAR(I)) GO TO 1160
      IF(XX.LE.XSCLAR(I-1)) GO TO 1160
      IF(XX.GT.FLOAT(MAXX)) GO TO 1180
C      MOVE ITH CHAR TO ITEST, RIGHT JUSTIFIED, 0 FILLED
      ITEST = 0
      LTEST(4) = FMTA(I)
C      CHECK FOR BLANK OR STAR IMPLYING NO PREVIOUS OCCURANCES
      IF (ITEST.EQ.RCDRLK .OR. ITEST.EQ.RCDSTR) GO TO 1140
C      CHECK FOR X IMPLYING MAXIMUM COUNT ALREADY REACHED
      IF (ITEST.EQ.RCDX) GO TO 1180
C      ADD 1 TO COUNT. IF MORE THAN 9 OCCURANCES, CHG COUNT TO X
      ITEST = ITEST + 1
      IF (ITEST.GT.RCD9) ITEST =RCDX
C      STORE UPDATED COUNT BACK IN FMT ARRAY
      FMTA(I) = LTEST(4)
      GO TO 1180
C      BLANK OR STAR, STORE 1 FOR FIRST OCCURANCE
      1140 FMTA(I) = LOG1
      GO TO 1180
      1160 CONTINUE
      1180 CONTINUE
C
C      BUILD 45 DEGREE ANGLE LINE OF PLOT
      IF (MOD(K,2).EQ.0) GO TO 1220
      NOSPAC = NOSPAC - 3
      LTEST(4) = FMTA(NOSPAC)
      IF (ITEST.NE.LHLANK) GO TO 1220
      FMTA(NOSPAC) = LSTAR
C
C      SET UP LABEL
C      1220 CONTINUE
C      DETERMINE WHETHER OR NOT LABEL IS PRINTED
      IF (K.LT.LASTLN) GO TO 1340
      IF (K.GT.INITLN) GO TO 1340
      IF (K.NE.WICLAR(NDX)) GO TO 1340
      IF (K.EQ.52.OR.K.EQ.40) GO TO 1320
C
C      MOVE LABARY TO FMTARY
      IJ = FMDX + 5
      IF (MOD(K,2).EQ.0) IJ =FMDX + 5
      DO 1240 II=FMDX,IJ
      FMTARY(II) = LABARY(LABNDX)
      LABNDX = LABNDX + 1
1240 CONTINUE
      NDX = NDX + 1
      GO TO 1340
1320 CONTINUE
      FMTARY(FMDX+3)=IHLANK
      NDX = NDX + 1
1340 CONTINUE
      IF (K.EQ.MLNCT) GO TO 1380
      IF (MOD(K+7,8).EQ.0) GO TO 1380
      WRITE (6,1560)FMTARY
      GO TO 1440
```

PL001590  
PL001600  
PL001610  
PL001620  
PL001630  
PL001640  
PL001650  
PL001660  
PL001670  
PL001680  
PL001690  
PL001700  
PL001710  
PL001720  
PL001730  
PL001740  
PL001750  
PL001760  
PL001770  
PL001780  
PL001790  
PL001800  
PL001810  
PL001820  
PL001830  
PL001840  
PL001850  
PL001860  
PL001870  
PL001880  
PL001890  
PL001900  
PL001910  
PL001920  
PL001930  
PL001940  
PL001950  
PL001960  
PL001970  
PL001980  
PL001990  
PL002000  
PL002010  
PL002020  
PL002030  
PL002040  
PL002050  
PL002060  
PL002070  
PL002080  
PL002090  
PL002100  
PL002110  
PL002120  
PL002130  
PL002140  
PL002150  
PL002160  
PL002170  
PL002180  
PL002190  
PL002200  
PL002210  
PL002220  
PL002230  
PL002240  
PL002250  
PL002260  
PL002270  
PL002280  
PL002290  
PL002300  
PL002310  
PL002320  
PL002330  
PL002340  
PL002350  
PL002360  
PL002370

FILE: PLOT

```
1560 FORMAT (1H ,22X,1H*,23A4)
1380 CONTINUE
C PRINT PLOT FOR 1108
  IF (M.LE.1) GO TO 1480
  IF (K.FQ.MLNCT) WRITE (6,1660)FMTARY
1660 FORMAT(1H ,21X,2H *,23A4)
  IF (K.FQ.MLNCT) GO TO 1440
  WRITE (6,1540)SCLARY(M),FMTARY
1540 FORMAT(1H ,14X,E7.2,2H *,23A4)
  M = M - 1
1440 CONTINUE
C
C WRITE X AND Y LABELS AS SELECTED
DO 1460 I1460=1,INDEX
  FMTARY(I1460)=IBLANK
1460 CONTINUE
  IF (K.NF.40)GO TO 1030
  IF (PRCKEY.EQ.3 .OR. PRCKEY.EQ.4)WRITE(6,1710)
1710 FORMAT(1H*,*LINEAR*)
  IF (PRCKEY.EQ.1 .OR. PRCKEY.EQ.2)WRITE(6,1720)
1720 FORMAT(1H*,*SELECTED*)
1030 IF (K.NF.38)GO TO 1035
  IF (PRCKEY.EQ.3 .OR. PRCKEY.EQ.4) WRITE(6,1730)
1730 FORMAT(1H*,*COMB.*)
  IF (PRCKEY.EQ.1 .OR. PRCKEY.EQ.2) WRITE(6,1740)
1740 FORMAT(1H*,*CHANNELS*)
1035 CONTINUE
1480 CONTINUE
C
C * * * PRINT X-AXIS LINE AND LABELS
C * * * 1108 SIZE PLOT
  WRITE (6,1580)(SCLARY(L),L=1,8)
1580 FORMAT(1H ,22X,7(12H* * * * * ),1H*/15X,E7.2,3X,7(5X,E7.2))
  WRITE(6,1750)
1750 FORMAT(/60X,*ALL CHANNELS*)
  IF (CRIKEY.EQ.3)WRITE(6,1690)
1690 FORMAT(/50X,*INTERCLASS BHATTACHARYYA DISTANCE*)
  IF (CRIKEY.NF.3)WRITE(6,1700)
1700 FORMAT(/55X,*PAIRWISE DIVERGENCE*)
  RETURN
  END
```

PL002380  
PL002390  
PL002400  
PL002410  
PL002420  
PL002430  
PL002440  
PL002450  
PL002460  
PL002470  
PL002480  
PL002490  
PL002500  
PL002510  
PL002520  
PL002530  
PL002540  
PL002550  
PL002560  
PL002570  
PL002580  
PL002590  
PL002600  
PL002610  
PL002620  
PL002630  
PL002640  
PL002650  
PL002660  
PL002670  
PL002680  
PL002690  
PL002700  
PL002710  
PL002720  
PL002730  
PL002740  
PL002750  
PL002760  
PL002770  
PL002780  
PL002790  
PL002800

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: PRELIM

```
* SURROUTINE PRELIM(COVMTX,AVEMTX,DIVTAB,WEIGHT,S,
  WRKRY,WRKSIZ)
C*
C* THIS SUBROUTINE PERFORMS SOME OF THE PRELIMINARY TASKS FOR
C* FEATURE SELECTION. THE INTERCLASS MEASURES USING ALL FEATURES
C* ARE COMPUTED AND STORED ON A SCRATCH FILE FOR LATER PRINTING.
C*
C* IN ADDITION, THIS SUBROUTINE COMPUTES THE 'S' MATRICES USED IN
C* COMPUTING WEIGHTED AVERAGE DIVERGENCE IF CRIKEY=1. IF WEIGHTS
C* ARE TO BE SET BY DEFAULT, THE SUBROUTINE ALSO PERFORMS THIS TASK.
C
C IMPLICIT INTEGER(A-Z)
C INCLUDE COMPK1.LIST
C INCLUDE COMPK7.LIST
COMMON/INFORM/NOCLS2,NOSURP,NOFET2,VAR522,TOTVT2,NOFLD2,
  AVAR2,COVAR2,CLSID2,SURN02,SURNS2,FLDSV2,VERTX2,
  FETVC2(30),SIHVC2(75),SURPTR(75),CLSV2(60),
  KPPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
  GRPCHK(61),GROUPS(124)
COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
  INCVEC(30),ICOUNT,SETWGT,EVALRF(100),FETVC4(30),
  NOFFT4,VAR574,CORBAS,DTAH4,WGHS14,RESTVC(10),DIVSIZ
  ,STATKY,ADRES0,ADHESP,ADRESF,ADRSH1,ADRSH2
INTEGER ADRES0,ADRESF,ADRSF,ADRSH1,ADRSH2,STATKY
DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C*END
COMMON/HESTKN/ KPPPTS(60), IPRIOR, KREST, NCPASS
REAL COVMTX(VARS22,NOCLS2), AVEMTX(NOFET2,NOCLS2),
  WEIGHT(DIVSIZ), S(VARS22,NOCLS2)
REAL T(30),APRWGT(750),ANUMPX
DOUBLE PRECISION DIVTAB(DIVSIZ),DUM
DOUBLE PRECISION WRKRY(1)
REAL TW(60)
ANUMPX = 0.
DO 1 I=1,NOCLS2
1 ANUMPX = ANUMPX + FLOAT(KPPPTS(I))
  K = 0
  NC = NOCLS2 - 1
  DO 2 I=1,NC
  IK = I + 1
  DO 2 II = IK,NOCLS2
  K = K + 1
  APRWGT(K) = FLOAT(KPPPTS(I)*KPPPTS(II))/(ANUMPX**2)
  APRWGT(K) = SQRT(APRWGT(K))
2 IF (IPRIOR.NE.0) WRITE(6,990)
990 FORMAT(1H1,' IPRIORI WEIGHT MULTIPLIERS AND TOTAL NO PIXELS')
1000 IF (IPRIOR.NE.0) WRITE(6,1000) (APRWGT(I),I=1,DIVSIZ),ANUMPX
  FORMAT( )
  IF (SETWGT.F0.2) GO TO 6
  DO 5 I=1,DIVSIZ
  WEIGHT(I) = 1.0
5 IF (IPRIOR.F0.0) GO TO 9
6 DO 7 I=1,DIVSIZ
  WEIGHT(I) = WEIGHT(I)*APRWGT(I)
7 CONTINUE
9 SET IPART SO PARTIALS WILL NOT BE COMPUTED.
C* IPART=-1
  IF(ILL=1)
  GO TO(10,70,80,90),CRIKEY
C*
C* CRITERIA - WEIGHTED AVERAGE DIVERGENCE
C* --COMPUTE INTERCLASS DIVERGENCES
C* --SET WEIGHTS, IF SETWGT=0
C* --COMPUTE S-MATRICES
C* --COMPUTE WEIGHTED AVERAGE DIVERGENCE FOR ALL FEATURES
C*
10 CALL DIVERG(COVMTX,VAR522,AVEMTX,DIVTAB,NOCLS2,NOFET2,
  WRKRY,WRKSIZ)
  IF (SETWGT.NE.0) GO TO 25
  DO 20 K=1,DIVSIZ
  WEIGHT(K) = DEXP(-DIVTAB(K)/16.)
  IF (IPRIOR.NE.0) WEIGHT(K) = WEIGHT(K)*APRWGT(K)
20 COMPUTE S-MATRICES
C*
25 CONTINUE
  DO 30 J=1,NOCLS2
  DO 30 I=1,VAR572
  S(I,J) = 0.0
30 NC=NOCLS2-1
  DO 60 N=1,NOCLS2
```

FILE: PRELIM

```
C*
C*
C*      SELECT ALL WEIGHTS FOR CLASS N
      KT=0
      K=0
      MN=0
      DO 35 J=1,NC
      IJ=J+1
      DO 35 I=IJ,NOCLS2
      K=K+1
      IF (J.NE.N.AND.I.NE.N)GO TO 35
      KT=KT+1
      TW(KT)=WEIGHT(K)
35 CONTINUE
      DO 50 M=1,NOCLS2
      IF (M.EQ.N)GO TO 50
      MN=MN+1
      DO 40 I=1,NOFET2
40 T(I)=AVEMTX(I,N)-AVEMTX(I,M)
      J=0
      DO 45 I=1,NOFET2
      DO 45 K=1,I
      J=J+1
45 S(J,N)=S(J,N)+TW(MN)*(COVMTX(J,M)+T(I)*T(K))
50 CONTINUE
60 CONTINUE
C*      COMPUTE CFAC
      CFAC=0
      DO 65 I=1,DIVSIZ
45 CFAC = CFAC + WEIGHT(I)
      CFAC=1./CFAC
C*      COMPUTE AVERAGE WEIGHTED DIVERGENCE
      CALL AVEDIV(TOTMSR,COVMTX,S,DUM,DUM,WRKRY,WRKSIZ,
      *      IPART,DUM,DUM,IFULL)
      TOTMSR=DABS(TOTMSR)
      GO TO 85
C*
C*
C*      CRITERIA - WEIGHTED AVERAGE TRANSFORMED DIVERGENCE
C*
C*
70 CALL TRNDIV(TOTMSR,COVMTX,AVEMTX,DUM,DUM,WEIGHT,DIVTAB,
      *      WRKRY,WRKSIZ,IPART,DUM,DUM,IFULL)
      GO TO 85
C*
C*
C*      CRITERIA - BHATTACHARYYA DISTANCE
C*
80 CALL BHTCHR(TOTMSR,COVMTX,AVEMTX,WEIGHT,DIVTAB,DUM,DUM,
      *      WRKRY,WRKSIZ,IPART,DUM,DUM,IFULL)
C*
85 SAVE INTERCLASS WEIGHTS ON DRUM
      IQ=DIVSIZ*2
      CALL KWRITE(ADRESO,DIVTAB,IQ,ISTAT)
86 IF(ISTAT.EQ.1)GO TO 86
90 RETURN
      END
```

PRE00810  
PRE00820  
PRE00830  
PRE00840  
PRE00850  
PRE00860  
PRE00870  
PRE00880  
PRE00890  
PRE00900  
PRE00910  
PRE00920  
PRE00930  
PRE00940  
PRE00950  
PRE00960  
PRE00970  
PRE00980  
PRE00990  
PRE01000  
PRE01010  
PRE01020  
PRE01030  
PRE01040  
PRE01050  
PRE01060  
PRE01070  
PRE01080  
PRE01090  
PRE01100  
PRE01110  
PRE01120  
PRE01130  
PRE01140  
PRE01150  
PRE01160  
PRE01170  
PRE01180  
PRE01190  
PRE01200  
PRE01210  
PRE01220  
PRE01230  
PRE01240  
PRE01250  
PRE01260  
PRE01270  
PRE01280  
PRE01290  
PRE01300  
PRE01310  
PRE01320  
PRE01330  
PRE01340

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: PRTFLD

```

      SUPROUTINE PRTFLD(COVMTX,AVEMTX,FLDMTX,VERTEX,
      CLSNAM,SUBNAM)
      IMPLICIT INTEGER (A-H,O-Z)
      PRINT TRAINING FIELDS AND CLASS STATISTICS

      INCLUDE COMRK1.LIST
      DATA LPRN/ICH(1/,RPRN/)/
      INCLUDE COMRK6.LIST
      INCLUDE COMRK7.LIST
      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARZ2,TOTVT2,NOFLD2,
      AVAR2,COVAR2,CLS1D2,SUBNO2,SURDS2,FLOSV2,VERTX2,
      FETVC2(30),SURVC2(75),SURPTR(75),CLSV2(60),
      KEPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),
      GRPCHK(61),GROUPS(124)
      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAE,SAVTAP,RMFILE,RMKEY,
      HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
      DRUMAD,DRMWDOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
      .NHSTUN,NHSTFI,SCRUN,MAPEIL
      .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
      CRDUNT,PRUNT,RANDIO
      COMMON/FSL/CFAC,TOTMSR,SEPMSR,PCKEY,CRIKEY,INCFET,
      INCVFC(10),ICOUNT,SETWGT,EVALBF(100),FETVC4(30)
      .NOFET4,VARZ4,CORHAS,DAT4,WGHS14,RESTVC(10),DIVSIZ
      .STATKY,ADRESU,ADRESP,ADRESF,ADRS1,ADRS2,STATKY
      INTEGER ADRESU,ADRESP,ADRESF,ADRS1,ADRS2,STATKY
      DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
      DIMENSION COVMTX(VARZ2,NOSUB2),AVEMTX(NOFET2,NOSUB2),
      FLDMTX(4,NOFLD2),VERTEX(2,TOTVT2),
      CLSNAM(NOCLS2),SUBNAM(NOSUB2)
      DATA ONE/1/,SCFSZ2/3600/,RCDTWO/12/

      WRITE OUT TRAINING FIELDS
      -----
      CALL WRTFLD(FLDMTX,VERTEX,NOFLD2,1,CLSNAM,SUBNAM)
      -----
      PRINT THE COVARIANCE AND MEAN
      -----
210 IF (STATKY.F0.0) GO TO 300
      CNT = 7*(5+3+2*NOFET2)*((NOFET2+11)/12)
      CNT = PAGSIZ/CNT
      INC = CNT

      DO 290 ICLAS = 1,NOSUB2
      IF (INC.LT.CNT) GO TO 220
      WRITE (6,HEAD)
      INC = 0
220 WRITE (6,230) SUBNAM(ICLAS)
230 FORMAT (/1X,'SUBCLASS ',A4 )
      DO 240 LOC=1,NOFET2+12
      STOP = LOC+11
      IF ( STOP .GT. NOFET2 ) STOP = NOFET2
      WRITE (6,250) (LPRN,FETVC2(I),RPRN,I=LOC,STOP)
240 WRITE (6,260) (AVFMTX(I,ICLAS),I=LOC,STOP)
250 FORMAT (10X,12(A3,I2,A1,3X))
260 FORMAT ('0MEAN',3X,12F9.2)
      WRITE (6,240)
270 FORMAT (10 COVARIANCE MATRIX)
      CALL WRTMTX(COVMTX(1,ICLAS),NOFET2,RCDTWO)
      INC = INC+1
290 CONTINUE

300 CONTINUE
      RETURN
      END

```

PRT00010  
PRT00020  
PRT00030  
PRT00040  
PRT00050  
PRT00060  
PRT00070  
PRT00080  
PRT00090  
PRT00100  
PRT00110  
PRT00120  
PRT00130  
PRT00140  
PRT00150  
PRT00160  
PRT00170  
PRT00180  
PRT00190  
PRT00200  
PRT00210  
PRT00220  
PRT00230  
PRT00240  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00060  
PRT00320  
PRT00330  
PRT00340  
PRT00350  
PRT00360  
PRT00370  
PRT00380  
PRT00390  
PRT00400  
PRT00410  
PRT00420  
PRT00430  
PRT00440  
PRT00450  
PRT00460  
PRT00470  
PRT00480  
PRT00490  
PRT00500  
PRT00510  
PRT00520  
PRT00530  
PRT00540  
PRT00550  
PRT00560  
PRT00570  
PRT00580  
PRT00590  
PRT00600  
PRT00610  
PRT00620  
PRT00630  
PRT00640  
PRT00650  
PRT00660  
PRT00670  
PRT00680  
PRT00690  
PRT00700  
PRT00710  
PRT00720  
PRT00730  
PRT00740  
PRT00750

FILE: SCALE

```

      SURROUTINE SCALE(MAXX,MLNCT,INCRE,YSCLAR,XSCLAR,SCLARY,XLNVLU,
      YLNVLU,MMHORIZ,NOXPT,NOYPT)
C
      INCLUDE COMR7.LIST
      COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
      INCVFC(30),ICOUNT,SETWGT,EVALRF(100),FFTVC4(30)
      .NOFET4,VAR574,CORHAS,DTAH4,NGHS14,RESTVC(10),DIVSIZ
      .STATKY,ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2
      INTEGER ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY
      DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C$END
      INTEGER CRIKEY
      REAL YSCLAR(MLNCT), XSCLAR(MMHORIZ),XLNVLU,YLNVLU,RSAVE,XPTS,YPTS
      DIMENSION SCLARY(R)
C * * * ZF40 OUT SCALE LABEL ARRAY - SCLARY
      DO 1000 I=1,M
      SCLARY(I) = 0
      1000 CONTINUE
      IF(CRIKEY.EQ.3 .OR. CRIKEY.EQ.4)GO TO 1070
C * * * DETERMINE LABELS FOR X AND Y AXIS
      INCR = MAXX/7
      IF (MOD(INCR,5) .NE. 0) INCR = 5 - MOD(INCR,5) + INCR
      ISAVE = INCR
      DO 1020 I=2,M
      SCLARY(I) = ISAVE
      ISAVE = INCR + ISAVE
      1020 CONTINUE
C * * * DETERMINE THE VALUE OF EACH POINT ON THE X AND Y AXIS
      XPTS = NOXPT
      YPTS = NOYPT
      RSAVE = INCR
      XLNVLU = RSAVE/XPTS
      YLNVLU = RSAVE/YPTS
      YSCLAR(1) = 0.0
      XSCLAR(1) = 0.0
      RSAVE = YLNVLU
      DO 1040 I=2,MLNCT
      YSCLAR(I) = RSAVE
      RSAVE = RSAVE + YLNVLU
      1040 CONTINUE
      RSAVE = XLNVLU
      DO 1060 I=2,MMHORIZ
      XSCLAR(I) = RSAVE
      RSAVE = RSAVE + XLNVLU
      1060 CONTINUE
      RETURN
      1070 CONTINUE
      XPTS=NOXPT
      YPTS=NOYPT
      IEXP=7
      DO 1075 I=1,8
      SCLARY(I) = 1. / 10.**IEXP
      1075 IEXP = IEXP-1
      K=0
      DO 1080 I=1,7
      K=K+1
      RNCRE = (SCLARY(I+1) - SCLARY(I))/YPTS
      YSCLAR(K)=SCLARY(I)
      DO 1090 J=2,NOYPT
      K=K+1
      1080 YSCLAR(K)=YSCLAR(K-1) + RNCRE
      K=0
      DO 1090 I=1,7
      K=K+1
      XSCLAR(K)=SCLARY(I)
      RNCRE = (SCLARY(I+1)-SCLARY(I))/XPTS
      DO 1090 J=2,NOXPT
      K=K+1
      1090 XSCLAR(K)=XSCLAR(K-1)+RNCRE
      1090 CONTINUE
      RETURN
      END

```

SCA00010  
 SCA00020  
 SCA00030  
 SCA00040  
 SCA00050  
 SCA00060  
 SCA00070  
 SCA00080  
 SCA00090  
 SCA00100  
 SCA00110  
 SCA00120  
 SCA00130  
 SCA00140  
 SCA00150  
 SCA00160  
 SCA00170  
 SCA00180  
 SCA00190  
 SCA00200  
 SCA00210  
 SCA00220  
 SCA00230  
 SCA00240  
 SCA00250  
 SCA00260  
 SCA00270  
 SCA00280  
 SCA00290  
 SCA00300  
 SCA00310  
 SCA00320  
 SCA00330  
 SCA00340  
 SCA00350  
 SCA00360  
 SCA00370  
 SCA00380  
 SCA00390  
 SCA00400  
 SCA00410  
 SCA00420  
 SCA00430  
 SCA00440  
 SCA00450  
 SCA00460  
 SCA00470  
 SCA00480  
 SCA00490  
 SCA00500  
 SCA00510  
 SCA00520  
 SCA00530  
 SCA00540  
 SCA00550  
 SCA00560  
 SCA00570  
 SCA00580  
 SCA00590  
 SCA00600  
 SCA00610  
 SCA00620  
 SCA00630  
 SCA00640  
 SCA00650  
 SCA00660  
 SCA00670  
 SCA00680  
 SCA00690  
 SCA00700  
 SCA00710

FILE SETUP4

```

C      SURROUTINE SETUP4 (ARRAY, TOP, STOPFG, JTIME, SUBRAY, SUBSIZ)
      IMPLICIT INTEGER (A-H, O-Z)
-----
      PURPOSE.. COORDINATES ROUTINES TO ANALYZE SUPERVISOR
                CONTROL CARDS FOR 'SELECT' PROCESSOR
      RETURNS.. SUPERVISOR INFORMATION AND REDUCED STATISTICS
-----

      INCLUDE COMBK1.LIST
      INCLUDE COMBK4.LIST
      DIMENSION COOVEC(23), CARD(20)
      DIMENSION WGHBIIF(400), NUMVEC(30), COMVEC(2)
      DIMENSION EQUVEC(2)
      DATA EQUVEC/1,1//
      DATA COOMAX /23/
      DATA WSIZ/400/, ESIZ/100/, BLANK/1H /, COMVEC/1,1//
      DATA SYMM4/60/
      DATA COMVEC/ 'CHAN', 'SURC', 'BEST', 'OPTI', 'WEIG', 'EVAL', 'MODU',
2        'THRU', 'PROC', 'CRIT', 'R-MA', 'INCL', 'ICOU', 'DATE', 'HED1',
3        'HED2', 'COM', 'END', 'SEND', 'STAT', 'APRI', 'BSPA', 'NCPA' /
      INCLUDE COMBK6.LIST
      INCLUDE COMBK7.LIST
      COMMON/INFORM/ NOCLS2, NOSUB2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
      AVAR2, COVAR2, CLSID2, SUBNO2, SUH092, FLDSV2, VERTX2,
      FEIVC2(30), SURVC2(75), SURPTR(75), CLSVC2(60),
      KFPPTS(60), NUGMP, GRPNAM(60), GRPDEX(61),
      GMPCHK(61), GROUPS(124)
      DIMENSION HED1(15), HED2(15), DATE(3), COMENT(15)
      EQUIVALENCE (HED1(1), HEAD(4)), (DATE(1), HEAD(22)),
      (HED2(1), HEAD(30)), (COMENT(1), HEAD(48))
      COMMON/GLORAL/ HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
      H1SFIL, H1SKY, TRFORM, ER1PT, ERPKY, MAPUNT, NOFILE,
      DRUMAD, DRMND, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
      NHSTUN, NHSTFL, SCTRUN, MAPFIL
      DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, H1STFL, PCHUNT,
      CRDUNT, PRUNT, MANDIO
      COMMON/FSL/ CFAC, TOTMSR, SEPMSR, PRCKY, CRPKY, INCFET,
      INCVEC(30), ICOUNT, SETXGT, EVALHF(100), FEIVC4(30)
      NOFET4, VARSZ4, COMHAS, DTAH4, WGHM14, BESTVC(10), DIVSIZ
      STATKY, ADRES0, ADRESF, ADRESF, ADRESH1, ADRESH2
      INTEGER ADRES0, ADRESF, ADRESF, ADRESH1, ADRESH2, STATKY
      DOUBLE PRECISION CFAC, TOTMSR, SEPMSR
CSEND  COMMON/DESTKN/ KFPPTS(60), IPRIOR, KBEST, NCPASS
      DIMENSION PROC(3,4), CHI(3,3)
      DATA CHI/ 'DIVE', 'NGEN', 'CE', 'TRAN', 'S. D', 'IV', 'BHAT', 'T. D',
      'IST' //
      DATA PROC/ EX, 'SFA', 'RCH', 'WITH', 'OUT', 'RPLC',
      'DAVI', 'NON', 'EVL', 'B MA', 'TRIX', 'EVAL', 'UATE' //
      'RST', 'K', 'PASS' //
      DIMENSION ARRAY(1), CARD(62)
      DIMENSION SUBRAY(1)
      REAL SUBRAY
      DATA CHCD/'C'//, SB CD/'S'//, PHCD/'P'//, UB CD/'U'//, FB CD/'F'//
      DATA HB CD/'R'//
-----
      NOGRP=0
-----

```

SET00010  
 SET00020  
 SET00030  
 SET00040  
 SET00050  
 SET00060  
 SET00070  
 SET00080  
 SET00090  
 SET00100  
 SET00110  
 SET00120  
 SET00130  
 SET00140  
 SET00150  
 SET00160  
 SET00170  
 SET00180  
 SET00190  
 SET00200  
 SET00210  
 SET00220  
 SET00230  
 SET00240  
 SET00250  
 SET00260  
 SET00270  
 SET00280  
 SET00290  
 SET00300  
 SET00310  
 SET00320  
 SET00330  
 SET00340  
 SET00350  
 SET00360  
 SET00370  
 SET00380  
 SET00390  
 SET00400  
 SET00410  
 SET00420  
 SET00430  
 SET00440  
 SET00450  
 SET00460  
 SET00470  
 SET00480  
 SET00490  
 SET00500  
 SET00510  
 SET00520  
 SET00530  
 SET00540  
 SET00550  
 SET00560  
 SET00570  
 SET00580  
 SET00590  
 SET00600  
 SET00610  
 SET00620  
 SET00630  
 SET00640  
 SET00650  
 SET00660  
 SET00670  
 SET00680  
 SET00690  
 SET00700  
 SET00710  
 SET00720  
 SET00730  
 SET00740  
 SET00750  
 SET00760







FILE SETUP4

```

C* 170 J=NUMBER(CARD,COL,NUMVEC,0)
      ICOUNT=NUMVEC(1)
      GO TO 10
C*
C* 180 DATE CARD
      READ(30,6000)DATE
      REWIND RRUNIT
      GO TO 10
C*
C* 190 HED1 CARD
      READ(30,6000)HED1
      REWIND RRUNIT
      GO TO 10
C*
C* 200 HED2 CARD
      READ(30,6000)HED2
      REWIND RRUNIT
      GO TO 10
C 205 APRIORI CARD
      IPRIOR = 1
      GO TO 10
C* 207 J = NUMBER(CARD,COL,NUMVEC,0)
      KREST = NUMVEC(1)
      GO TO 10
C* 208 J=NUMBER(CARD,COL,NUMVEC,0)
      NCPASS=NUMVEC(1)
      GO TO 10
C*
C* 210 COMMENT CARD
      READ(30,6000)COMENT
      REWIND RRUNIT
      GO TO 10
C*
C* STAT FILE NO.
C*
C* 215 M = NXTCHR(CARD ,COL)
      IF(M.EQ.HLANK) GO TO 10
      IF(M.EQ.URCD) GO TO 1702
      IF(M.EQ.FRCD) GO TO 1703
C* 1723 WRITE(6,755)
C* 755 FORMAT(' ERROR ON STAT FILE CARD *')
      GO TO 10
C* 1702 J=FIND12(CARD ,COL,EQUVEC)
      IF(J.EQ.-1) GO TO 1723
      M=NUMBER(CARD ,COL,SAVTAP,ZERO)
      COL=COL-1
      GO TO 215
C* 1703 J=FIND12(CARD ,COL,EQUVEC)
      IF(J.EQ.-1) GO TO 1723
      FILNO = NUMBER(CARD ,COL,STAFIL,FILNO)
      STAFIL = STAFIL - 1
      COL=COL-1
      GO TO 215
C*
C* *END* - END OF THIS SET OF CONTROL CARDS
C* - - GET STATS AND FETVEC INTO CORE
C* 220 CONTINUE
C*
C* IF R-MATRIX IS INPUT, OBTAIN DIMENSIONING INFORMATION AND
C* FFTVC2 FROM RMFILE.
C* 225 IF(RMSWT.EQ.0)GO TO 230
      CALL RMFIL (DUMMY,NOFET4,NOFET2,FETVC2,3)
C*
C* READ AND REDUCE STATS
C*
C* 230 CALL REDSAV (ARRAY, TOP, RMSWT)
C*
C* CODE ADDED TO CHECK FOR EXIT FOR ONE CLASS INPUT
C* ON PROCEDURES 1,2,3,OR 6
C*
C* NOFFT4=NOFET2
C* DO 240 I=1,NOFFT2
C* FFTVC4(I)=FFTVC2(I)
C* 240 CONTINUE
C

```

```

SET02290
SET02300
SET02310
SET02320
SET02330
SET02340
SET02350
SET02360
SET02370
SET02380
SET02390
SET02400
SET02410
SET02420
SET02430
SET02440
SET02450
SET02460
SET02470
SET02480
SET02490
SET02500
SET02510
SET02520
SET02530
SET02540
SET02550
SET02560
SET02570
SET02580
SET02590
SET02600
SET02610
SET02620
SET02630
SET02640
SET02650
SET02660
SET02670
SET02680
SET02690
SET02700
SET02710
SET02720
SET02730
SET02740
SET02750
SET02760
SET02770
SET02780
SET02790
SET02800
SET02810
SET02820
SET02830
SET02840
SET02850
SET02860
SET02870
SET02880
SET02890
SET02900
SET02910
SET02920
SET02930
SET02940
SET02950
SET02960
SET02970
SET02980
SET02990
SET03000
SET03010
SET03020
SET03030
SET03040

```

FILE SETUP4

```

IF(NOCLS2.GT.1)GO TO 250
IF(PRCKEY.EQ.4.OR.PRCKEY.EQ.5)GO TO 250
IF(RUNKEY.EQ.1)GO TO 250
WRITE(6,9000)
GO TO 10
CODE ADDED TO QUIT IF ONLY ONE CLUSTER INPUT
250 IF(NOSUR2.GT.1)GO TO 260
WRITE(6,9400)
9400 FORMAT(5X,'PROGRAM CANNOT PROCESS ONLY ONE CLUSTER INPUT')
GO TO 10
CHECK 'BEST' REQUESTS
260 IF(NOBEST.EQ.0)GO TO 270
CALL BSTCHK(NOBEST)
IF(NOBEST.GT.0)GO TO 280
270 NOBEST=1
BESTVC(NOBEST)=0
CHECK ON EVALUATE REQUESTS
280 CONTINUE
EPTR=EPTR+1
IF(PRCKEY.EQ.6) BESTVC(1) = KBEST
EVALRF(EPTR)=0
PRINT USER REQUESTS
WRITE(6,9300) (PROC(M,PRCKEY),M=1,3),
(CRI(M,CRIKEY),M=1,3)
WRITE(6,9310) (BESTVC(I),I=1,NOBEST)
WRITE(6,9320) (FETVC2(I),I=1,NOFET2)
IF(INCFET.GT.0)WRITE(6,9330) (INCVEC(I),I=1,INCFET)
IF(SETWGT.EQ.2)WRITE(6,9340)
IF(WTKEY.EQ.1) WRITE(6,9360)
IF(SETWGT.NE.2.AND.WTKEY.NE.1)WRITE(6,9350)
IF(IPRIOR.NE.0) WRITE(6,9370)
WRITE(6,9380) NCPASS
PRINT OUT SAVED TRAINING FIELDS AND REDUCED COVARIANCES.
CALL PRFLD(ARRAY(COVAR2),ARRAY(AVAR2),ARRAY(FLDSV2),
ARRAY(VRTX2),ARRAY(CLSID2),ARRAY(SUBDS2))
IF CLSWT OPTION IS INPUT, SET UP WEIGHT ARRAY FOR INTERCLASS
SUBCLASS WEIGHTS.
W1 = 1 * WPTR * 4
285 IF(WTKEY.EQ.1) GO TO 9500
STORAGE FOR FIELD INFORMATION NO LONGER NEEDED.
MOVE CLASS ID INFORMATION
AND MEANS AND COVARIANCES.
---REDSAV STORES INTO ARRAY IN THE FOLLOWING ORDER
1.CLASS DESCRIPTIONS
2.NO. OF SURCLASSES IN EACH CLASS
3.SURCLASS DESCRIPTIONS
4.TRAINING FIELD INFO.
5.TRAINING FIELD VERTICES
6.COVARIANCE MATRICES
7.MEAN VECTORS
ITFMS 1,2,4,5 ARE NOT NEEDED AFTER PRINTING, SO THE STORAGE
IS REASSIGNED AS FOLLOWS, ADDING STORAGE FOR OTHER ARRAYS.
1. SURCLASS DESCRIPTIONS
2. COVARIANCE MATRICES
3. MEAN VECTORS
4. INTER(SUR)CLASS WEIGHTS
5. INTER(SUB)CLASS SEPARABILITY MEASURE TABLE
IN SELECT DRIVER 'ARRAY' IS ALSO USED TO STORE ADDITIONAL INFO.
287 CONTINUE
DO 290 I=1,NOSUR2
290 ARRAY(CLSID2+I-1) = ARPAY(SUBDS2+I-1)

```

SET03050  
SET03060  
SET03070  
SET03080  
SET03090  
SET03100  
SET03110  
SET03120  
SET03130  
SET03140  
SET03150  
SET03160  
SET03170  
SET03180  
SET03190  
SET03200  
SET03210  
SET03220  
SET03230  
SET03240  
SET03250  
SET03260  
SET03270  
SET03280  
SET03290  
SET03300  
SET03310  
SET03320  
SET03330  
SET03340  
SET03350  
SET03360  
SET03370  
SET03380  
SET03390  
SET03400  
SET03410  
SET03420  
SET03430  
SET03440  
SET03450  
SET03460  
SET03470  
SET03480  
SET03490  
SET03500  
SET03510  
SET03520  
SET03530  
SET03540  
SET03550  
SET03560  
SET03570  
SET03580  
SET03590  
SET03600  
SET03610  
SET03620  
SET03630  
SET03640  
SET03650  
SET03660  
SET03670  
SET03680  
SET03690  
SET03700  
SET03710  
SET03720  
SET03730  
SET03740  
SET03750  
SET03760  
SET03770  
SET03780  
SET03790  
SET03800

PAGE IS  
OF POOR QUALITY

FILE SETUP4

```

SURDS2=CLSID2
NA=SUBDS2 * NOSUR2
IWRDS = (VARSZ2*NOFET2)*NOSUB2
00 295 I=1,IWRDS
295 ARRAY(NA+I-1) = ARRAY(COVAR2+I-1)
COVAR2=NA
AVAR2=COVAR2 * NOSUB2*VARSZ2
C*
C** FROM HTER ON THROUGH SELECT SUBCLASSES ARE REFERRED TO AS CLASSES
C**
NOCLS2=NOSUB2
C*
C** COMPUTE BASES FOR OTHER ARRAYS.
C**
DIVSZ=NOCLS2*(NOCLS2-1)/2
WGHS14=AVAR2 * NOFET2*NOCLS2
DTAB4 = WGHS14 * DIVSZ
CORBAS = DTAB4 * DIVSZ*2
IC=CORBAS
IF(CORBAS.LT.TOP)GO TO 300
WRITE(6,9100)IC
CALL CMERR
C*
C** SET UP ARRAY OF INTERCLASS WEIGHTS IF INPUT - IF DEFAULT IS TAKEN
C** WEIGHTS ARE COMPUTED IN PRELIM.
C**
300 IF(SETWGT.NE.2.AND.WTKEY.NE.1) GO TO 310
C* PASS KEYS TO SUBROUTINE IN ALREADY EXISTING STORAGE
ARRAY(WGHS14) = SETWGT
NT = WGHS14 * 1
ARRAY(NT) = WTKEY
CALL WGTCHK(ARRAY(WGHS14),ARRAY(CLSID2),SUBRAY,WGMBUF,WPTR,
SUBRAY(W1),NOCLS2)
SETWGT = 2
310 CONTINUE
RETURN
C*
C** SEND* CARD
C**
350 STOPFG=1
RETURN
1000 FORMAT(' $SELECT')
2000 FORMAT(A4,5X,62A1)
3000 FORMAT(5X,A4,6X,62A1)
4000 FORMAT(' TOO MANY EVALUATE REQUESTS--REMAINDER IGNORED')
5000 FORMAT(' GROUP CARD IN ERROR - IGNORED')
6000 FORMAT(10X,15A4)
7000 FORMAT(' PROGRAM CANNOT PROCESS LESS THAN 2 CHANNELS')
8000 FFORMAT(' INVALID CONTROL CARD - IGNORED')
9000 FORMAT(' PROGRAM CANNOT PROCESS LESS THAN 2 CLASSES')
9100 FORMAT(' CORE NEEDED IN ARRAY FOR THIS PROBLEM IS',I6,' WORDS')
9200 FORMAT(' ERROR IN ATTEMPT TO READ STATISTICS FILE-EXECUTION TERMIN
*ATED FROM SETUP4')
9300 FORMAT(' YOU HAVE SELECTED THE FOLLOWING OPTIONS:/'
5X,'PROCEDURE',T35,3A4/ 5X,'CRITERIA',T35,3A4)
9310 FORMAT(5X,'SELECT THE BEST SET(S) OF',T35,10(I2,','))
9320 FORMAT(5X,'FROM CHANNELS',T35,30(I2,','))
9340 FORMAT(5X,'USE INPUT WEIGHTS')
9350 FORMAT(5X,'USE DEFAULT WEIGHTS')
9330 FORMAT(5X,'INCLUDE IN THE BEST SET, CHANNELS',30(I2,','))
9360 FORMAT(5X,'USE AUTOMATIC INTERCLASS SUBCLASS WEIGHTS')
9370 FORMAT(5X,'USE APRIORI WEIGHTING TO MODIFY INTERSUBCLASS WTS')
9380 FORMAT(5X,'NUMBER CHANNELS PER PASS IS',T35,I5)
C* INITIALIZE ALL SUBCLASS WEIGHT PAIRS TO 0.0 IN WORKING ARRAY
9500 CONTINUE
DO 2100 IK=1,NOSUR2
DO 2100 JK=1,NOSUB2
IIDUM=(W1+IK-1+(JK-1)*NOSUB2)
IIDUM=(W1+JK-1+(IK-1)*NOSUB2)
SUBRAY(IIDUM)=0.0
2100 SUBRAY(IIDUM)=0.0
C*
C** REPLACE INTERCLASS SURCLASS PAIRS WITH WEIGHT = 1.0
C**
END1 = 0
NK = NOCLS2 - 1
DO 2200 KI=1,NK

```

SET03810  
SET03820  
SET03830  
SET03840  
SET03850  
SET03860  
SET03870  
SET03880  
SET03890  
SET03900  
SET03910  
SET03920  
SET03930  
SET03940  
SET03950  
SET03960  
SET03970  
SET03980  
SET03990  
SET04000  
SET04010  
SET04020  
SET04030  
SET04040  
SET04050  
SET04060  
SET04070  
SET04080  
SET04090  
SET04100  
SET04110  
SET04120  
SET04130  
SET04140  
SET04150  
SET04160  
SET04170  
SET04180  
SET04190  
SET04200  
SET04210  
SET04220  
SET04230  
SET04240  
SET04250  
SET04260  
SET04270  
SET04280  
SET04290  
SET04300  
SET04310  
SET04320  
SET04330  
SET04340  
SET04350  
SET04360  
SET04370  
SET04380  
SET04390  
SET04400  
SET04410  
SET04420  
SET04430  
SET04440  
SET04450  
SET04460  
SET04470  
SET04480  
SET04490  
SET04500  
SET04510  
SET04520  
SET04530  
SET04540  
SET04550  
SET04560

FILE SETUP4

```
JJ1=ARRAY(SURN02-1*KI)
START1 = END1 + 1
END1 = START1 + JJ1 - 1
DO 2300 I=START1,END1
FND2 = END1
D = KI + 1
DO 400 K2 = 0,NOCLS2
JJ2=ARRAY(SURN02-1*K2)
START2 = END2 + 1
FND2 = START2 + JJ2 - 1
DO 500 N = START2,END2
KKDUM=(W1-1+I+(N-1)*NOSUR2)
KKDUM=(W1-1+N+(I-1)*NOSUB2)
SURRAY(KDUM)=1.0
SURRAY(KKDUM)=1.0
500 CONTINUE
400 CONTINUE
2300 CONTINUE
2200 CONTINUE
GO TO 287
END
```

```
SET04570
SET04580
SET04590
SET04600
SET04610
SET04620
SET04630
SET04640
SET04650
SET04660
SET04670
SET04680
SET04690
SET04700
SET04710
SET04720
SET04730
SET04740
SET04750
SET04760
SET04770
```

FILE: TRACE

```
FUNCTION TRACE(A,B,N)
C*
C* DOUBLE PRECISION TRACE
C* FUNCTION ROUTINE TO COMPUTE THE TRACE OF THE PRODUCT OF TWO
C* SYMMETRIC MATRICES, STORED IN SYMMETRIC NOTATION. THE DIMENSIONS
C* OF A AND B ARE  $N*(N+1)/2$ 
C*
DOUBLE PRECISION A,B,SUM,SUM1
DIMENSION A(1),B(1)
K=0
SUM1=0.0
DO 20 I=1,N
M=I-1
SUM=0.0
IF (M.EQ.0) GO TO 15
SUM=0.0
DO 10 J=1,M
K=K+1
10 SUM = SUM + A(K)*B(K)
15 K=K+1
SUM1 = SUM1 + A(K)*B(K) + SUM*2.
20 CONTINUE
TRACE = SUM1
RETURN
END
```

FILE: TRNDIV

```

C* SURROUTINE TRNDIV(SPMSR,COVMTX,AVEMTX,COVMT2,AVEMT2,
C*   WEIGHT,DIVTAB,
C*   WRKRY,IWRKSZ,IPART,PARTLS,BMAT,IFULL)
C*
C* SURROUTINE TO COMPUTE THE AVERAGE WEIGHTED TRANSFORMED
C* DIVERGENCE, AND PARTIALS WITH RESPECT TO B.
C*
C* IF IFULL=1 COMPUTE TRANSFORMED DIVERGENCE FOR ALL 'NOFET' CHANNELS.
C* PARTIALS CANNOT BE COMPUTED WHEN IFULL=1.
C*
C* INCLUDE COMR7.LIST
C* DOUBLE PRECISION SPMSR
C* INCLUDE COMR1.LIST
C* COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
C*   AVAR2,COVAR2,CLSID2,SURNO2,SURDS2,FLDSV2,VERTX2,
C*   FETVC2(30),SURVC2(75),SURPTR(75),CLSV2(60),
C*   KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
C*   GRPCHK(61),GROUPS(124)
C* COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
C*   INCVEC(30),ICOUNT,SETHGT,EVALRF(100),FETVC4(30)
C*   ,NOFET4,VARSZ4,CORPAS,UTAB4,WGHS14,RESTVC(10),DIVSIZ
C*   ,STATKY,ADRES0,ADRES1,ADRESF,ADRSH1,ADRSH2
C*   ,INTEGR ADRES0,ADRES1,ADRESF,ADRSH1,ADRSH2,STATKY
C*   ,DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C*
C* SEND
C* INTEGER VAR SZ4
C* INTEGER VAR SZ2,DIVSIZ
C* DOUBLE PRECISION BMAT,PARTLS
C* DOUBLE PRECISION DIVTAB,DET,DET2,CON,TRACE
C* DOUBLE PRECISION COVMT2,AVEMT2,WRKRY(1),T(30)
C* DIMENSION COVMTX(VARSZ2,NOCLS2), COVMT2(VARSZ4,NOCLS2),
C*   AVEMTX(NOFET2,NOCLS2), AVEMT2(NOFET4,NOCLS2),
C*   WEIGHT(DIVSIZ), DIVTAB(DIVSIZ), PARTLS(1), BMAT(1)
C*
C* IVSZ=VAR SZ4
C* NF=NOFET4
C* IF(IFULL.EQ.1)IVSZ=VAR SZ2
C* IF(IFULL.EQ.1)NF=NOFET2
C* ICOV1=1
C* ICOV2=ICOV1+IVSZ
C* IS2=ICOV2+IVSZ
C* IW1=IS2 + IVSZ
C* IS1=IW1+IVSZ
C* ITFST=IS1
C* IF(IPART.LT.0)GO TO 3
C* ZERO PARTIALS
C* IQ=NOFET2*NOFET4
C* DO 2 IK=1,IQ
C* 2 PARTLS(IK)=0.0
C*   IW2=IS1+VAR SZ2
C*   IW3=IW2+IQ
C*   IW4=IW3+IQ
C*   ITFST=IW4+IQ
C* 3 CONTINUE
C* IF(IWRKSZ/2.GE.ITFST)GO TO 1
C* WRITE(6,600)IWRKSZ
C* CALL CMERR
C* 1 CONTINUE
C* SPMSR=0.0
C* MN=0
C* NC=NOCLS2-1
C* DO 100 I=1,NC
C*   NS=I+1
C*   FIND INVERSE COVAR FOR CLASS I
C*   DO 5 II=1,IVSZ
C*     IF(IFULL.EQ.1)WRKRY(II)=COVMTX(II,I)
C*     IF(IFULL.NE.1)WRKRY(II)=COVMT2(II,I)
C* 5 CONTINUE
C*   CALL COLINV(WRKRY(ICOV1),NF,IERR,3,DET)
C*   IF(IERR.EQ.0)GO TO 6
C*   WRITE(6,500)I
C*   GO TO 100
C* 6 DO 90 J=NS,NOCLS2
C*   DO 7 II=1,NF
C*     IF(IFULL.EQ.1)T(II)=AVEMTX(II,I)-AVEMTX(II,J)
C*     IF(IFULL.NE.1)T(II)=AVEMT2(II,I)-AVEMT2(II,J)
C* 7 CONTINUE
C*   MN=MN+1
C*   K=0

```

FILE: TRNDIV

```
DO 12 II=1,NF
DO 12 IJ=1,II
K=K+1
IF(IFULL.NE.1)GO TO 10
WRKRY(IS2+K-1)=COVMTX(K,I)+COVMTX(K,J)+T(II)*T(IJ)
GO TO 12
10 WRKRY(IS2+K-1)=COVMT2(K,I)+COVMT2(K,J)+T(II)*T(IJ)
12 CONTINUE
C*
C* IF PARTIALS ARE TO BE CALCULATED COMPUTE FULL 'S' MATRIX FOR
C* CLASSES I AND J
IF(IPART.LT.0)GO TO 25
DO 15 II=1,NOFET2
15 T(II)=AVEMTX(II,I)-AVEMTX(II,J)
K=0
DO 20 II=1,NOFET2
DO 20 IJ=1,II
K=K+1
20 WRKRY(IS1+K-1)=COVMTX(K,I)+COVMTX(K,J)+T(II)*T(IJ)
C*
C* FIND INVERSE FOR CLASS J
25 DO 30 II=1,IVSZ
IF(IFULL.EQ.1)WRKRY(ICOV2+II-1)=COVMTX(II,J)
IF(IFULL.NE.1)WRKRY(ICOV2+II-1)=COVMT2(II,J)
30 CONTINUE
CALL COLINV(WRKRY(ICOV2),NF,IERR,3,DET2)
IF(IERR.EQ.0)GO TO 35
WRITE(6,500)J
GO TO 90
C*
C* SUM INVERSES AND COMPUTE TRACE OF SUM * S2
35 DO 40 II=1,IVSZ
40 WRKRY(IW1+II-1)=WRKRY(ICOV1+II-1)+WRKRY(ICOV2+II-1)
DIVTAB(MN)=TRACE(WRKRY(IW1),WRKRY(IS2),NF)/2.-2.*NF
DIVTAB(MN)=DEXP(-DIVTAB(MN)/16.)
SPMSR=SPMSR+DIVTAB(MN)*WEIGHT(MN)
IF(IPART.LT.0)GO TO 90
C*
C* COMPUTE PARTIALS
CALL MT1(RMAT,COVMTX(1,I),WRKRY(IW2),NOFET4,NOFET2)
CALL MT3(WRKRY(ICOV1),WRKRY(IW2),WRKRY(IW3),NOFET4,NOFET4,NOFET2,
1.0)
* CALL MT3(WRKRY(IS2),WRKRY(IW3),WRKRY(IW2),NOFET4,NOFET4,NOFET2,
1.0)
* CALL MT3(RMAT,WRKRY(IS1),WRKRY(IW4),NOFET4,NOFET2,NOFET2,0.1)
DO 42 IK=1,IQ
L=IK-1
42 WRKRY(IW4+L)=WRKRY(IW4+L)-WRKRY(IW2+L)
CALL MT3(WRKRY(ICOV1),WRKRY(IW4),WRKRY(IW2),NOFET4,NOFET4,
NOFET2,1.0)
* CALL MT1(RMAT,COVMTX(1,J),WRKRY(IW3),NOFET4,NOFET2)
CALL MT3(WRKRY(ICOV2),WRKRY(IW3),WRKRY(IW4),NOFET4,NOFET4,
NOFET2,1.0)
* CALL MT3(WRKRY(IS2),WRKRY(IW4),WRKRY(IW3),NOFET4,NOFET4,
NOFET2,1.0)
* CALL MT3(RMAT,WRKRY(IS1),WRKRY(IW4),NOFET4,NOFET2,NOFET2,0.1)
DO 43 IK=1,IQ
L=IK-1
43 WRKRY(IW4+L)=WRKRY(IW4+L)-WRKRY(IW3+L)
CALL MT3(WRKRY(ICOV2),WRKRY(IW4),WRKRY(IW3),NOFET4,NOFET4,
NOFET2,1.0)
DO 44 IK=1,IQ
L=IK-1
44 WRKRY(IW2+L)=WRKRY(IW2+L)+WRKRY(IW3+L)
CON=WEIGHT(MN)*DIVTAB(MN)/(16.*NOCLS2)
DO 50 IK=1,IQ
PARTLS(IK)=PARTLS(IK)-CON*WRKRY(IW2+IK-1)
50 CONTINUE
90 CONTINUE
100 CONTINUE
SPMSR=SPMSR/NOCLS2
RETURN
500 FORMAT(' REDUCED COVARIANCE MATRIX FOR CLASS',I3,' IS NOT POSITIVE')
* DFFI(ITE)
600 FORMAT(' NOT ENOUGH WORK AREA IN TRNDIV -- IWRKSZ=',I5)
END
```

TRN00810  
TRN00820  
TRN00830  
TRN00840  
TRN00850  
TRN00860  
TRN00870  
TRN00880  
TRN00890  
TRN00900  
TRN00910  
TRN00920  
TRN00930  
TRN00940  
TRN00950  
TRN00960  
TRN00970  
TRN00980  
TRN00990  
TRN01000  
TRN01010  
TRN01020  
TRN01030  
TRN01040  
TRN01050  
TRN01060  
TRN01070  
TRN01080  
TRN01090  
TRN01100  
TRN01110  
TRN01120  
TRN01130  
TRN01140  
TRN01150  
TRN01160  
TRN01170  
TRN01180  
TRN01190  
TRN01200  
TRN01210  
TRN01220  
TRN01230  
TRN01240  
TRN01250  
TRN01260  
TRN01270  
TRN01280  
TRN01290  
TRN01300  
TRN01310  
TRN01320  
TRN01330  
TRN01340  
TRN01350  
TRN01360  
TRN01370  
TRN01380  
TRN01390  
TRN01400  
TRN01410  
TRN01420  
TRN01430  
TRN01440  
TRN01450  
TRN01460  
TRN01470  
TRN01480  
TRN01490  
TRN01500  
TRN01510  
TRN01520  
TRN01530  
TRN01540  
TRN01550



FILE: TRNSFR

```

C      SUBROUTINE TRNSFR(A,A2,W,BMAT)
C      INCLUDE COMRK7.LIST
C      INCLUDE COMRK1.LIST
COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VAR5Z2,TOTVT2,NOFLD2,
*      AVAR2,COVAR2,CLSID2,SURNO2,SUR0S2,FLDSV2,VERTX2,
*      FFTVC2(30),SUHVC2(75),SURPTR(75),CLSV2(60),
*      KEPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
*      GRPCHK(61),GROUPS(124)
COMMON/FSL/CFAC,TOTMSR,SEPMSR,PKCKEY,CRKEY,INCFET,
*      INCVEC(30),ICOUNT,SETWGT,EVALHF(100),FETVC4(30)
*      ,NOFET4,VAR5Z4,CORHAS,DTAH4,WGH514,RESTVC(10),DIVSIZ
*      ,STATKY,ADRESO,ADRESF,ADRESF,ADRS1,ADRS2
INTEGER ADRESO,ADRESF,ADRESF,ADRS1,ADRS2,STATKY
DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
CSEND  INTEGER VAR5Z4,VAR5Z2
        DOUBLE PRECISION SUM
        DOUBLE PRECISION RMAT(NOFET4,NOFET2)
        DOUBLE PRECISION A2(VAR5Z4,NOCLS2),W (NOFET4,NOFET2)
        DIMENSION A(VAR5Z2,NOCLS2)
C*
C*      MULTIPLY BMAT * A * BMAT(TRANPOSE) AND STORE IN A2
C*
        DO 140 JJ=1,NOCLS2
        DO 150 I=1,NOFET4
        DO 150 J=1,NOFET2
        SUM=0.0
        DO 140 K=1,NOFET2
        IF(K.GE.J)IP=K*(K-1)/2 + J
        IF(K.LT.J)IP=J*(J-1)/2 + K
        140 SUM=SUM + BMAT(I,K)* A(IP,JJ)
        W(I,J)=SUM
        150 CONTINUE
C*
        DO 160 I=1,NOFET4
        DO 170 J=1,NOFET4
        SUM=0.0
        DO 160 K=1,NOFET2
        160 SUM=SUM + W(J,K) * BMAT(I,K)
        IF(J.GE.I)IP=J*(J-1)/2 + I
        IF(J.LT.I)IP=I*(I-1)/2 + J
        A2(IP,JJ)=SUM
        170 CONTINUE
        180 CONTINUE
        190 CONTINUE
        RETURN
        END
TRN00010
TRN00020
TRN00030
TRN00040
TRN00050
TRN00060
TRN00070
TRN00080
COM00010
COM00020
COM00030
COM00040
COM00050
COM00060
TRN00160
TRN00170
TRN00180
TRN00190
TRN00200
TRN00210
TRN00220
TRN00230
TRN00240
TRN00250
TRN00260
TRN00270
TRN00280
TRN00290
TRN00300
TRN00310
TRN00320
TRN00330
TRN00340
TRN00350
TRN00360
TRN00370
TRN00380
TRN00390
TRN00400
TRN00410
TRN00420
TRN00430
TRN00440
TRN00450
TRN00460
TRN00470
TRN00480
```

FILE: USERIN

```

      SURROUTINE USERIN(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,AVEMT2,S,S2,  USE00010
      BMAT,WRKRY,IWRKSZ)  USE00020
C**  USE00030
C**  SURROUTINE USERIN COORDINATES THE NECESSARY ROUTINES TO COMPUTE  USE00040
C**  THE REQUESTED SEPARABILITY MEASURE FOR THE INPUT B-MATRIX.  USE00050
C**  USE00060
C**  INCLUDE COMMK1.LIST  USE00070
C**  INCLUDE COMMK7.LIST  USE00080
C**  COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARZ2,TOTVT2,NOFLD2,  COM00010
      AVAR2,COVAR2,CLSID2,SURN02,SURDS2,FLDSV2,VERTX2,  COM00020
      FETVC2(30),SUHVC2(75),SUHPTR(75),CLSV2(60),  COM00030
      KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),  COM00040
      GRPCHK(61),GROUPS(124)  COM00050
C**  COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,  COM00010
      INCVEC(30),ICOUNT,SETWGT,EVALBF(100),FETVC4(30)  COM00020
C**  .NOFET4,VARZ4,COWAS,DIAR4,WGHS14,RESTVC(10),DIVSZ  COM00030
C**  .STATKY,ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2  COM00040
C**  INTEGER ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY  COM00050
C**  DOUBLE PRECISION CFAC,TOTMSR,SEPMSR  COM00060
C$END  USE00100
      INTEGER CRIKEY,VARZ2,VARZ4  USE00110
      DIMENSION COVMTX(VARZ2,NOCLS2), AVEMTX(NOFET2,NOCLS2),  USE00120
      S(VARZ2,NOCLS2),  USE00130
      WEIGHT(1),WRKRY(1)  USE00140
      DOUBLE PRECISION COVMT2(VARZ4,1),AVEMT2(NOFET4,1),  USE00150
      S2(VARZ4,1),BMAT(1)  USE00160
      DOUBLE PRECISION DIVTAB(1)  USE00170
      DIMENSION DUM(1)  USE00180
C**  USE00190
C**  GET B-MATRIX FROM FILE IN SINGLE PRECISION THEN STORE IN D.P.  USE00200
C**  USE00210
      CALL BMFIL(WRKRY,NOFET4,NOFET2,FETVC2,2)  USE00220
      IK=NOFET4*NOFET2  USE00230
      DO 10 I=1,IK  USE00240
10  RMAT(I)=WRKRY(I)  USE00250
C**  USE00260
C**  GET TRANSFORMED STATISTICS  USE00270
C**  USE00280
      CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DUM,BMAT,WRKRY,  USE00290
      IWRKSZ)  USE00300
C**  EVALUATE SEPARABILITY MEASURE  USE00310
      IPART=-1  USE00320
C**  CALL EVALSP(SEPMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,  USE00330
      WEIGHT,IPART,DUM,BMAT,WRKRY,IWRKSZ)  USE00340
      IF(CRIKEY.NE.1)RETURN  USE00350
C**  USE00360
C**  EVALUATE INTERCLASS DIVERGENCES  USE00370
C**  USE00380
      CALL DIVRG1(COVMT2,VARZ4,AVEMT2,DIVTAB,NOCLS2,NOFET4,  USE00390
      WRKRY,IWRKSZ)  USE00400
      RETURN  USE00410
      END  USE00420
```

FILE: WGTCHK

```
SUBROUTINE WGTCHK(WEIGHT,CLSNAM,NAMPR,WGHT,WPTR,WRKRY,NOCLS2)      WGT00010
IMPLICIT INTEGER(A-Z)                                             WGT00020
REAL WEIGHT(1),WGHT(1),WRKRY(NOCLS2,NOCLS2),WHT                  WGT00030
DIMENSION CLSNAM(NOCLS2),NAMPR(2,WPTR)                           WGT00040
REAL RWGR,RKEY                                                    WGT00050
LOGICAL LTEMP(4),LSTOR(4),NSTOR(4)                               WGT00060
EQUIVALENCE (RWGR,SETWGT),(RKEY,WTKEY),(LSTORE,LSTOR(1)),(NSTORE, WGT00070
,NSTOR(1)),(ITEMP,LTEMP(1))                                       WGT00080
DATA BLANK/' ',OTHERS/'OTHER',BLANKS/' '                          WGT00090
C* PASS KEYS IN ALREADY EXISTING STORAGE                           WGT00100
RWGR = WEIGHT(1)                                                  WGT00110
RKEY = WEIGHT(2)                                                  WGT00120
C* DELETE BLANKS FOR TESTING                                       WGT00130
DO 20 I=1,NOCLS2                                                 WGT00140
LSTORE = CLSNAM(I)                                               WGT00150
NSTORE = BLANKS                                                  WGT00160
IZ = 1                                                            WGT00170
DO 10 IY=1,4                                                     WGT00180
ITEMP=BLANKS                                                    WGT00190
ITEMP(IY)=LSTORE(IY)                                           WGT00200
IF((ITEMP.EQ.BLANK).AND.(IZ.EQ.1))GO TO 10                       WGT00210
NSTOR(IZ) = LSTORE(IY)                                          WGT00220
IZ = IZ + 1                                                     WGT00230
10 CONTINUE                                                       WGT00240
20 CLSNAM(I) = NSTORE                                           WGT00250
IF(WTKEY.EQ.1) GO TO 27                                          WGT00260
C* SET ALL CLASS PAIR WEIGHTS TO 1.0 IF USER HAS NOT INPUT WEIGHT FOR WGT00270
C* 'OTHERS' OR 'CLSNAM'. SET TO 'OTHERS' VALUE IF INPUT.        WGT00280
WHT=1.0                                                           WGT00290
DO 22 I=1,WPTR                                                  WGT00300
IF(NAMPR(I,1) .NE. OTHERS)GO TO 22                               WGT00310
WHT=WGHT(I)                                                       WGT00320
GO TO 24                                                          WGT00330
22 CONTINUE                                                       WGT00340
24 DO 25 IK=1,NOCLS2                                           WGT00350
DO 25 JK=1,NOCLS2                                           WGT00360
WRKRY(IK,JK)=WHT                                               WGT00370
25 WRKRY(JK,IK)=WHT                                             WGT00380
C* PLACEMENT OF INPUTED WEIGHT VALUES                             WGT00400
C*                                                                 WGT00410
C*                                                                 WGT00420
C*                                                                 WGT00430
27 IF(SFTWGT.NE.2) GO TO 55                                       WGT00440
DO 50 I = 1,WPTR                                               WGT00450
DO 40 J=1,NOCLS2                                               WGT00460
IF (CLSNAM(J) .NE. NAMPR(1,I))GO TO 40                          WGT00470
C* FOUND MATCH ON FIRST NAME IN NAMPR - INDEX J                 WGT00480
C* NOW SEE ABOUT SECOND NAME                                     WGT00490
C*                                                                 WGT00500
IF(NAMPR(2,I).EQ. BLANK) GO TO 35                               WGT00510
DO 30 K=1,NOCLS2                                               WGT00520
IF (CLSNAM(K) .NE. NAMPR(2,I))GO TO 30                          WGT00530
C* FOUND SECOND MATCH - INDEX K                                  WGT00540
C*                                                                 WGT00550
C*                                                                 WGT00560
WRKRY(J,K)=WGHT(I)                                             WGT00570
WRKRY(K,J)=WGHT(I)                                             WGT00580
GO TO 50                                                         WGT00590
30 CONTINUE                                                       WGT00600
WRITE(6,100) NAMPR(2,I)                                         WGT00610
GO TO 50                                                         WGT00620
C* ALL PAIRS FOR CLASS J SET TO SAME WEIGHT                       WGT00630
C*                                                                 WGT00640
C*                                                                 WGT00650
35 DO 36 IK=1,NOCLS2                                           WGT00660
WRKRY(IK,J)=WGHT(I)                                           WGT00670
36 WRKRY(J,IK)=WGHT(I)                                         WGT00680
GO TO 50                                                         WGT00690
40 CONTINUE                                                       WGT00700
IF(NAMPR(1,I) .EQ. OTHERS)GO TO 50                              WGT00710
46 WRITE(6,100) NAMPR(1,I)                                       WGT00720
50 CONTINUE                                                       WGT00730
55 CONTINUE                                                       WGT00740
C* REDUCE WRKRY MATRIX AND STORE IN WEIGHTS                       WGT00750
C*                                                                 WGT00760
C*                                                                 WGT00770
K=0                                                                WGT00780
                                                                WGT00790
```

FILE: WGTCHK

```
NC=NOCLS2-1
DO 60 I=1.NC
IK=I+1
DO 60 J=IK.NOCLS2
K=K+1
WEIGHT(K)=WPKRY(I,J)
60 CONTINUE
RETURN
100 FORMAT(1 SURCLASS 'A6' IS NOT AMONG INPUT SUBCLASSES - WEIGHT IN
*PUT IGNORED*)
END
```

WGT00800  
WGT00810  
WGT00820  
WGT00830  
WGT00840  
WGT00850  
WGT00860  
WGT00870  
WGT00880  
WGT00890  
WGT00900

FILE: WGTSCN

```
SUBROUTINE WGTSCN(CARD,COL,NAMPR,WGHT,WSIZ,NCNT)
IMPLICIT INTEGER(A-Z)
-----
CALL... J=WGTSCN(CARD,COL,NAMPR,WGHT,WSIZ)
-----
ARGS... CARD - ARRAY OF CHARACTERS TO BE SCANNED.
          COL - COLUMN IN CARD TO BEGIN SCAN, ON OUTPUT
          NAMPR- ARRAY CONTAINING, ON OUTPUT, THE PAIRS OF CLASS
          WGHT - ARRAY CONTAINING WEIGHT FOR CORRESPONDING
          WSIZ - SIZE OF WGHT BUFFER
          NCNT - RUNNING COUNT OF NAME PAIRS SCANNED
PURPOSE. SCANS THE WEIGHTS CONTROL CARD, SAVING THE CLASS NAME
          PAIRS AND ASSOCIATED WEIGHT FOR LATER VERIFICATION IN
          SUBROUTINE WGTCHK. THE WEIGHT CARD MAY TAKE THE FOLLOWIN
          FORMS:
          WEIGHT CLASS 1=10.5, CLASS 2=12.0, OTHERS=20.0
          OR:
          WEIGHTS (CLASS 1, CLASS 2)=15.0, CLASS 3=1.0, OTHERS=5
-----
DIMENSION COMVEC(2),EQUVEC(2),RPNVEC(2)
DATA BLANK/' ',LFTPRN/'(',COMMA/',',EQUAL/'=',RHTPRN/')'
DATA BLANKS/' '/
DATA COMVEC/1,' ',EQUVEC/1,'=',RPNVEC/1,')'
REAL WGHT
DIMENSION CARD(1),NAMPR(2,1),WGHT(1)
LOGICAL*1 LSTOR(4),LCARD(4)
EQUIVALENCE (MCARD,LCARD(1)),(STOR,LSTOR(1))
-----
C*
1 J=NXTCHR(CARD,COL)
IF(J.EQ.BLANK)GO TO 60
IF(J.NE.LFTPRN)GO TO 2
COL=COL+1
C*
2 CONTINUE
BLANK OUT NAMPR FOR THIS PAIR
NCNT=NCNT+1
DO 3 I=1,2
3 NAMPR(I,NCNT)=BLANKS
IR=WSIZ-NCNT
IF(IR.GT.0)GO TO 4
WRITE(6,200)WSIZ
GO TO 60
4 CONTINUE
STOR=BLANKS
WCNT = 0
6 WCNT = WCNT + 1
IF(CARD(COL).EQ.BLANK)GO TO 10
IF(CARD(COL).EQ.EQUAL)GO TO 45
IF(CARD(COL).EQ.COMMA)GO TO 20
MCARD = CARD(COL)
LSTOR(WCNT)=LCARD(1)
NAMPR(1,NCNT)=STOR
10 COL=COL+1
IF(WCNT.EQ.4)GO TO 15
GO TO 6
C*
C*
C*
ONLY FOUR CHARACTERS PER NAME ALLOWED - IGNORE REMAINDER
FIND = 0H , - ERROR OTHERWISE
C*
15 CONTINUE
J=IND12(CARD,COL,COMVEC)
IF(J.EQ.-1)GO TO 40
GO TO 19
C*
C*
COMMA FOUND - ANOTHER NAME SHOULD FOLLOW
20 WCNT = 0
21 COL=COL+1
WCNT = WCNT + 1
STOR = BLANKS
IF(CARD(COL).EQ.RHTPRN)GO TO 40
IF(CARD(COL).EQ.BLANK)GO TO 25
-----
WGT00010
WGT00020
WGT00030
WGT00040
WGT00050
WGT00060
WGT00070
WGT00080
WGT00090
WGT00100
WGT00110
WGT00120
WGT00130
WGT00140
WGT00150
WGT00160
WGT00170
WGT00180
WGT00190
WGT00200
WGT00210
WGT00220
WGT00230
WGT00240
WGT00250
WGT00260
WGT00270
WGT00280
WGT00290
WGT00300
WGT00310
WGT00320
WGT00330
WGT00340
WGT00350
WGT00360
WGT00370
WGT00380
WGT00390
WGT00400
WGT00410
WGT00420
WGT00430
WGT00440
WGT00450
WGT00460
WGT00470
WGT00480
WGT00490
WGT00500
WGT00510
WGT00520
WGT00530
WGT00540
WGT00550
WGT00560
WGT00570
WGT00580
WGT00590
WGT00600
WGT00610
WGT00620
WGT00630
WGT00640
WGT00650
WGT00660
WGT00670
WGT00680
WGT00690
WGT00700
WGT00710
WGT00720
WGT00730
WGT00740
WGT00750
WGT00760
WGT00770
WGT00780
WGT00790
```

FILE: WGTSCN

```

M CARD = CARD(COL)
LSTOR(WCNT) = LCARD(1)
NAMPR(2,NCNT)=STOR
25 COL=COL+1
   IF(WCNT.EQ.4)GO TO 30
   GO TO 21
30 CONTINUE
   J=FINDI2(CARD,COL,RPNVEC)
   IF(J.NE.-1)GO TO 40
   WRITE(6,100)
   NCNT=NCNT-1
   GO TO 60
C*
C*
C*
40 J=FINDI2(CARD,COL,EUVEC)
   IF(J.NE.-1)GO TO 45
   WRITE(6,100)
   NCNT=NCNT-1
   GO TO 40
45 J=FLTHIM(CARD,COL,WGHT(NCNT),IR)
   COL=COL-1
   J=NXTCHM(CARD,COL)
   IF(J.EQ.COMMA)GO TO 1
   IF(J.EQ.BLANK)GO TO 60
   IF(J.NE.LFIPRN)GO TO 2
   COL=COL+1
   GO TO 7
50 CONTINUE
   RETURN
100 FORMAT(' SYNTAX ERROR ON WEIGHT CARD-REMAINDER OF CARD IGNORED')
200 FORMAT(' WEIGHT BUFFER IS FILLED-ONLY',IS,' CLASS NAME PAIRS ALLOW
      *D')
END
```

WGT00800  
WGT00810  
WGT00820  
WGT00830  
WGT00840  
WGT00850  
WGT00860  
WGT00870  
WGT00880  
WGT00890  
WGT00900  
WGT00910  
WGT00920  
WGT00930  
WGT00940  
WGT00950  
WGT00960  
WGT00970  
WGT00980  
WGT00990  
WGT01000  
WGT01010  
WGT01020  
WGT01030  
WGT01040  
WGT01050  
WGT01060  
WGT01070  
WGT01080  
WGT01090  
WGT01100  
WGT01110  
WGT01120  
WGT01130

FILE: WMRPLC

```
SUBROUTINE WMRPLC(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,      WMR00010
    AVEMT2,S,S2,WRKRY,IWRKSZ)                               WMR00020
C*                                                           WMR00030
C* SUBROUTINE TO FIND THE BEST SET OF NOFET4 FEATURES     WMR00040
C* USING THE WITHOUT REPLACEMENT PROCEDURE.              WMR00050
C*                                                           WMR00060
C* INTEGER FETVC2,FFVC4,TVEC,TRYVEC,KEEP                 WMR00070
C* INCLUDE COMARK7.LIST                                   WMR00080
C* INCLUDE COMARK1.LIST                                   WMR00090
C* COMMON/INFORM/NOCLS2,NOSUP2,NOFET2,VARSZ2,TOTVT2,NOFLD2, WMR00100
C*     AVAR2,COVAR2,CLS1D2,SURN02,SUBDS2,FLOS2,VERTX2,    WMR00110
C*     FETVC2(30),SURVC2(75),SURPTR(75),CLSV2(60),        WMR00120
C*     KEMPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),           WMR00130
C*     GRPCHK(61),GROUPS(124)                             WMR00140
C* COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,    WMR00150
C*     INCVEC(30),ICOUNT,SETWGT,EVALHF(100),FFVC4(30)     WMR00160
C*     NOFET4,VARSZ4,COHAS,DTAR4,WGMS14,HESTVC(10),DIVSZ  WMR00170
C*     ,STATKY,ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2        WMR00180
C*     INTEGER ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY WMR00190
C*     DOUBLE PRECISION CFAC,TOTMSR,SEPMSR                WMR00200
CSEND                                                       WMR00210
    DOUBLE PRECISION COVMT2(1),AVEMT2(1),S2(1)           WMR00220
    DOUBLE PRECISION DIVTAB(1),TMSR,DUM(1),DM            WMR00230
    INTEGER CRIKEY                                         WMR00240
    DIMENSION COVMTX(1),AVEMTX(1),WEIGHT(1),             WMR00250
    S(1),WRKRY(1)                                          WMR00260
    IPART=-1                                               WMR00270
    DIMENSION TVEC(30),NREST(30),TRYVEC(30)              WMR00280
C*                                                           WMR00290
C* SAVE THE VALUE OF NOFET4                               WMR00300
C* NFSAVE=NOFET4                                          WMR00310
    IF(NRST.GT.0)GO TO 15                                  WMR00320
    IF(INCFET.LE.0)GO TO 15                                WMR00330
    DO 10 I=1,INCFET                                       WMR00340
    DO 5 J=1,NOFET2                                         WMR00350
    IF(INCVEC(I).EQ.FFVC2(J))GO TO 6                       WMR00360
5  CONTINUE                                               WMR00370
    WRITE(6,100)INCVEC(I)                                  WMR00380
    GO TO 10                                               WMR00390
6  NRST=NRST+1                                            WMR00400
    NREST(NRST)=J                                         WMR00410
10  CONTINUE                                              WMR00420
C*                                                           WMR00430
C* SET UP VECTOR OF FEATURES TO TRY WITH NREST           WMR00440
C*                                                           WMR00450
15  IF(NRST.GE.NFSAVE)GO TO 50                             WMR00460
    SEPMSR=1.E+35                                          WMR00470
    NTRY=0                                                 WMR00480
    DO 25 I=1,NOFET2                                       WMR00490
    IF(NRST.EQ.0)GO TO 24                                  WMR00500
    DO 20 J=1,NRST                                         WMR00510
    IF(I.EQ.NREST(J))GO TO 25                             WMR00520
20  CONTINUE                                              WMR00530
24  NTRY=NTRY+1                                           WMR00540
    TRYVEC(NTRY)=I                                         WMR00550
25  CONTINUE                                              WMR00560
C*                                                           WMR00570
C* TRY EACH FEATURE IN TRYVEC WITH THE 'NREST' SO FAR   WMR00580
C* AND KEEP THE ONE WHICH GIVES MAXIMUM SEPARABILITY     WMR00590
C* MEASURE.                                               WMR00600
C*                                                           WMR00610
    NF=NRST+1                                             WMR00620
    NOFET4=NF                                             WMR00630
    DO 40 I=1,NTRY                                         WMR00640
    IF(NRST.EQ.0)GO TO 35                                  WMR00650
    DO 30 J=1,NRST                                         WMR00660
30  TVEC(J)=NREST(J)                                       WMR00670
35  TVEC(NF)=TRYVEC(I)                                       WMR00680
    CALL ORDER(TVEC,NF)                                     WMR00690
    CALL GSTATIC(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,TVEC,DM,WRKRY,IWRKSZ) WMR00700
    CALL EVALSP(TMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB, WMR00710
    WEIGHT,IPART,DUM,DUM,WRKRY,IWRKSZ)                   WMR00720
    IF(SEPMSR.LT.TMSR)GO TO 40                             WMR00730
    KEEP=TRYVEC(I)                                         WMR00740
    SEPMSR=TMSR                                            WMR00750
40  CONTINUE                                              WMR00760
    NRST=NRST+1                                           WMR00770
    NREST(NRST)=KEEP                                       WMR00780
    GO TO 15                                               WMR00790
50  NOFET4=NFSAVE                                         WMR00800
```

FILE: WHRPLC

```
DO 60 I=1,NOFET4
K=NRST(I)
TVFC(I)=K
60 FETVC4(I)=FETVC2(K)
C*
C* COMPUTE INTERCLASS MEASURES FOR FEATURES CHOSEN.
C*
CALL ORDER(TVEC,NOFET4)
CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,TVEC,DUM,WRKRY,
IWRKSZ)
* CALL FVALSP(SFPMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,
WEIGHT,IPART,DUM,DUM,WRKRY,IWRKSZ)
* IF(CRIKEY.NF.1)RETURN
CALL DIVRG1(COVMT2,VAR24,AVEMT2,DIVTAB,NOCLS2,
NOFET4,WRKRY,IWRKSZ)
* RETURN
100 FORMAT(' THE INCLUDE REQUEST FOR FEATURE',I4,
' IS NOT A LEGITIMATE REQUEST--IGNORED')
END
```

WHR00810  
WHR00820  
WHR00830  
WHR00840  
WHR00850  
WHR00860  
WHR00870  
WHR00880  
WHR00890  
WHR00900  
WHR00910  
WHR00920  
WHR00930  
WHR00940  
WHR00950  
WHR00960  
WHR00970  
WHR00980  
WHR00990

ORIGINAL PAGE IS  
OF POOR QUALITY



# 11. CLASSIFY PROCESSOR

FILE: CLSFY

```

SURROUTINE CLSFY (ARRAY, TOP)                                CLS00010
IMPLICIT INTEGER (A-H,O-Z)                                  CLS00020
DIMENSION ARRAY (3000)                                       CLS00030
-----                                                     CLS00040
-----                                                     CLS00050
CALL..    CALL CLSFY (ARRAY, TOP)                            CLS00060
-----                                                     CLS00070
ARGS..    ARRAY = SFE 'MONITOR'                              CLS00080
          TOP   = SFE 'MONITOR'                              CLS00090
-----                                                     CLS00100
REQUIRES. COMMONS /INFORM/CLASS/GLOBAL/BMTRX/SCRACH/        CLS00110
          ROUTINES SETUP2 CLSFY1 CLSFY2                     CLS00120
-----                                                     CLS00130
PURPOSE.. COORDINATES THE VARIOUS ROUTINES                  CLS00140
          FOR 'CLASSIFICATION' STEP                          CLS00150
-----                                                     CLS00160
RETURNS.. NONE                                              CLS00170
-----                                                     CLS00180
-----                                                     CLS00190
-----                                                     CLS00200
-----                                                     CLS00210
-----                                                     CLS00220
-----                                                     CLS00230
-----                                                     CLS00240
-----                                                     CLS00250
-----                                                     CLS00260
-----                                                     CLS00270
-----                                                     CLS00280
-----                                                     CLS00290
-----                                                     CLS00300
-----                                                     CLS00310
-----                                                     CLS00320
-----                                                     CLS00330
-----                                                     CLS00340
-----                                                     CLS00350
-----                                                     CLS00360
-----                                                     CLS00370
-----                                                     CLS00380
-----                                                     CLS00390
-----                                                     CLS00400
-----                                                     CLS00410
-----                                                     CLS00420
-----                                                     CLS00430
-----                                                     CLS00440
-----                                                     CLS00450
-----                                                     CLS00460
-----                                                     CLS00470
-----                                                     CLS00480
-----                                                     CLS00490
-----                                                     CLS00500
-----                                                     CLS00510
-----                                                     CLS00520
-----                                                     CLS00530
-----                                                     CLS00540
-----                                                     CLS00550
-----                                                     CLS00560
-----                                                     CLS00570
-----                                                     CLS00580
-----                                                     CLS00590
-----                                                     CLS00600
-----                                                     CLS00610
-----                                                     CLS00620
-----                                                     CLS00630
-----                                                     CLS00640
-----                                                     CLS00650
-----                                                     CLS00660
-----                                                     CLS00670
-----                                                     CLS00680
-----                                                     CLS00690
-----                                                     CLS00700
-----                                                     CLS00710
-----                                                     CLS00720
-----                                                     CLS00730
-----                                                     CLS00740
-----                                                     CLS00750
-----                                                     CLS00760

```







FILE CATGRY

C  
C  
C  
C  
C

FIND THE MAX. CATEGORY PDF (THIS PDF IS THE SUMMATION OF ALL  
THE PDF'S OF THE SUBCLASS WITHIN  
THE CATEGORY)

PK = 0.0  
DO 175 LL=1,NOCAT  
IF ( P(ILL) .LE. PK) GO TO 175  
PK = P(ILL)  
IC = LL  
175 CONTINUE

C  
C  
C  
C  
C

ALL OF THE SURCLASS PDF'S WERE TOO SMALL TO EXP., THEREFORE  
THIS PIXEL WILL NOT BE CLASSIFIED.

IF (PK .NE. 0.0) GO TO 180  
IR(II) = NOSUB2 + 1  
VR(II) = 0.0  
PTSTHS = PTSTHS + 1  
GO TO 250  
180 CONTINUE

C  
C  
C  
C  
C

STORE THE LARGEST SUBCLASS PDF AND SUBCLASS NO. OF THE CATEGORY  
WITH THE MAX. PDF

IR(II) = SUBNUM(IC)  
VR(II) = TFMAX(IC)  
250 CONTINUE  
350 CONTINUE  
RETURN  
END

CAT01520  
CAT01530  
CAT01540  
CAT01550  
CAT01560  
CAT01570  
CAT01580  
CAT01590  
CAT01600  
CAT01610  
CAT01620  
CAT01630  
CAT01640  
CAT01650  
CAT01660  
CAT01670  
CAT01680  
CAT01690  
CAT01700  
CAT01710  
CAT01720  
CAT01730  
CAT01740  
CAT01750  
CAT01760  
CAT01770  
CAT01780  
CAT01790  
CAT01800  
CAT01810  
CAT01820  
CAT01830

FILE: CATSCN

```
C      FUNCTION CATSCN(CARD,KCLSNA,CATNME,KK,NOCLSS,NOCAT)          CAT00010
C      IMPLICIT INTEGER (A-Z)                                       CAT00020
C      DIMENSION KCLSNA(1),CARD(62),IBUFF(6)                       CAT00030
C-----CAT00040
C      CATSCN SCANS THE CATEGORY CARD FROM CLASSIFY AND STORES THE  CAT00050
C      CATEGORY NAME IN CATNAM AND STORE THE CLASS NAMES IN KCLSNA  CAT00060
C-----CAT00070
C      DATA BLANK/' ',SLASH/'/',STAR/'*',COMMA/',',/             CAT00080
C      LOGICAL *I LCHAR(4),LLCHAR(4)                               CAT00090
C      DIMENSION ICHAR(1),IICHAR(1)                               CAT00100
C      EQUIVALENCE(LCHAR(1),ICHAR(1)),(LLCHAR(1),IICHAR(1))     CAT00110
C      K = 1                                                        CAT00120
C      6 COL = 0                                                    CAT00130
C      KK = KK + 1                                                 CAT00140
C      10 J = NIXCHR(CARD,COL)                                       CAT00150
C      IF(J.EQ.BLANK)GO TO 110                                       CAT00160
C      IF(J.EQ.COMMA)GO TO 10                                       CAT00170
C      IF(J.EQ.SLASH)GO TO 10                                       CAT00180
C      IF(J.EQ.STAR)GO TO 100                                       CAT00190
C      PICK CHARACTERS OFF CARD ONE AT A TIME                       CAT00200
C      LL = 1                                                        CAT00210
C      DO 20 I=1,6                                                  CAT00220
C      J2 = CARD(COL)                                               CAT00230
C      IF(J2.EQ.SLASH)GO TO 30                                       CAT00240
C      IF(J2.EQ.STAR)GO TO 30                                       CAT00250
C      IF(J2.EQ.COMMA)GO TO 30                                       CAT00260
C      IF(J2.EQ.BLANK)GO TO 20                                       CAT00270
C      IBUFF(LL) = J2                                               CAT00280
C      LL = LL + 1                                                  CAT00290
C      COL = COL + 1                                               CAT00300
C      20 CONTINUE                                                 CAT00310
C      30 COL = COL - 1                                             CAT00320
C      IF (LL.NE. 7) GO TO 35                                       CAT00330
C      GO TO 50                                                      CAT00340
C      35 DO 40 JJ=LL,6                                             CAT00350
C      40 IRIFF(JJ) = BLANK                                         CAT00360
C      CONTINUE                                                     CAT00370
C      50 I=1,4                                                     CAT00380
C      IICHAR(I)=IBUFF(I)                                           CAT00390
C      LCHAR(I)=LLCHAR(I)                                           CAT00400
C      WRD1=ICHAR(I)                                                CAT00410
C      GO TO (70,80),K                                              CAT00420
C      WRD1 CONTAINS CATEGORY NAME                                  CAT00430
C      70 CATNME = WRD1                                             CAT00440
C      K = 2                                                         CAT00450
C      GO TO 10                                                      CAT00460
C      WRD1 CONTAINS CLASS NAME                                    CAT00470
C      80 KCLSNA(KK) = WRD1                                         CAT00480
C      NOCLSS = NOCLSS + 1                                         CAT00490
C      GO TO 8                                                       CAT00500
C      NEXT CARD IS A CONTINUATION CARD                            CAT00510
C      100 READ(21,500)CARD                                         CAT00520
C      COL = 0                                                       CAT00530
C      WRITE(6,550)CARD                                             CAT00540
C      500 FORMAT(10X,62A1)                                         CAT00550
C      550 FORMAT(115, 62A1)                                       CAT00560
C      GO TO 10                                                      CAT00570
C      FINISHED SCANNING CARD                                       CAT00580
C      110 KK = KK - 1                                             CAT00590
C      CATSCN = KK                                                 CAT00600
C      RETURN                                                       CAT00610
C      FND                                                           CAT00620
C      CAT00630
C      CAT00640
C      CAT00650
C      CAT00660
C      CAT00670
C      CAT00680
C      CAT00690
C      CAT00700
C      CAT00710
C      CAT00720
C      CAT00730
C      CAT00740
C      CAT00750
C      CAT00760
C      CAT00770
C      CAT00780
C      CAT00790
```

11-6

142

FILE: CLSFY1

```
SUBROUTINE CLSFY1(COVMTX,AVEMTX,FLDMTX,CLSMTX,APRIOR,
* BMATR,VERTEX,SUBDES,SUBNO,COVNEW,AVENEW,KATNO)
IMPLICIT INTEGER (A-H,O-Z)
-----
CALL... CALL CLSFY1(COVMTX,AVEMTX,FLDMTX,RUNMTX,CLSMTX,APRIOR,
          BMATR,COVNEW,AVENEW )
ARGS... COVMTX : LOCATION OF COVARIANCE MATRICES ( SYMETTRIC
          STORAGE ) FOR NOCLS2 TRAINING CLASSES.
          AVEMTX : LOCATION OF NOCLS2 TRAINING CLASS MEAN VECTORS
          ( NOFET2 MEANS PER CLASS )
          FLDMTX : LOCATION OF TRAINING FIELD(S) INFORMATION
          CLSMTX : LOCATION OF NAME FOR EACH CLASS
          APRIOR : LOCATION OF APRIORI PROBABILITY VALUES FOR
          EACH CLASS
VERTEX : LOCATION OF VERTICES OF SAVED TRAINING FIELDS
SUBDES : LOCATION OF SUBCLASS NAMES
SUBNO : LOCATION OF ARRAY CONTAINING NO. OF SUBCLASSES IN
        EACH CLASS
        COVNEW : LOCATION USED TO STORE 'B'-TRANSFORMED
        COVARIANCE MATRICES.
CONTINUE
          AVENEW : LOCATION USED TO STORE THE 'B'-TRANSFORMED
          MEAN VECTORS.
          BMATR : LOCATION OF THE 'B'-TRANSFORMATION MATRIX,
          IF AVATLABLE, FOR APPLICATION TO THE CLASS
          MEANS AND COVARIANCE MATRICES.
KATNO : CATEGORY - CLASS CORRESPONDENCE
PURPOSE...
          IF AVAILABLE, THE 'B'-TRANSFORMATION MATRIX IS APPLIED
          TO THE SUBCLASS MEAN VECTORS AND COVARIANCE MATRICES,
          OBTAINS THE (MODIFIED) CHOLESKY FACTORIZATION OF THE
          SUBCLASS COVARIANCE MATRICES, PROVIDES THE 'CONSTANT' OF
          THE PROBABILITY DENSITY FUNCTION AND DETERMINANT FOR
          EACH SUBCLASS, AND OBTAINS THE SUBCLASS-PAIR THRESHOLDS FOR
          USE BY SUBR. CONTEX IN CLASSIFICATION OF INPUT SCAN
          LINES. PUBLISHES AND OUTPUTS ON MAPTAP THE TRAINING
          FIELD(S) INFORMATION AND THE STATISTICS FOR EACH OF THE
          TRAINING CLASSES.
          RETURNS...CHOLESKY FACTORIZATION OF THE INPUT COVARIANCE MATRICES
          (AFTER 'B'-TRANSFORMATION, IF APPLICABLE), SUBCLASSPAIR
          THRESHOLDS, AND SUBCLASS STATISTICS OUTPUT ON MAPTAP.
          CONTINUE
-----
INCLUDE COMPK1.LIST
-----
INCLUDE COMPK2.LIST
* COMMON/INFORM/NOCL52, NOSUH2, NOFET2, VARS22, TOTVT2, NOFLD2,
  AVAR2, COVAR2, CLSID2, SURNO2, SURDS2, FLDSV2, VERTX2,
  * FFTVC2(30), SUHVC2(75), SUBPTR(75), CLSVC2(60),
```

FILE: CLSFY1

```

*          KEPPTS(60),NOGRP,GHPNAM(60),GHPDEX(61),          CLS00800
*          GRPCHK(61),GROUPS(124)                            CLS00810
COMMON /CLASS/ APRFLG,BMCOMB,BMFEAT,BMFLG,NOCAT,THIJ1,IDATA1, CLS00820
*          NFILE,STATKY,CATNAM(60),                          CLS00830
3          CLSSYM(60),CON(60),DET(60),FLDESC,FLDINF(6),      CLS00840
4          KCLSNA(60),NOCTCL(60),SUBCAT(60)                  CLS00850
*          ,NOCHAN,CHNVEC(30)                                 CLS00860
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,    CLS00870
*          HISFIL,WISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFIL,  CLS00880
*          DRUMAD,DRMWDOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL  CLS00890
*          ,NHSTUN,NHSTFI,SCRUN,MAPFIL                       CLS00900
*          ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, CLS00910
*          CRDUNT,PRUNT,RANDIO                                CLS00920
CSEND                                                       CLS00930
-----                                                    CLS00940
CLS00950
CLS00960
CLS00970
REAL CON,DET,COVMTX(VAHSZ2,NOSUR2)                          CLS00980
REAL AVEMTX(NOFET2,NOSUB2),APRIOR(1),VEC(30),BXXXT(120)    CLS00990
REAL SUM,SUMTR,DUM(60),RERROR(60),APR,COV(465)             CLS01000
REAL RELERR                                                 CLS01010
-----                                                    CLS01020
CLS01030
CLS01040
CLS01050
COMMON /SCRACH/SCR1(2000),SCR2(10500)                       CLS01060
-----                                                    CLS01070
CLS01080
CLS01090
EQUIVALENCE ( VEC(1), SCR1(1) ), ( BXXXT(1), SCR1(31) ),   CLS01100
1          ( SUM, SCR1(151) ), ( SUMTR, SCR1(152) )         CLS01110
2          , ( DUM(1), SCR1(153) ), ( APR, SCR1(215) )      CLS01120
3          , ( RERROR(1), SCR1(216) ), ( COV(1), SCR1(280) ) CLS01130
-----                                                    CLS01140
CLS01150
CLS01160
CLS01170
CLS01180
CLS01190
DIMENSION CLSMTX(1),FLDMTX(4,NOFLD2),SUBNO(1),SUBDES(1),   CLS01200
* VERTEX(2,TOTVT2),KATNO(1)                                CLS01210
CLS01220
DATA BCDTWO/'2'/,BCDFOR/'4'/,DASH/'----'/                 CLS01230
DATA LPHN/'/'/'/'/'/'/'/'/'/'/'/'/'/'/'/'/'/'/'/'/'/'/' CLS01240
-----                                                    CLS01250
CLS01260
CLS01270
CLS01280
CLS01290
CLS01300
CLS01310
REAL BMATX(BMCOMB,BMFEAT)                                   CLS01320
REAL AVENEW(BMCOMB,BMFEAT),COVNEW(BMFLG,NOSUR2)           CLS01330
-----                                                    CLS01340
CLS01350
CLS01360
CLS01370
INCLUDE COM4K6,LIST                                         CLS01380
-----                                                    CLS01390
CLS01400
CLS01410
CLS01420
CLS01430
CLS01440
CLS01450
CLS01460
CLS01470
CLS01480
CLS01490
HEADER RECORD NO. 2 FOR MAPTAP                              CLS01500
IF (NOCAT .GT. 0) NOCAT1 = NOCAT                            CLS01510
IF ( NOCAT .LE. 0 ) NOCAT1 = NOCLS2                         CLS01520
WRITE(MAPTAP) (CATNAM(I),I=1,NOCAT1), (CLSMTX(I),I=1,NOCLS2), CLS01530
1          (SUMNO(I),I=1,NOCLS2), (SURDES(I),I=1,NOSUR2),   CLS01540
2          ((FLDMTX(I,J),I=1,4),J=1,NOFLD2), ((VERTEX(I,J),I=1,2), CLS01550
3          J=1,TOTVT2), (SUBCAT(I),I=1,NOSUB2), (CLSVC2(I),   CLS01560
4          I=1,NOSUR2), (KATNO(I),I=1,NOCLS2), (KEPPTS(I),  CLS01570
5          I=1,NOSUB2)                                       CLS01580
WRITE OUT TRAINING FIELD INFORMATION                       CLS01590
-----                                                    CLS01600

```



FILE: CLSFY1

```

C
65 FORMAT(3X, '*** CLASSIFICATION STUDY *** MAPTAP FILE ',5X, I2//)
10 CALL WRFLD(FLDMTX, VERTEX, NOFLD2, CLSMTX, SURDES)

      PUBLISH THE CLASSES AND CHANNELS (WITH SPECTRAL BAND) TO BE
      CONSIDERED IN CLASSIFICATION.

WRITE(6, HEAD)
WRITE(6, 65) NFILE
WRITE(6, 20)
20 FORMAT( '////T27, 'SUBCLASSES CONSIDERED', T90, 'CHANNELS CONSIDERED',
*//T21, 'SYMBOL', T32, 'SUBCLASS', T45, 'A PRIOR', T88, 'TRAINING RECOGN
*ITION')
      II = NOSUR2
      IF ( II .LT. NOFET2 ) II = NOFET2
      DO 30 I=1, II
      WRITE(6, 40)
      IF ( I .LE. NOSUR2 ) WRITE(6, 50) CLSSYM(I), SURDES(I), APRIOR(I)
      IF ( I .LE. NOFET2 ) WRITE(6, 60) FETVC2(I), CHNVEC(I)
30 CONTINUE
40 FORMAT(' ')
50 FORMAT(' ', T23, A1, T33, A4, T45, F7.4)
60 FORMAT(' ', T91, I2, T103, I2)

      SAVE AND PUBLISH THE MEAN AND COVARIANCE
      -----
      IF R-MATRIX IS AVAILABLE. TRANSFORM THE COVARIANCE MATRIX AND
      MEAN VECTOR

      TEMPF2 = NOFET2
      IF( RMFLG .LE. 0 ) GO TO 611
      DO 610 NCLS=1, NOSUR2
      DO 605 K=1, RMCOMB
      RRM102 = ( K * (K-1) ) / 2
      DO 603 C=1, R
      I = RRM102 + C
      SUMTR = 0.0
      DO 602 J=1, NOFET2
      SUM = 0.0
      DO 601 K=1, NOFET2
      II = ( K * (K-1) ) / 2 + J
      IF( J .GT. K ) II = ( J * (J-1) ) / 2 + K
      601 SUM = SUM + RMATR(R, K) * COVMTX(II, NCLS)
      602 SUMTR = SUMTR + RMATR(C, J) * SUM
      603 RXXXT(I) = SUMTR
      SUM = 0.0
      DO 604 J=1, NOFET2
      604 SUM = SUM + RMATR(R, J) * AVEMTX(J, NCLS)
      VEC(R) = SUM
      605 CONTINUE
      DO 606 J=1, I
      606 COVNEW(J, NCLS) = RXXXT(J)
      DO 607 J=1, RMCOMB
      607 AVNEW(J, NCLS) = VEC(J)
      610 CONTINUE
      VARSZ2 = RMFLG
      NOFET2 = RMCOMB
      611 CONTINUE
      GO TO R0
      70 CONTINUE

      OBTAIN THE SUBCLASS-PAIR THRESHOLDS, FOR USE BY CLSFY2/CONTEX

      IF (NOCAT .GT. 0) GO TO 612
      NPL1 = NOFET2 + 1
      CALL THRESH(NOSUR2, NOFET2, NPL1, APRIOR, AVEMTX, COVMTX, DET, VARSZ2,
      1 SCR2(1), SCR2(901), SCR2(1801), SCR2(1831), SCR2(1861),
      2 SCR2(2791), SCR1(TH1J1))
      612 CONTINUE

      NOFET2 = TEMPF2
      RETURN
      R0 III = 0

```

CLS01590  
 CLS01600  
 CLS01610  
 CLS01620  
 CLS01630  
 CLS01640  
 CLS01650  
 CLS01660  
 CLS01670  
 CLS01680  
 CLS01690  
 CLS01700  
 CLS01710  
 CLS01720  
 CLS01730  
 CLS01740  
 CLS01750  
 CLS01760  
 CLS01770  
 CLS01780  
 CLS01790  
 CLS01800  
 CLS01810  
 CLS01820  
 CLS01830  
 CLS01840  
 CLS01850  
 CLS01860  
 CLS01870  
 CLS01880  
 CLS01890  
 CLS01900  
 CLS01910  
 CLS01920  
 CLS01930  
 CLS01940  
 CLS01950  
 CLS01960  
 CLS01970  
 CLS01980  
 CLS01990  
 CLS02000  
 CLS02010  
 CLS02020  
 CLS02030  
 CLS02040  
 CLS02050  
 CLS02060  
 CLS02070  
 CLS02080  
 CLS02090  
 CLS02100  
 CLS02110  
 CLS02120  
 CLS02130  
 CLS02140  
 CLS02150  
 CLS02160  
 CLS02170  
 CLS02180  
 CLS02190  
 CLS02200  
 CLS02210  
 CLS02220  
 CLS02230  
 CLS02240  
 CLS02250  
 CLS02260  
 CLS02270  
 CLS02280  
 CLS02290  
 CLS02300  
 CLS02310  
 CLS02320  
 CLS02330  
 CLS02340  
 CLS02350  
 CLS02360  
 CLS02370



FILE1 CLSFY1

```
21TE1// 35X, DETERMINANT = .F20.4//5X, ***** TERMINATING PROGRAM CLS03170
3 EXECUTION ***** ) CLS03180
C CLS03190
RADELG = 99 CLS03200
GO TO 195 CLS03210
183 APR = APRIOR(NCLS)**(-2) CLS03220
CON(NCLS) = ALOG( APR * DET(NCLS) ) CLS03230
CLS03240
OBTAIN THE RELATIVE ERROR OF FACTORED MATRIX ( RATIO OF CLS03250
EUCLIDEAN NORM OF DIFFERENCE, K-LDL* , TO EUCLIDEAN NORM OF K) CLS03260
ERROR(NCLS) = RELERR( COVMTX(1,NCLS),COV,NOFET2,VARSZ2) CLS03270
CLS03280
195 CONTINUE CLS03290
ERROR TERMINATION, IF ONE OR MORE SUBCLASSES HAVE AN INVALID CLS03300
(SINGULAR, OR NON-POSITIVE DEFINITE) COVARIANCE MATRIX CLS03310
IF( BADFLG .GT. 0) CALL EXIT CLS03320
CLS03330
----- CLS03340
OUTPUT THE ( MODIFIED ) CHOLESKY FACTORIZATION OF THE CLS03350
COVARIANCE MATRIX, ON THE CLASSIFICATION OUTPUT FILE, MAPTAP CLS03360
----- CLS03370
CLS03380
CLS03390
CLS03400
CLS03410
CLS03420
CLS03430
CLS03440
CLS03450
CLS03460
CLS03470
CLS03480
CLS03490
CLS03500
CLS03510
CLS03520
CLS03530
CLS03540
CLS03550
CLS03560
CLS03570
CLS03580
CLS03590
CLS03600
CLS03610
CLS03620
CLS03630
CLS03640
CLS03650
200 WRITE(4,220) CLSMTX(L),L,SURDES(III),III,DET(III),CON(III) CLS03660
CALL WRITMX(COVMTX(1,III),NOFET2,RCDFOR) CLS03670
C CLS03680
WRITE(6,205) ERROR(III) CLS03690
205 FORMAT(1X, ' * RELATIVE ERROR ( EUCLIDEAN NORM (K-LDL*)/EUCLIDEAN CLS03700
INORM K ) = ' , F15.8 // ) CLS03710
C CLS03720
INC = INC + 1 CLS03730
210 CONTINUE CLS03740
220 FORMAT(1H0// T50, MULTISPECTRAL CHARACTERISTICS FOR // T57, A4, CLS03750
1 ( CLASS, 13, ' ) // T56, A4, 2X, ( SUBCLASS, 13, ' ) // CLS03760
2 1H0, DETERMINANT = .F25.4 / 1H0, PROR. DENSITY FU CLS03770
2 NCTION - CONSTANT TERM = .F10.4// 1H0, COVARIANCE MATRIX (CHOLESKY CLS03780
3 FACTORIZATION) : / ) CLS03790
C CLS03800
C CLS03810
230 CONTINUE CLS03820
GO TO 70 CLS03830
END CLS03840
```

FILE: CLSFY2

```
SUBROUTINE CLSFY2(COVMTX,AVEMTX,FLOMTX,CLSMTX,SUBDES,SUBNO,
KATNO,BMATRX)
IMPLICIT INTEGER (A-M,O-Z)
-----
CALL... CALL CLSFY2(COVMTX,AVEMTX,FLOMTX,RUNMTX,CLSMTX,BMATRX)
ARGS... COVMTX : LOCATION OF THE COVARIANCE MATRICES
          ( IN 'SYMETTRIC' STORAGE) FOR NOCLS2 TRAINING
          CLASSES
          AVEMTX : LOCATION OF THE MATRIX OF TRAINING CLASS MEAN
          VECTORS (NOFET2 MEANS PER CLASS)
          FLOMTX : LOCATION OF MATRIX OF TRAINING FIELD(S)
          INFORMATION
          CLSMTX : LOCATION OF MATRIX OF TRAINING CLASS NAMES
          BMATRX : B-TRANSFORMATION MATRIX,IF AVAILABLE
          SURDES : LOCATION OF SUBCLASS NAMES
          SURNO : LOCATION OF NO. OF SUBCLASSES IN EACH CLASS
          KATNO : CATEGORY - CLASS CORRESPONDENCE ARRAY
          BMATRX : LOCATION OF THE 'B'-TRANSFORMATION MATRIX,
          IF AVAILABLE, FOR TRANSFORMATION OF INPUT
          SAMPLE VECTOR IN SUBR. CONTEX
PURPOSE... CLASSIFIES THE SET OF SAMPLES ( MULTI-CHANNEL DATA
POINTS) ON EACH SCAN LINE OF THE SET OF SCAN LINES
PRESCRIBED BY THE 'FIELD DEFINITION' CARD INPUT TO
THE CLASSIFICATION PROCESSOR. CLASSIFICATION IS
PERFORMED BY THE METHOD OF MAXIMUM LIKELIHOOD
(MINIMUM PROBABILITY OF MIS-CLASSIFICATION) IN
SUBROUTINE CONTEX . THE DIMENSIONALITY OF THE SAMPLE
IS PRESCRIBED BY CONTROL CARD INPUT TO THE PROCESSOR
(BMCOMB) OF CHANNELS (BMFEAT) IN THE 'B' - MATRIX,
('CHANNELS' ) OR BY THE NO. OF LINEAR COMBINATIONS
IF AVAILABLE .
RETURNS... 1. IF STANDARD CLASSIFIER IS USED, THE SUBCLASS
NUMBER AND PROBABILITY DENSITY FUNCTION
VALUE FOR EACH POINT OF EVERY SCAN LINE OF THE FIELD
IS OUTPUT ON THE CLASSIFICATION OUTPUT FILE, MAPTAP.
2. IF CATEGORY CLASSIFIER IS USED, THE SUBCLASS
NUMBER OF THE SUBCLASS WITH THE LARGEST PROBABILITY
DENSITY FUNCTION WITHIN THE CHOSEN CATEGORY AND THE
PROBABILITY DENSITY FUNCTION VALUE OF THE CHOSEN
CATEGORY FOR EACH POINT OF EVERY SCAN LINE OF THE
FIELD IS OUTPUT ON THE CLASSIFICATION OUTPUT FILE.
-----
INCLUDE COMHKA.LIST
INCLUDE COMHK1.LIST
COMMON/INFO/M/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
AVAR2,COVAR2,CLSID2,SUBNO2,SURDS2,FLDSV2,VERTX2,
FETVC2(30),SUHVC2(75),SURPTH(75),CLSVC2(60),
KEPPTS(60),NOGMP,GMPNAM(60),GMPDFX(61),
GRPCHK(61),GROUPS(124)
COMMON /CLASS/ APRFLG,HMCOMB,HMFEAT,HMFLG,NOCAT,THIJ1,IDATA1,
NFILE,STATKY,CATNAM(60),
CLASSYM(60),CON(60),DET(60),FLDESC,FLDINF(6),
KCLSN(60),NOCTCL(60),SUBCAT(60)
,NOCHAN,CHNVEC(30)
COMMON/GLUBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,HMFILE,BMKEY.
```

FILE: CLSPY2

```

* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE, CLS00800
* DRUMAD,DRMWS,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVL CLS00810
* NMSTUN,NMSTFI,SCTRUN,MAPFIL CLS00820
* DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCMUNT, CLS00830
* CHDUNT,PTUNT,RANDIO CLS00840
CSEND CLS00850
* REAL CON,DET, VR(1000),COVMTX(VARSZ2,NOSUB2),AVEMTX(NOFET2, CLS00860
* NOSUB2),BMATRIX CLS00870
CLS00880
CLS00890
CLS00900
CLS00910
CLS00920
DIMENSION FLOMTX(4,NOFLD2),CLSMTX(1),VERTCS(22) CLS00930
1 COL(3,110),OUT(110),IR(1000) CLS00940
* SURNO(1),SURDES(1),KATNO(1) CLS00950
CLS00960
CLS00970
*** NOTE : THE IDATA ARRAY IS USED FOR INPUT OF THE SCAN LINE TO CLS00980
BE CLASSIFIED TO SURR, CONTEX, AND ALSO TO STORE THE CLS00990
CLASSIFIED SCAN LINE, BY CONTEX, THE 'IR' AND 'VR' CLS01000
ARRAYS, EQUIVALENCED TO THE IDATA ARRAY, ARE CLS01010
LOCATIONS USED BY CONTEX TO OUTPUT THE CLASSIFIED CLS01020
SCAN LINE AND THE PROBABILITY DENSITY FUNCTION CLS01030
VALUES, RESPECTIVELY. CLS01040
CLS01050
CLS01060
CLS01070
CLS01080
CLS01090
----- CLS01100
INCLUDE COMPK2.LIST CLS01110
CLS01120
EQUIVALENCE (FLDINF(1),LINSTR), (FLDINF(2),LINEND), CLS01130
(FLDINF(3),LININC), (FLDINF(4),SAMSTR), CLS01140
(FLDINF(5),SAMEND), (FLDINF(6),SAMINC), CLS01150
(FLDINF(7),FLDTYP) CLS01160
CLS01170
----- CLS01180
CLS01190
----- CLS01200
CLS01210
COMMON /SCRACH/ IDATA(12500) CLS01220
CLS01230
CLS01240
CLS01250
----- CLS01260
CLS01270
CLS01280
CLS01290
CLS01300
----- CLS01310
CLS01320
CLS01330
DATA LINMAX/1000/, ENDBCD/'SEND'/, DASH/'-----'/ CLS01340
CLS01350
CLS01360
EQUIVALENCE (IR,COL) CLS01370
DATA LEWN/'( /,HLANK/' // CLS01380
CLS01390
----- CLS01400
CLS01410
----- CLS01420
CLS01430
CLS01440
CLS01450
CLS01460
READ THE FIELD DESCRIPTION CARD,CONTAINING LINE-SAMPLE COORDINATES CLS01470
OF THE FIELD TO BE CLASSIFIED CLS01480
----- CLS01490
CALL TAPHDR(DATAPE,DATAFIL) CLS01500
CLS01510
10 CONTINUE CLS01520
PSTMS = 0 CLS01530
ICK = LAHEAD(FLDESC,VERTCS,FLDINF,NC) CLS01540
IF (ICK.EQ.0) GO TO 160 CLS01550
IF (ICK.EQ.-1).OR.(ICK.EQ.-2) GO TO 10 CLS01560
CLS01570
30 CONTINUE CLS01580

```

ORIGINAL PAGE IS  
OF POOR QUALITY

11-13

149

FILE: CLSFY2

```
WRITE(6,HEAD)
NC1 = 2*NC
NV = NC - 1
WRITE(6,35)FLDESC,NV,FLDINF(6),FLDINF(3),(LPRN,VERTCS(I),
* VERTCS(I+1),I=1,NC1,2)
30 FORMAT(///156,'AREA OF CLASSIFICATION'/// 36X,'SAMPLE',3X,
1 'LINE', / 5X,'FIELD NAME',3X,'NO. OF VERTICES',3X,'INC. ',
2 3X,'INC.',30X,'VERTICES' / 7X,A4,14X,I2,10X,I2,6X,I2,5X,
3 5(A1,I4,' ',I4,' '),2X) / (52X,5(A1,I4,' ',I4,' '),2X)/)
SAVE CLASSIFIED FIELD INFORMATION ON MAPTAP
-----
50 CONTINUE
INITIALIZE TAPE READING FOR THIS FIELD
CALL FLDINT(FLDINF,CHNVEC,NUCHAN)

LINES = (LINEND-LINSTR)/LININC + 1
PTS = (SAMEND-SAMSTR)/SAMINC + 1
IF (PTS .GT. LINMAX) WRITE(6,55)
55 FORMAT( // 5X,'WIDTH OF RECTANGULAR FIELD SURROUNDING CLASSIFICATI
*ON FILD CANNOT EXCEED 1000 POINTS. ')
IF (PTS .GT. LINMAX) CALL CMERR

SCANLN = PTS * NOFET2
IF (SCANLN .GT. (12500-IDATA1+1)) GO TO 170

FIELD RECORD FOR MAPTAP
WRITE(MAPTAP)(FLDINF(I),I=1,6),PTS,LINES,FLDESC,NC,
* (VERTCS(I),I=1,NC),(VERTCS(I+NC),I=1,NC)

CLASSIFY THE FIELD
-----
CALL SETMRG(66,0,66)

WRITE(6,HEAD)
CALL MAPHDG(NOCAT,CLSSYM,CATNAM,KATNO,CLSMTX,SUBNO,SUBDES)
WRITE(6,800)
800 FORMAT(///)
J = 0
DO AN I=SAMSTR,SAMEND,SAMINC
J = J+1
COL(1,J) = I/100
COL(2,J) = MOD(I,100)/10
COL(3,J)=MOD(I,10)
IF (J.EQ.110) GO TO 90
90 CONTINUE
DO 100 I=1,3
100 WRITE(6,110)(COL(I,K),K=1,J)
110 FORMAT(' ',9X,110I1)
WRITE(6,115)
115 FORMAT( / )

ILINE=LINSTR-LININC
DO 140 JLINE=1,LINES
CALL LTNRD(IDATA(IDATA1),ENDTAP)
IF (ENDTAP .EQ. -1) GO TO 150
ILINE=ILINE+LININC

*** CLASSIFY THE SCAN LINE IN IDATA

CALL STANDARD CLASSIFIER

IF (NOCAT .LE. 0) CALL CGITEX(NOFET2,NOSUB2,PTS,AVEMTX,COVMTX,
* BMATRX,IDATA(IDATA1),VERTCS,NC,IR,VR,ILINE,IDATA(THIJI))

CALL CATEGORY CLASSIFIER

IF (NOCAT .GT. 0) CALL CATGRY(NOFET2, PTS,AVEMTX,COVMTX,
* IR,VR,HMATRX,IDATA(IDATA1),ILINE,VERTCS,NC,PTSTHS)
```

FILE: CLSFY2

```
C
DO 120 JKL=1,J
ISYMB = IR(JKL)
IF ( ISYMB .EQ. 0 ) OUT(JKL) = BLANK
IF ( ISYMB .NE. 0 ) OUT(JKL) = CLSSYM(ISYMB)
120 CONTINUE
WRITE (6,130) ILINE,(OUT(I),I=1,J)
130 FORMAT(' ',I5,4X,110A1)
C
C-----
SAVE CLASSIFIED INFORMATION ON MAPTAP
C-----
DATA RECORD FOR MAPTAP
C
WRITE(MAPTAP) ILINE,(IR(I),I=1,PTS),(VR(I),I=1,PTS)
140 CONTINUE
IF (PTSTHS .GT. 0) WRITE(6,145) PTSTHS
145 FORMAT(///9X,'AS THE COMPUTER CANNOT EXPONENTIATE A NUMBER SMALLER
* R THAN EXP(-88),' ,I6,' PTS WERE NOT CLASSIFIED IN THIS FIELD')
150 ILINE = 0
C
C-----
END OF FIELD RECORD FOR MAPTAP
C-----
WRITE(MAPTAP) ILINE,(IR(I),I=1,PTS),(VR(I),I=1,PTS)
GO TO 10
GO HOME
C-----
160 PTS = 0
C
C-----
END OF RUN RECORD FOR MAPTAP
C-----
WRITE(MAPTAP) (FLDINF(I),I=1,6),PTS,LINES,FLDESC,NC,
* (VERTCS(I),I=1,NC),(VERTCS(I+NC),I=1,NC)
CALL SFTMRG(66,4,62)
RETURN
170 WRITE(6,175)
175 FORMAT(//5X,'TOO MUCH DATA REQUESTED. DO ONE OF THE FOLLOWING:
* /7X, '1) FOR STANDARD CLASSIFIER - /11X, 'REDUCE PARAMETERS SUCH TH
* AT /11X, '(NO. OF SUBCLASSES-1)*(NO. OF SUBCLASSES-2)/2 + NO. OF SU
* RCLASSES + (PTS PER SCAN LINE)*(NO. OF CHANNELS) + 12500' /7X,
* 2) FOR CATEGORY CLASSIFIER - /11X, 'REDUCE DATA REQUESTED SUCH T
* HAT / 11X, '(PTS PER SCAN LINE)*(NO. OF CHANNELS) + 12500.')
CALL CMERR
END
```

CLS02380  
CLS02390  
CLS02400  
CLS02410  
CLS02420  
CLS02430  
CLS02440  
CLS02450  
CLS02460  
CLS02470  
CLS02480  
CLS02490  
CLS02500  
CLS02510  
CLS02520  
CLS02530  
CLS02540  
CLS02550  
CLS02560  
CLS02570  
CLS02580  
CLS02590  
CLS02600  
CLS02610  
CLS02620  
CLS02630  
CLS02640  
CLS02650  
CLS02660  
CLS02670  
CLS02680  
CLS02690  
CLS02700  
CLS02710  
CLS02720  
CLS02730  
CLS02740  
CLS02750  
CLS02760  
CLS02770  
CLS02780  
CLS02790  
CLS02800  
CLS02810  
CLS02820  
CLS02830  
CLS02840  
CLS02850  
CLS02860  
CLS02870

FILE: CONTEX

```

SURROUTINE CONTEX (NCHAN, NC, NPTS, AVE, COV, BMATR, IDATA, VERTCS, VT,
* IR, VR, ILINE, TH)
-----
INCLUDE COMBK2, LIST
COMMON /CLASS/ APRFLG, BMCOMB, BMFEAT, BMFLG, NOCAT, TH(1), IDATA(1),
* NFILE, STATKY, CATNAM(60),
* CLASSYM(60), CON(60), DET(60), FLDESC, FLDINF(6),
* KCI, SNA(60), NOCTCL(60), SUBCAT(60)
, NOCHAN, CHNVEC(30)
-----
INTEGER BMFLG, BMCOMB, BMFEAT
* SAMSTR, SAMINC, SAMEND
INTEGER VERTCS, VT, FL
-----
LOGICAL BMFLAG, KDI
-----
DIMENSION IDATA(1), AVE(1), COV(1), DATA(30)
1, JORDER(60), BMATR(BMCOMB, BMFEAT), JTEST(60), IR(1), VR(1),
* NCNT(60), DM(30), FL(22), VERTCS(1)
-----
DIMENSION TH(1)
-----
USING THE "MODIFIED" CHOLESKY DECOMPOSITION OF THE COVAR. MATRIX,
THE ROUTINE COMPUTES THE PROBABILITY DENSITY FUNCTION AND OBTAINS THE
MAXIMUM PROBABILITY ( "MAXIMUM LIKELIHOOD" ) OVER ALL CLASSES OF
THE SET OF TRAINING CLASSES, FOR ASSIGNING A CLASS TO EACH RESOLUTION
ELEMENT ( "PIXEL" ) IN THE INPUT SAMPLE VECTOR ( INPUT SCAN LINE ).
THE PRE-COMPUTED CLASS-PAIR THRESHOLDS, IN TH, ARE USED TO MINIMIZE
THE NUMBER OF CLASS PROBABILITY DENSITY FUNCTIONS (PDF) COMPUTED
TO OBTAIN THE MAXIMUM PDF FOR A GIVEN SAMPLE BEING CLASSIFIED.
THE IRVR ARRAY IS USED TO RETURN THE CLASS NUMBER AND PDF VALUE
FOR EACH SAMPLE ON THE INPUT SCAN LINE
-----
CALL... CALL CONTEX (NCHAN, NC, NPTS, AVE, COV, BMATR, IDATA, VERTCS,
VT, IR, VR, ILINE)
-----
ARGS.. NCHAN : THE NUMBER OF CHANNELS TO BE USED IN
CLASSIFICATION OF EACH DATA SAMPLE.
NC : THE NUMBER OF SUBCLASSES ( TRAINING CLASSES,
CONTINUE FOR WHICH COVARIANCE MATRICES, MEAN VECTORS,
AND CLASS-PAIR THRESHOLDS ARE AVAILABLE -
REPRESENTS THE MAXIMUM NUMBER OF POSSIBILITIES
FOR CLASSIFICATION OF EACH DATA SAMPLE.)
NPTS : THE NUMBER OF INPUT DATA POINTS (PER CHANNEL)
ON THE RECTANGULAR FIELD
VERTCS : VERTICES OF FIELD TO BE CLASSIFIED

```

CON00010  
CON00020  
CON00030  
CON00040  
CON00050  
CON00060  
CON00070  
CON00080  
CON00090  
CON00100  
CON00110  
CON00120  
CON00130  
CON00140  
CON00150  
CON00160  
CON00170  
CON00180  
CON00190  
CON00200  
CON00210  
CON00220  
CON00230  
CON00240  
CON00250  
CON00260  
CON00270  
CON00280  
CON00290  
CON00300  
CON00310  
CON00320  
CON00330  
CON00340  
CON00350  
CON00360  
CON00370  
CON00380  
CON00390  
CON00400  
CON00410  
CON00420  
CON00430  
CON00440  
CON00450  
CON00460  
CON00470  
CON00480  
CON00490  
CON00500  
CON00510  
CON00520  
CON00530  
CON00540  
CON00550  
CON00560  
CON00570  
CON00580  
CON00590  
CON00600  
CON00610  
CON00620  
CON00630  
CON00640  
CON00650  
CON00660  
CON00670  
CON00680  
CON00690  
CON00700  
CON00710  
CON00720  
CON00730





FILE: CONTEX

```
C      IF ( BMFLAG ) GO TO 5
C      DO 4 I=1,NCHAN
      IDUM = NPTS * ( I - 1 ) + II
4     DATA(I) = IDATA(IDUM)
      GO TO 15
C
C 5     DO 7 I=1,BMCOMB
      SUM = 0.0
      DO 6 K=1,NCHAN
      INDUM = NPTS * ( K - 1 ) + I
      FDATA = IDATA(INDUM)
6     SUM = SUM + BMATR(I,K) * FDATA
C
C 7     DATA(I) = SUM
C
C CALC. THE LIKLIHOOD VALUES (PROBABILITIES, IF YOU WILL)
C
C 15    TFMAX = -1.0E35
      JI = 0
C
C      COMPUTE THE PDF FOR CLASS JJ
C
C 20    JJMI = JJ - 1
      LC = NPC * JJMI
      LOCATION (-1) OF COV. MATRIX, CLASS JJ
C
      IMN = JJMI * NV
      LOCATION (-1) OF MEAN VECTOR, CLASS JJ
C
      KM = IMN + 1
      S = DATA(I) - AVE(KM)
      DM(1) = S
      LC = LC + 1
      TF = CON(JJ) + ( S * S )/COV(LC)
      IF(KDI) GO TO 146
C
C LOOP FOR COMPUTING THE KD-TH ELEMENT OF Y (=L**1 * (X-M) ) , WHICH
C IS STORED IN S
C
      DO 145 KD=2,NV
      KM = IMN + KD
      S = DATA(KD) - AVE(KM)
      JI = KD - 1
      DO 140 LD = 1,JI
      LC = LC + 1
140    S = S - COV(LC) * DM(LD)
C
      DM(KD) = S
      LC = LC + 1
C
C COMPUTE THE KD-TH TERM IN :  $\frac{1}{2} * Y * D^{**1} * Y$ 
C =  $\frac{1}{2} * (X-M) * K^{**1} * (X-M)$ 
C
C 145    TF = TF + ( S * S )/COV(LC)
C
C 146    TF = -.5 * TF
C
C
C TEST THIS SAMPLE PDF FOR CLASS JJ - IF GREATER THAN THE PDF FOR
C CURRENT CLASS IC, SET IC = JJ, TEST THE CLASS-PAIR THRESHOLDS FOR
C OTHER POSSIBLE CLASSES FOR THIS SAMPLE - IF THE PDF FOR CURRENT
C CLASS IS EXCEEDED BY ANY CLASS-PAIR THRESHOLD, EVALUATE THE PDF
C FOR THE OTHER CLASS OF THE CLASS-PAIR, AND REPEAT THE TEST FOR
C MAX. PDF. IF ALL PDF'S FOR WHICH CLASS-PAIR THRESHOLDS HAVE
C DICTATED TO BE TESTED HAVE BEEN EVALUATED, AND THE CURRENT PDF
C FOR CLASS IC IS THE MAX. PDF OF ALL PDF'S EVALUATED,
C CLASSIFY THE SAMPLE AS CLASS IC
C
C IF( TF .LE. TFMAX) GO TO 149
      TFMAX = TF
      IC = JJ
149    JTEST(JJ) = 0
150    JI = JI + 1
      IF( JI .GT. NC ) GO TO 152
      J = JORDER(JI)
C
CON01520
CON01530
CON01540
CON01550
CON01560
CON01570
CON01580
CON01590
CON01600
CON01610
CON01620
CON01630
CON01640
CON01650
CON01660
CON01670
CON01680
CON01690
CON01700
CON01710
CON01720
CON01730
CON01740
CON01750
CON01760
CON01770
CON01780
CON01790
CON01800
CON01810
CON01820
CON01830
CON01840
CON01850
CON01860
CON01870
CON01880
CON01890
CON01900
CON01910
CON01920
CON01930
CON01940
CON01950
CON01960
CON01970
CON01980
CON01990
CON02000
CON02010
CON02020
CON02030
CON02040
CON02050
CON02060
CON02070
CON02080
CON02090
CON02100
CON02110
CON02120
CON02130
CON02140
CON02150
CON02160
CON02170
CON02180
CON02190
CON02200
CON02210
CON02220
CON02230
CON02240
CON02250
CON02260
CON02270
CON02280
CON02290
```



FILE: FALSY

```
SUBROUTINE FALSY(XL,XU,C,FXL,FXU,KC,XN,KT,T,K,KP1,S1,S2,U1,U2,RR) FAL00010
DIMENSION S1(K,K),S2(K,K),U1(K),U2(K),BB(K,KP1) FAL00020
RE=.1E-05*C FAL00030
IF(RF.LT..000001) RE=.000001 FAL00040
IF(KC.FQ.0) GO TO 7 FAL00050
FXL = G(XL,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00060
FXU = G(XU,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00070
7 R=(C-FXL)/(FXU-FXL) FAL00080
X=XL+R*(XU-XL) FAL00090
FX = G(X,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00100
FM=ABS(FX-C) FAL00110
I = 0 FAL00120
C GO TO P FAL00130
9 FXN = G(XN,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00140
I=I+1 FAL00150
E=ABS(FXN-C) FAL00160
C IF(E.GT.EM) GO TO 25 FAL00170
IF(E.LT.RE) RETURN FAL00180
EM=E FAL00190
IF(XN.LT.X) GO TO 12 FAL00200
XH=.5*(XU+XN) FAL00210
FXH = G(XH,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00220
IF(ABS(FXH-C).LT.RE) GO TO 30 FAL00230
IF(FXH.GT.C) GO TO 14 FAL00240
XL=XN FAL00250
FXL=FXN FAL00260
X=XH FAL00270
FX=FXH FAL00280
GO TO 15 FAL00290
14 XL=X FAL00300
FXL=FX FAL00310
X=XN FAL00320
FX=FXN FAL00330
XU=XH FAL00340
FXU=FXH FAL00350
GO TO 15 FAL00360
12 XH=.5*(XL+XN) FAL00370
FXH = G(XH,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00380
IF(ABS(FXH-C).LT.RE) GO TO 30 FAL00390
IF(FXH.GT.C) GO TO 13 FAL00400
XL=XH FAL00410
FXL=FXH FAL00420
XU=X FAL00430
FXU=FX FAL00440
X=XN FAL00450
FX=FXN FAL00460
GO TO 15 FAL00470
13 XU=XN FAL00480
FXU=FXN FAL00490
X=XH FAL00500
FX=FXH FAL00510
15 IF( I .EQ. 25) RETURN FAL00520
GO TO P FAL00530
25 XN=X FAL00540
RETURN FAL00550
30 XN=XH FAL00560
DFTUPN FAL00570
C FOLLOWING CODE FITS A QUADRATIC TO THE THREE POINTS FAL00580
C XL, X, XU WHICH IS AN APPROXIMATION OF THE FUNCTION G( H(X) ) FAL00600
C WITHIN THE DEFINED INTERVAL. A ROOT, XN, OF THE APPROXIMATING FAL00610
C QUADRATIC IS RETURNED TO BE USED AS A TRIAL SOLUTION OF FAL00620
C G( H(X) ) = C2 - C1 FAL00630
C FAL00640
P W1 = X - XL FAL00650
W2 = X*X - XL*XL FAL00660
W3=FX-FXL FAL00670
W4=XU-XL FAL00680
W5=XU*XU-XL*XL FAL00690
W6=FXU-FXL FAL00700
W7=C-FXL FAL00710
A=W1*W6-W3*W4 FAL00720
IF(ABS(A).LT.17.E-7)GO TO 44 FAL00730
B=W3*W5-W2*W6 FAL00740
F=-XL*XL*A-XL*W7*(W2*W4-W1*W5) FAL00750
D=B*B-4.*A*F FAL00760
IF(D.LT.0.)GO TO 44 FAL00770
D=SQRT(D) FAL00780
FAL00790
```

11-20

156

FILE: FALSY

```
XN=(-B+D)/(2.0*A)
IF (XN.GT.XU) XN=(-B-D)/(2.0*A)
GO TO 9
44 XN=X
GO TO 9
END
```

```
FAL00800
FAL00810
FAL00820
FAL00830
FAL00840
FAL00850
```



FILE: GJR

```
SUBROUTINE GJR(A,NC,NR,N,MC,*,JC,V)
DIMENSION A(NR,NC),JC(1),V(2)
CALL OVERFL(IJK)
IW=V(1)
M=1
S=1.
L=N+(MC-N)*(IW/4)
KD=2-MOD(IW/2,2)
IF(KD.EQ.1)V(1)=0.
C
JLS111069
KI=2-MOD(IW,2)
GO TO (5,20),KI
5
DO 10 I=1,N
10
JC(I)=I
20
DO 91 I=1,N
GO TO (22,21),KI
21
M=I
22
IF (I.EQ.N) GO TO 60
X=-1.
DO 30 J=I,N
IF (X.GT.ABS(A(J,I))) GO TO 30
X=ABS(A(J,I))
K=J
30
CONTINUE
IF (K.EQ.I) GO TO 60
S=-S
V(2)=-V(2)
C
JLS111069
GO TO (35,40),KI
35
MU=JC(I)
JC(I)=JC(K)
JC(K)=MU
40
DO 50 J=M,L
X=A(I,J)
A(I,J)=A(K,J)
A(K,J)=X
50
IF (ABS(A(I,I)).GT.0.) GO TO 70
IF (KD.EQ.1) V(1)=0.
JC(1)=I-1
RETURN
70
GO TO (71,72),KD
71
IF (A(I,I).LT.0.) S=-S
V(1)=V(1)+ALOG(ABS(A(I,I)))
C
JLS111069
72
X=A(I,I)
A(I,I)=1.
DO 80 J=M,L
A(I,J)=A(I,J)/X
CALL OVERFL(IFL)
IF (IFL.EQ.1) GO TO 150
80
CONTINUE
DO 91 K=1,N
IF (K.EQ.I) GO TO 91
X=A(K,I)
A(K,I)=0.
DO 90 J=M,L
A(K,J)=A(K,J)-X*A(I,J)
CALL OVERFL(IFL)
IF (IFL.EQ.1) GO TO 150
90
CONTINUE
91
CONTINUE
GO TO (95,140),KI
95
DO 130 J=1,N
IF (JC(J).EQ.J) GO TO 130
JJ=J+1
DO 100 I=JJ,N
IF (JC(I).EQ.J) GO TO 110
100
CONTINUE
110
JC(I)=JC(J)
DO 120 K=1,N
X=A(K,I)
A(K,I)=A(K,J)
120
A(K,J)=X
130
CONTINUE
140
JC(1)=N
IF (KD.EQ.1) V(2)=S
RETURN
150
JC(1)=I-1
IF (KD.EQ.1) V(2)=S
```

GJR00010  
GJR00020  
GJR00030  
GJR00040  
GJR00050  
GJR00060  
GJR00070  
GJR00080  
GJR00090  
GJR00100  
GJR00110  
GJR00120  
GJR00130  
GJR00140  
GJR00150  
GJR00160  
GJR00170  
GJR00180  
GJR00190  
GJR00200  
GJR00210  
GJR00220  
GJR00230  
GJR00240  
GJR00250  
GJR00260  
GJR00270  
GJR00280  
GJR00290  
GJR00300  
GJR00310  
GJR00320  
GJR00330  
GJR00340  
GJR00350  
GJR00360  
GJR00370  
GJR00380  
GJR00390  
GJR00400  
GJR00410  
GJR00420  
GJR00430  
GJR00440  
GJR00450  
GJR00460  
GJR00470  
GJR00480  
GJR00490  
GJR00500  
GJR00510  
GJR00520  
GJR00530  
GJR00540  
GJR00550  
GJR00560  
GJR00570  
GJR00580  
GJR00590  
GJR00600  
GJR00610  
GJR00620  
GJR00630  
GJR00640  
GJR00650  
GJR00660  
GJR00670  
GJR00680  
GJR00690  
GJR00700  
GJR00710  
GJR00720  
GJR00730  
GJR00740  
GJR00750  
GJR00760  
GJR00770  
GJR00780  
GJR00790

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: GJR

RETURN 1  
END

GJR00800  
GJR00810

~~11-24~~

160



FILE: MAPHDG

```

SUBROUTINE MAPHDG(NOCAT,CLSSYM,CATNAM,KATNO,CLSMTX,SURNO,SURDES) MAP00010
                                                                MAP00020
                                                                MAP00030
                                                                MAP00040
                                                                MAP00050
                                                                MAP00060
                                                                MAP00070
                                                                MAP00080
                                                                MAP00090
                                                                MAP00100
                                                                MAP00110
                                                                MAP00120
                                                                MAP00130
                                                                MAP00140
                                                                MAP00150
                                                                MAP00160
                                                                MAP00170
                                                                MAP00180
                                                                MAP00190
                                                                MAP00200
                                                                MAP00210
                                                                MAP00220
                                                                MAP00230
                                                                MAP00240
                                                                MAP00250
                                                                MAP00260
                                                                MAP00270
                                                                MAP00280
                                                                MAP00290
                                                                MAP00300
                                                                MAP00310
                                                                MAP00320
                                                                MAP00330
                                                                MAP00340
                                                                MAP00350
                                                                MAP00360
                                                                MAP00370
                                                                MAP00380
                                                                MAP00390
                                                                MAP00400
                                                                MAP00410
                                                                MAP00420
                                                                MAP00430
                                                                MAP00440
                                                                MAP00450
                                                                MAP00460
                                                                MAP00470
                                                                MAP00480
                                                                MAP00490
                                                                MAP00500
                                                                MAP00510
                                                                MAP00520
                                                                MAP00530
                                                                MAP00540
                                                                MAP00550
                                                                MAP00560
                                                                MAP00570
                                                                MAP00580
                                                                MAP00590
                                                                MAP00600
                                                                MAP00610
                                                                MAP00620
                                                                MAP00630
                                                                MAP00640
                                                                MAP00650
                                                                MAP00660
                                                                MAP00670
                                                                MAP00680
                                                                MAP00690
                                                                MAP00700
                                                                MAP00710
                                                                MAP00720
                                                                MAP00730
                                                                MAP00740
                                                                MAP00750

THIS ROUTINE PRINTS THE HEADER INFORMATION FOR THE CLASSIFICATION
MAP IN CLASSIFY AND DISPLAY

NOCAT -- NO. OF CATEGORIES
CLSSYM -- SYMBOLS FOR CATEGORIES OR SURCLASSES
CATNAM -- CATEGORY NAMES
KATNO -- CATEGORY EACH CLASS WAS ASSIGNED TO
CLSMTX -- CLASS NAMES
SURNO -- NO. OF SURCLASSES IN EACH CLASS
SURDES -- SURCLASS NAMES
CLSV2 -- CLASS EACH SUBCLASS WAS ASSIGNED TO (IN COMMON
BLOCK INFORM)

IMPLICIT INTEGER (A-Z)

INCLUDE COMCHK1.LIST
COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
* AVAR2,CVAR2,CLSID2,SURNO2,SURDS2,FLDSV2,VERTX2,
* FEVC2(30),SUHVC2(75),SUBPTH(75),CLSV2(60),
* KEPTS(60),NOGRP,GHPNAM(60),GRPDIX(61),
* GRPCHK(61),GROUPS(124)
CSEND
C
LOGICAL ISWTH
DIMENSION CLSSYM(1),CATNAM(1),KATNO(1),CLSMTX(1),SUBNO(1),
* SURDES(1)
C
PRINTS CATEGORY CLASSIFIER INFORMATION

IF (NOCAT .EQ. 0) GO TO 82
WRITE(6,200)
200 FORMAT(//T42,'MAP OF CATEGORY CLASSIFIER CLASSIFICATION RESULTS',
* ///T32,'CATEGORY',T62,'CLASS',T93,'SUBCLASS',T131,
* 'NO.',T17,'NAME',T60,'NO.',T66,'NAME',
* TRR,'NO.',T94,'NAME',T101,'SYMBOL')
C
DO 68 I=1,NOCAT
C
WRITE(6,210)I,CATNAM(I)
210 FORMAT(/T31,I2,T37,A4)
ISWTH = .TRUE.
DO 63 J=1,NOCLS2
IF (KATNO(J) .EQ. I) GO TO 64
GO TO 63
64 IF (ISWTH) GO TO 65
WRITE(6,220)J,CLSMTX(J)
220 FORMAT(/T40,I2,T66,A4)
ISWTH = .TRUE.
GO TO 66
65 WRITE(6,230)J,CLSMTX(J)
230 FORMAT(1H+,T40,I2,T66,A4)
DO 67 K=J,NOSUR2
IF (CLSV2(K) .EQ. J) GO TO 70
GO TO 67
70 NSURCL = SURNO(J)
KK = 0
DO 75 L=1,NSUBCL
KK = K + L - 1
IF (ISWTH) GO TO 72
WRITE(6,250)KK,SURDES(KK),CLSSYM(KK)
250 FORMAT(1H8,I2,T94,A4,T103,A1)
GO TO 75
72 WRITE(6,240)KK,SURDES(KK),CLSSYM(KK)
240 FORMAT(1H+,T40,I2,T94,A4,T103,A1)
ISWTH = .FALSE.
75 CONTINUE
GO TO 63
67 CONTINUE
63 CONTINUE
6A CONTINUE
RETURN

PRINTS STANDARD CLASSIFIER INFORMATION
```

FILE: MAPHDG

```
A2 CONTINUE MAP00760
C WRITE(4,260) MAP00770
260 FORMAT(// T42, 'MAP OF STANDARD CLASSIFIER CLASSIFICATION RESULT' MAP00780
  *S1 //T45, 'CLASS', T77, 'SUBCLASS'/ T42, 'NO.', T50, 'NAME', T72, 'NO.', MAP00790
  *T78, 'NAME', T85, 'SYMBOL') MAP00800
C MAP00810
  CLSNUM = 1 MAP00820
  ISWTH = .TRUE. MAP00830
  DO A9 I=1, NOSUB2 MAP00840
  IF (CLSNUM.EQ. CLSVC(I)) GO TO A5 MAP00850
  CLSNUM = CLSNUM + 1 MAP00860
  GO TO A7 MAP00870
A5 IF (ISWTH) GO TO A7 MAP00880
  WRITE(6,270) I, SURDES(I), CLSSYM(I) MAP00890
  270 FORMAT(T72, I2, T78, A4, T87, A1) MAP00900
  GO TO A9 MAP00910
A7 WRITE(6,240) CLSNUM, CLSMTX(CLSNUM), I, SURDES(I), CLSSYM(I) MAP00920
  240 FORMAT(/T42, I2, T50, A4, T72, I2, T78, A4, T87, A1) MAP00930
  ISWTH = .FALSE. MAP00940
A9 CONTINUE MAP00950
  RETURN MAP00960
  END MAP00970
  MAP00980
```

FILE: MCMLSK

SUBROUTINE MCMLSK( KK,NV,DUM,DET)

THIS ROUTINE COMPUTES THE MODIFIED CHOLESKY DECOMPOSITION OF THE COVARIANCE MATRIX. THE DECOMPOSITIONS OVERLAY THE ELEMENTS OF THE COVARIANCE MATRIX.

KK = L D L\*

KK = COVARIANCE MATRIX STORED IN SYMMETRIC STORAGE

NV = NO. OF CHANNELS

DUM = A WORK AREA OF SIZE NV-1

DET = THE DETERMINANT OF THE COVARIANCE MATRIX

REAL KK  
LOGICAL JE1  
DIMENSION KK(1), DUM(1)

DOUBLE PRECISION TF, R, R1, DUM, T1

JE1 = .TRUE.

J1 = 0

JD = 0

DET = 1.0

LOOP OVER ALL CHANNELS

DO 10 J=1,NV

KL = J-1

L = J+1

JD = J

J1 = J1 + J

TF = KK(J)

IF (JE1) GO TO 12

K1 = 0

COMPUTE THE DIAGONAL ELEMENTS OF D AND STORE IN KK

TEMPORARILY STORE THE PRODUCT KK(I,I)\*KK(J,I) IN DUM(I)

DO 15 I=1,KL

R = KK(JD + I)

K1 = K1 + I

R1 = KK(K1) \* R

TF = TF - R1 \* R

DUM(I) = K1

15

CONTINUE

12

KK(J) = TF

CONTINUE

DET = DET \* TF

IF (L .GT. NV) GO TO 10

IRD = J1 - L + 1

COMPUTE THE R, J-TH ELEMENT OF L, USING T1

DO 20 IR=1,IRD

IRD = IRD + IR - 1

T1 = KK(IRD + J)

IF (JE1) GO TO 16

DO 25 I=1,KL

T1 = T1 - DUM(I) \* KK(IRD + I)

25

CONTINUE

14

KK(IRD + J) = T1/TF

20

CONTINUE

10

JE1 = .FALSE.

CONTINUE

KK CONTAINS, IN SYMMETRIC STORAGE, THE MODIFIED CHOLESKY

FACTORIZATION OF THE INPUT MATRIX. THE LOWER TRIANGULAR MATRIX, L,

OCCUPIES THE OFF-DIAGONAL ELEMENTS OF KK, AND THE DIAGONAL

MATRIX, D, IS STORED IN THE DIAGONAL ELEMENTS IN KK.

RETURN

END

FILE: REDIF2

```
SUBROUTINE PEDIF2 (ARRAY, TOP, APRIOR, KATNO, BMATRX, PRIORI)
IMPLICIT INTEGER (A-H, O-Z)
DIMENSION ARRAY(1)
-----
CALL... CALL REDIF2 (ARRAY, TOP, APRIOR, KATNO, BMATRX, PRIORI)
ARGs... ARRAY - SEE MONTOR
        TOP - SEE MONTOR
APRIOR - APRIORI VALUES FOR EACH SUBCLASS
KATNO - CATFGOMY - CLASS CORRESPONDENCE
BMATRX - R-TRANSFORMATION, IF AVAILABLE
PRIORI - TEMPORARY STORAGE FOR A PRIORI VALUES
REQUIRES. COMMONS /INFORM/GLOBAL/CLASS/
        ROUTINES FIND12, CRDSTA, GRPSCN, FLTNUM, CATSCN
PURPOSE.. HEADS AND ANALYZES SUPER CONTROL CARDS
        FOR 'CLASSIFY'
RETURNS.. SUPERVISOR INFORMATION AND STATISTICS
-----
CONTINUE
INCLUDE COMMK1, LIST
COMMON /INFORM/ NOCLS2, NOSUR2, NOFET2, VARSZ2, TOIVT2, NOFLD2,
* AVAR2, COVAR2, CLSID2, SUBN02, SURDS2, FLDSV2, VERTX2,
* FETVC2(30), SUBVC2(75), SURPTR(75), CLSVC2(60),
* KEPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),
* GRPCHK(61), GROUPS(124)
COMMON /CLASS/ APRFLG, BMCOMB, BMFEAT, BMFLG, NOCAT, TH1J1, IDATA1,
* NFILE, STATKY, CATNAM(60),
3 CLSSYM(60), CON(60), DET(60), FLDESC, FLDINF(6),
4 KCLSNA(60), NOCTCL(60), SUBCAT(60)
* NOCHAN, CHNVEC(30)
COMMON /GLOBAL/ HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
* HISFIL, HISKEY, TRFORM, ERIPTP, ERPKEY, MAPUNT, NOFILE,
* DRUMAD, DRMSDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL
* NHSTUN, NHSTFI, SCTRUN, MAPFIL
* DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
* CRDUNT, PRTUNT, RANDIO
C$END
EQUIVALENCE (HEAD(1), HEAD(4)), (DATE(1), HEAD(22)),
1 (HEAD(2), HEAD(30)), (COMENT(1), HEAD(48))
-----
INCLUDE COMMK6, LIST
-----
DIMENSION CODTAR(16), KATNO(60)
1 , HEAD1(10), HEAD2(10), DATE(2), COMENT(10), BMCOF(3)
2 , APRIOR(1), CARO2(62), COMVEC(2)
DIMENSION EQUVEC(2), ACARD(20)
```

FILE: REDIF2

```
C
C
C      INCLUDE COMRK2,LIST
C      DATA CODTAB/'SURC','CHAN','DATE','*END','COMM',
1  'HED1','HED2','OPTI','MODU','GROU','B-MA',
2  'APRI','CATE','DATA','STAT','MAPT'/
C
C      DATA BLANK/' ' //, NBCD/'N' //, OBCD/'O' //, SBCD/'S' //, CBCD/'C' //
1  ' COMVEC/1, ' ' /, HMCDF/2, 'C' , 'F' /
C      DATA IHCD/'I' //
C      DIMENSION SLASH(2)
C      DATA SLASH/1, ' ' //
C
C      DATA FNDCHD/'*END' //, EQUVEC/1, ' ' //
C
C      DATA DRCD/'D' //, FB CD/'F' //, TESTB/'ILE ' //, UBCD/'U' //
C      REAL RMATRX(1), APRIOR, PRIORI(60)
C      LOGICAL *1 LCHAR(4), LLCHAR(4)
C      DIMENSION ICHAR(1), IICHAR(1)
C      DATA BLANKS/' ' //
C      EQUIVALENCE (LCHAR(1), ICHAR(1)), (LLCHAR(1), IICHAR(1))
C
C-----
C-----
C
C      INI7
C-----
C-----
C
C      SYMMAX = 60
C      NF=0
C      NOSUR2=0
C      L = 0
C      NOCAT = 0
C      NOCLS2 = 0
C      DATSWT = 0
C      NOGRP = 0
C      GRPTR = 0
C      RMFLG = 0
C      APRFLG = 0
C      APRKEY = 0
C      NOCHAN = 0
C      DO 10 I=1,60
C      CATNAM(I) = BLANK
10  NOCTCL(I) = 0
C      DO 1 I=1,SYMMAX
C      APRIOR(I) = 0.0
1  GRPCHK(I) = 0
C
C      READ SUPERVISOR CONTROL CARDS
C-----
C-----
C
C      225 CONTINUE
C
C      PUT THE NEXT CARD IN THE REREAD BUFF-ER->RQ+X)
C
C      READ(21,230) (ACARD(I),I=1,20)
230  FORMAT(20A4)
C      WRITE(30,230) (ACARD(I),I=1,20)
C      PEWIND 30
C      READ(30,2204) CODE,CARD2
C      PEWIND 30
2204  FORMAT(44,6X,62A1)
C      COL = 0
C      WRITE(6,2206) CODE,CARD2
2206  FORMAT(15,44,6X,62A1)
C      DO 237 I = 1,14
235  IF ( CODE .EQ. CODTAB(I) )
1  GO TO (240,250,260,270,500,510,520,530,540,560,580,590,600,610,
* 620,630),I
237  CONTINUE
C      GO TO 1000
C
C      SURCLASS
C-----
```

FILE: REDIF2

```
C
240 M = NXTCHR(CARD2,COL)
    IF ( M.EQ. BLANK ) GO TO 225
    COL = COL - 1
    NOSUB2 = NUMBER(CARD2,COL,SUBVC2,NOSUB2)
    CALL ORDER(SUBVC2,NOSUB2)
C
    GO TO 225
C
    CHANNELS
    -----
C
250 IF ( RMFLG.GT. 0) GO TO 225
    M = NXTCHR( CARD2, COL )
    IF ( M.EQ. BLANK ) GO TO 225
    J = FIND12(CARD2,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 251
    IF (M.EQ. SPCD) GO TO 253
    IF (M.EQ. D4CD) GO TO 255
251 WRITE(6,252)
252 FORMAT(' ERROR ON CHANNELS CARD*')
    GO TO 225
253 NF = NUMBER(CARD2,COL,FETVC2,NF)
    NOFET2 = NF
    CALL ORDER(FETVC2,NOFET2)
    COL = COL - 1
    GO TO 250
255 NOCHAN = NUMBER(CARD2,COL,CHNVEC,NOCHAN)
    CALL OPDER(CHNVEC,NOCHAN)
    COL = COL - 1
    GO TO 250
C
    DATE
    -----
C
240 M = NXTCHR(CARD2,COL)
    IF ( M.EQ. BLANK ) GO TO 225
    READ (30,9999) DATE
    REWIND 30
9999 FORMAT(10X,11A4)
    GO TO 225
C
    COMMENT
    -----
C
500 CONTINUE
    READ (30,9999) COMENT
    REWIND 30
    GOTO 225
C
    HED1
    -----
C
510 CONTINUE
    READ (30,9999) HED1
    REWIND 30
    GOTO 225
C
    HED2
    -----
C
520 CONTINUE
    READ (30,9999) HED2
    REWIND 30
    GOTO 225
C
    OPTION CARD
    -----
C
530 M = NXTCHR(CARD2,COL)
    IF ( M.EQ. BLANK ) GO TO 225
    IF ( M.NE. SRCD ) GO TO 1000
    STATK = 1
    M = FIND12( CARD2,COL,COMVEC)
    IF ( M.LE. 0) GO TO 225
    GO TO 530
C
    MODULE
    -----
```

RED01590  
RED01600  
RED01610  
RED01620  
RED01630  
RED01640  
RED01650  
RED01660  
RED01670  
RED01680  
RED01690  
RED01700  
RED01710  
RED01720  
RED01730  
RED01740  
RED01750  
RED01760  
RED01770  
RED01780  
RED01790  
RED01800  
RED01810  
RED01820  
RED01830  
RED01840  
RED01850  
RED01860  
RED01870  
RED01880  
RED01890  
RED01900  
RED01910  
RED01920  
RED01930  
RED01940  
RED01950  
RED01960  
RED01970  
RED01980  
RED01990  
RED02000  
RED02010  
RED02020  
RED02030  
RED02040  
RED02050  
RED02060  
RED02070  
RED02080  
RED02090  
RED02100  
RED02110  
RED02120  
RED02130  
RED02140  
RED02150  
RED02160  
RED02170  
RED02180  
RED02190  
RED02200  
RED02210  
RED02220  
RED02230  
RED02240  
RED02250  
RED02260  
RED02270  
RED02280  
RED02290  
RED02300  
RED02310  
RED02320  
RED02330  
RED02340  
RED02350  
RED02360  
RED02370



FILE: REDIF2

```
5872 FORMAT(/// 5X,***** TERMINATING PROGRAM EXECUTION FROM REDIF2 **RED03170
      1***I/IH1) RED03180
      CALL EXIT RED03190
C     APRIORI CARD --- TRAINING CLASS APRIORI PROBABILITY VALUES RED03200
C     590 M = NXTCHR(CARD2,COL) RED03210
      IF(M .EQ. FRCD)GO TO 592 RED03220
      APRKEY = -777777 RED03230
C     FLAG FOR COMPUTING APRIOR VALUES FROM STATFILE RED03240
      GO TO 596 RED03250
C     592 M=SYMMAX - APRKEY RED03260
      COL = 0 RED03270
      NAPR = APRKEY + 1 RED03280
      APRKEY = APRKEY + FLINUM(CARD2,COL,PRIORI(NAPR),M) RED03290
      IF (APRKEY.NE.1) GO TO 596 RED03300
      WRITE(6,594) RED03310
C     594 FORMAT(/T5,*** CLSFY/REDIF2 - BAD CARD INPUT ON APRIORI CARD - DERED03340
      *FAULT APRIORI PROBABILITY VALUES WILL BE USED.*/) RED03350
      APRKEY = 0 RED03360
C     596 APRFLG = APRKEY RED03370
C     GO TO 225 RED03380
C     CATEGORY CARD RED03390
C     ----- RED03400
C     600 IF(NOCAT.EQ.-7654321)GO TO 225 RED03410
      LL = NXTCHR(CARD2,COL) RED03420
      IF (LL.NE.FPCD) GO TO 605 RED03430
      LL = NXTCHR(CARD2,COL) RED03440
      IF (LL.NE.IBCD) GO TO 605 RED03450
      NOCAT = -7654321 RED03460
      GO TO 225 RED03470
C     605 NOCAT = NOCAT + 1 RED03480
      L = CATSNC(CARD2,KCLSNA,CATNAM(NOCAT),L,NOCTCL(NOCAT),NOCAT) RED03490
      GO TO 225 RED03500
C     DATA FILE CARD RED03510
C     610 M = NXTCHR(CARD2,COL) RED03520
      IF(M .EQ. BLANK)GO TO 225 RED03530
      IF(M .EQ. URCD)GO TO 616 RED03540
      IF(M .EQ. FRCD) GO TO 617 RED03550
C     613 WRITE(6,753) RED03560
C     753 FORMAT(' ERROR ON DATA FILE CARD') RED03570
      GO TO 225 RED03580
C     616 J = FIND12(CARD2,COL,EQUVEC) RED03590
      IF (J .EQ. -1) GO TO 613 RED03600
      M = NUMBER(CARD2,COL,DATAPE,ZERO) RED03610
      COL = COL - 1 RED03620
      GO TO 610 RED03630
C     617 J = FIND12(CARD2,COL,EQUVEC) RED03640
      IF (J .EQ. -1) GO TO 613 RED03650
      M = NUMBER(CARD2,COL,DATAFIL,ZERO) RED03660
      DATAFIL = DATAFIL - 1 RED03670
      IF (DATAFIL .LT. 0) DATAFIL = 0 RED03680
      COL = COL - 1 RED03690
      GO TO 610 RED03700
C     STAT FILE CARD RED03710
C     620 M = NXTCHR(CARD2,COL) RED03720
      IF(M .EQ. BLANK)GO TO 225 RED03730
      IF(M .EQ. URCD)GO TO 625 RED03740
      IF(M .EQ. FRCD)GO TO 627 RED03750
C     623 WRITE(6,755) RED03760
C     755 FORMAT(' ERROR ON STAT FILE CARD') RED03770
      GO TO 225 RED03780
C     625 J = FIND12(CARD2,COL,EQUVEC) RED03790
      IF (J .EQ. -1) GO TO 623 RED03800
      M = NUMBER(CARD2,COL,SAVTAP,ZERO) RED03810
      COL = COL - 1 RED03820
      GO TO 620 RED03830
C     627 J = FIND12(CARD2,COL,EQUVEC) RED03840
      IF (J .EQ. -1) GO TO 623 RED03850
      M = NUMBER(CARD2,COL,STAFIL,ZERO) RED03860
      STAFIL = STAFIL - 1 RED03870
      RED03880
      RED03890
      RED03900
      RED03910
      RED03920
      RED03930
      RED03940
      RED03950
```



FILE: REDIF2

IF (STAFIL .LT. 0) STAFIL = 0  
COL = COL - 1  
GO TO 620

CCCCC

-----

```

630 CONTINUE
    ERPKEY = 1
    J = NXTCHR(CARD2, COL)
631 IF (J.EQ.BLANK) GO TO 650
    IF (J.NE.URCD) GO TO 635
    J = FIND12(CARD2, COL, EQUVEC)
    IF (J.NE.2) GO TO 650
    ISTART = 0
    J = NUMBER(CARD2, COL, MAPTAP, ISTART)
    J = FIND12(CARD2, COL, EQUVEC)
    IF (J.NE.2) GO TO 650
    ISTART = 0
    J = NUMBER(CARD2, COL, ERPKEY, ISTART)
    GO TO 225
635 IF (J.NE.URCD) GO TO 640
    J = FIND12(CARD2, COL, SLASH)
    IF (J.NE.2) GO TO 650
    J = NXTCHR(CARD2, COL)
    IF (J.EQ.FRCD) GO TO 645
    IF (J.EQ.URCD) GO TO 631
    GO TO 650
640 IF (J.NE.FRCD) GO TO 650
645 J = FIND12(CARD2, COL, EQUVEC)
    IF (J.NE.2) GO TO 650
    ISTART = 0
    J = NUMBER(CARD2, COL, ERPKEY, ISTART)
    J = FIND12(CARD2, COL, EQUVEC)
    IF (J.NE.2) GO TO 650
    ISTART = 0
    J = NUMBER(CARD2, COL, MAPTAP, ISTART)
    GO TO 225
650 WRITE (6, 655)
655 FORMAT(' ERROR ON MAPTAP CONTROL CARD')
    GO TO 225
270 CONTINUE
290 CONTINUE
    IF (NOCAT.EQ.1) WRITE(6, 615)
    IF (NOCAT.EQ.1) CALL CMERR
615 FORMAT('/// 5X, *MUST HAVE AT LEAST TWO CATEGORIES')
    RETURN

```

CCCC

ERROR ROUTINES

```

-----
1000 WRITE (6, 10002) CODE, CARD2
10002 FORMAT('/// 5X, ***** CLSFY/REDIF2 --- BAD PROCESSOR CONTROL CAR
IN .....//5X, 2H'//A4, 6X, 62A1, 2H'//5X, 1***** TERMINATING PROGRAM EX
2ECUTION FROM REDIF2 *****//1H1)
GO TO 225
C
END

```

RED03960  
RED03970  
RED03980  
RED03990  
RED04000  
RED04010  
RED04020  
RED04030  
RED04040  
RED04050  
RED04060  
RED04070  
RED04080  
RED04090  
RED04100  
RED04110  
RED04120  
RED04130  
RED04140  
RED04150  
RED04160  
RED04170  
RED04180  
RED04190  
RED04200  
RED04210  
RED04220  
RED04230  
RED04240  
RED04250  
RED04260  
RED04270  
RED04280  
RED04290  
RED04300  
RED04310  
RED04320  
RED04330  
RED04340  
RED04350  
RED04360  
RED04370  
RED04380  
RED04390  
RED04400  
RED04410  
RED04420  
RED04430  
RED04440  
RED04450  
RED04460  
RED04470  
RED04480  
RED04490  
RED04500  
RED04510  
RED04520  
RED04530  
RED04540  
RED04550

FILE: RELERR

```
FUNCTION RFLERR(COVMTX, COV, NOFET2, VARSZ2)
INTEGER VARSZ2
COMMON/SCHACH/SCR1(2000),SCR2(10500)
DIMENSION D(30), COV(VARSZ2), COVMTX(VARSZ2)
REAL LDL(465)
EQUIVALENCE ( ENORMO, SCR2(1)), ( ENORMO, SCR2(2)) ,
C 1 ( D(1), SCR2(3)), ( LDL(1), SCR2(33)),
C 2 ( SUM, SCR2(963)), ( II, SCR2(964)), ( I, SCR2(965)),
C 3 ( J, SCR2(966)), ( L, SCR2(967)), ( JJ, SCR2(968)),
C 4 ( KK, SCR2(969)), ( JK, SCR2(970)), ( KP, SCR2(971)),
C 5 ( III, SCR2(972)), ( IJ, SCR2(973)), ( JP, SCR2(974)),
C 6 ( IP, SCR2(975))
CCCC
COMPUTE THE EUCLIDEAN NORM OF THE COVARIANCE MATRIX, BEFORE
CHOLESKY FACTORIZATION
ENORMO = 0.0
DO 181 I=1,VARSZ2
C 181 ENORMO = ENORMO + 2.0 * COV(I) * COV(I)
II = 0
DO 182 I=1,NOFET2
II = II + I
ENORMO = ENORMO - ( COV(II) * COV(II) )
C 182 D(II) = COVMTX(II)
COVMTX(II) = 1.0
ENORMO = SQRT(ENORMO)
C
IJ = 0
DO 187 I=1,NOFET2
IK = I
II = ( IK * (IK-1) )/2
DO 186 J=1,IK
JK = J
SUM = 0.0
JJ = ( JK * (JK-1) )/2
DO 185 KP=1,JK
KK = KP
JP = II + KP
C 185 SUM = SUM + ( COVMTX(JP) * COVMTX(IP) * D(KP) )
C 186 LDL(IJ) = SUM
=L * D * L*
C 187 CONTINUE
C
II = 0
DO 188 L=1,NOFET2
II = II + L
C 188 COVMTX(II) = D(L)
ENORMD = 0.0
III = 0
II = 0
DO 190 I=1,NOFET2
II = II + I
DO 189 J=1,I
III = III + 1
SUM = COV(III) - LDL(III)
C 189 ENORMD = ENORMD + 2.0 * SUM * SUM
SUM = COV(II) - LDL(II)
C 190 ENORMD = ENORMD - ( SUM * SUM )
C
IF ( ENORMD .LE. 1.0E-8) GO TO 191
ENORMD = SQRT(ENORMD)
C
C 191 RELERR = ENORMD/ENORMO
RETURN
END
```

FILE SETUP2

```

SUBROUTINE SETUP2 (ARRAY, TOP, FLDFLG, APRIOR, BMATRIX, KATNO)
IMPLICIT INTEGER (A-H, O-Z)
REAL CON, DET, BMATRIX, APRIOR, NORM
* , APRI, APRIO, PRIORI
SET00010
SET00020
SET00030
SET00040
SET00050
SET00060
SET00070
SET00080
SET00090
SET00100
SET00110
SET00120
SET00130
SET00140
SET00150
SET00160
SET00170
SET00180
SET00190
SET00200
SET00210
SET00220
SET00230
SET00240
SET00250
SET00260
SET00270
SET00280
SET00290
SET00300
SET00310
SET00320
SET00330
SET00340
SET00350
SET00360
SET00370
SET00380
SET00390
SET00400
SET00410
SET00420
SET00430
SET00440
SET00450
SET00460
SET00470
SET00480
SET00490
SET00500
SET00510
SET00520
SET00530
SET00540
SET00550
SET00560
SET00570
SET00580
SET00590
SET00600
SET00610
SET00620
SET00630
SET00640
SET00650
SET00660
SET00670
SET00680
SET00690
SET00700
SET00710
SET00720
SET00730
SET00740
SET00750
SET00760

DIMENSION ARRAY (1), PRIORI (60), VERTCS (22)
-----
CALL.. CALL SETUP2 (ARRAY, TOP, FLDFLG, APRIOR, BMATRIX, KATNO)
ARGs.. ARRAY - SEE MONTOR
TOP - SEE MONTOR
FLDFLG -
APRIOR - APRIORI VALUES FOR EACH SUBCLASS
BMATRIX - B-TRANSFORMATION MATRIX, IF AVAILABLE
KATNO - CATEGORY - CLASS CORRESPONDENCE
REQUIRES. COMMON /INFORM/CLASS/GLOBAL/
ROUTINES FIND12, RED1F2, REDSAV
PURPOSE ANALYSIZE SUPERVISOR INFORMATION
RETURNS.. SUPERVISOR INFORMATION AND REDUCED STATISTICS
FOR PROCESSING
-----
CONTINUE
INCLUDE COMBK1, LIST
EQUIVALENCE (DATE (1), HEAD (22))
-----
INCLUDE COMRK2, LIST
-----
-----
-----
INCLUDE COMBK6, LIST
COMMON /INFORM/ NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
* AVAR2, COVAR2, CLSID2, SUBNO2, SUBDS2, FLDSV2, VERTX2,
* FETVC2 (30), SURVC2 (75), SURPTR (75), CLSVC2 (60),
* KEPPTS (60), NOGRP, GRPNAM (60), GRPDEX (61),
* GRPCHK (61), GROUPS (124)
COMMON /CLASS/ APRFLG, BMCUMR, RMFEAT, BMFLG, NOCAT, THIJ1, IDATA1,
* NFILE, STATKY, CATNAM (60),
* 3 CLSSYM (60), CON (60), DET (60), FLDESC, FLDINF (6),
* 4 KCLNSA (60), NOCTCL (60), SUBCAT (60)
* , NOCHAN, CHNVEC (30)
COMMON /GLOBAL/ HEAD (63), MAPTAP, DATAPE, SAVTAP, RMFILE, RMKEY,
* HISFIL, HISKEY, TPFORM, ERIPTP, ERPKEY, MAPUNT, NOFILE,
* DRUMAD, DRMWD, PAGES7, DATFIL, STAFIL, ASAV, ASAVFL
* , NHSTUN, NHSTFI, SCTRUN, MAPFIL
* , DOTUNT, DOTFIL, NCHPAS, TRNSFL, RMTRFL, HISTFL, PCHUNT,
* CRDUNT, PRUNT, HANDIO
CSEND
DIMENSION DATE (2), FILVEC (2), APRIOR (1), KATNO (60)
DIMENSION CARD (62), CLSSY (60)
C
DATA YRCD/'Y' /, NRCD/'N' /, MODRCD/'MODU' /, BLANK/' ' /,
1 FILVEC/ '1' /, 'F' /

```



ORIGINAL PAGE IS  
OF POOR QUALITY

FILE SETUP2

```
C      IDATA1- BASE ADDRESS FOR DATA PASSED BACK FROM TAPERD          SET01530
C      THJ1 = 1                                                         SET01540
C      IDATA1 = THJ1 * (NOSUR2-1) * (NOSUB2-2) / 2 * NOSUB2           SET01550
C      IF (NOCAT .GT. 0) IDATA1 = 1                                    SET01560
C      STORE A BLANK IN DEFAULT SYMBOLS. THIS WILL BE USED IN        SET01570
C      PRINTING THE MAP CLASSIFICATION FOR THE UNCLASSIFIED PIXEL    SET01580
C      CLASSYM(NOSUB2 + 1) = BLANK                                     SET01590
C      IF (NOCAT .LE. 0) GO TO 465                                     SET01600
C      ALL CLASSES MUST BE ASSIGNED TO A CATEGORY                    SET01610
C      DO 410 II=1,NOCAT                                              SET01620
C      NOCLSS = NOCTCL(II) * NOCLSS                                   SET01630
C      IF ( NOCLSS .EQ. NOCLS2 ) GO TO 415                             SET01640
C      WRITE(6,450) (ARRAY(CLSID2+I-1),I=1,NOCLS2)                   SET01650
C      WRITE(6,460) (KCLSNA(I),I=1,NOCLSS)                            SET01660
C      CALL CMERR                                                      SET01670
C      SET UP KATNO ARRAY TO CONTAIN THE CATEGORY EACH CLASS        SET01680
C      BELONGS TO                                                    SET01690
C      415 CONTINUE                                                  SET01700
C      NOCLAS = 0                                                     SET01710
C      DO 435 L=1,NOCAT                                               SET01720
C      NOCLAS1 = NOCLAS + 1                                           SET01730
C      NOCLAS = NOCTCL(L) * NOCLAS1 - 1                               SET01740
C      DO 425 J=NOCLAS1,NOCLAS                                         SET01750
C      CLSNAM = KCLSNA(J)                                             SET01760
C      DO 420 K=1,NOCLS2                                              SET01770
C      IF (CLSNAM .EQ. ARRAY(CLSID2-1+K)) GO TO 430                   SET01780
C      420 CONTINUE                                                  SET01790
C      440 WRITE(6,450) (ARRAY(CLSID2+I-1),I=1,NOCLS2)               SET01800
C      450 FORMAT(// 'AN ERROR HAS OCCURRED IN GROUPING CLASSES INTO CATEGORIES'//
C      * 'IES.CHECK THE FOLLOWING :// 5X,'1. NOT ALL OF THE CLASSES HAVE BEEN'//
C      * 'EN ASSIGNED TO A CATEGORY.// 5X,'2. A CLASS NAME ON THE CATEGORY'//
C      * 'CARD HAS BEEN MISPELLED.// 10X, 'CLASS NAMES FROM SAVTAP FILE ARSET01900
C      * 'E : ' / 5X, (10(A6,2X)))                                     SET01910
C      WRITE(6,460) (KCLSNA(I),I=1,NOCLSS)                             SET01920
C      460 FORMAT( / 10X, 'CLASS NAMES FROM CATEGORY CARDS ARE : // 5X,
C      * (10(A4,4X)))                                                 SET01930
C      CALL CMERR                                                      SET01940
C      430 KATNO(K) = L                                               SET01950
C      425 CONTINUE                                                  SET01960
C      435 CONTINUE                                                  SET01970
C      SET UP SUBCAT ARRAY TO CONTAIN THE CATEGORY EACH SUBCLASS    SET01980
C      BELONGS TO                                                    SET01990
C      DO 437 II = 1,NOSUR2                                           SET02000
C      CLSNUM = CLSVC2(II)                                             SET02010
C      437 SUBCAT(II) = KATNO(CLSNUM)                                  SET02020
C      465 CONTINUE                                                  SET02030
C      PRINT OUT THE SUPERVISOR INFORMATION                           SET02040
C      -----                                                       SET02050
C      WRITE(6,HEAD)                                                  SET02060
C      WRITE(6,5012)                                                  SET02070
C      IF ( STATKY .EQ. 1 ) WRITE(6,5014)                             SET02080
C      IF ( NOCAT .GT. 0 ) WRITE(6,5018)                             SET02090
C      IF ( NOCAT .LE. 0 ) WRITE(6,5020)                             SET02100
C      IF (FILEOP.EQ.1) WRITE(6,5022)                                 SET02110
C      5012 FORMAT(12,'THE FOLLOWING OPTIONS HAVE BEEN SELECTED'//)   SET02120
C      5014 FORMAT(15,'PRINT MULTISPECTRAL STATISTICS.')
```



ORIGINAL PAGE IS  
OF POOR QUALITY

FILE SETUP2

```
C      COMPUTE DEFAULT APRIORI VALUE FOR CATEGORY CLASSIFIER
C      CLSNM = 0
      APR1 = 1.0 / FLOAT(NOCAT)
      DO 750 I=1,NOCAT
      II = NOCTCL(I)
      M = 1
      NOSACL = 0
      DO 715 K=1,II
      DO 705 KK=1,NOCLS2
      IF (KCLSNA(K*CLSNM) .EQ. ARRAY(CLSID2-1*KK)) GO TO 715
705 CONTINUE
715 NOSACL = ARRAY(SURN02-1*KK) * NOSBCL
      APR10 = APR1 * 1.0 / FLOAT(NOSBCL)
      DO 720 KKK=1,NOSUR2
      IF (I .EQ. SURCAT(KKK) ) APRIOR(KKK) = APR10
      IF (I .EQ. SUBCAT(KKK) ) M = M + 1
      IF (M .GT. NOSBCL) GO TO 740
720 CONTINUE
740 CLSNM = CLSNM + II
750 CONTINUE
      GO TO 6053

C      COMPUTE APRIORI VALUES FROM STATFILE
C      760 TKEPTS = 0
      DO 765 I=1,NOSUB2
765 TKEPTS = TKEPTS + KEPPTS(I)
C      TOTAL ALL SUBCLASS PIXELS
      DO 770 J=1,NOSUB2
770 APRIOR(I) = FLOAT(KEPPTS(I))/FLOAT(TKEPTS)
      IPAT = STAFIL * 1
      WRITE (6,775) IPAT
775 FORMAT(15,'APRIORI VALUES FROM STATFILE',I3,' APRIORI= NO. PI
      *XELS IN SURCLASS/TOTAL NO. PIXELS IN ALL SUBCLASSES ***')
      GO TO 6070

C      COMPUTE DEFAULT APRIOR VALUES FOR STANDARD CLASSIFIER
C      605 NORM = 1.0 / FLOAT(NOSUB2)
      DO 606 I=1,NOSUB2
606 APRIOR(I) = NORM
C      WRITE(6,6050)
C      6050 FORMAT(15,'DEFAULT APRIORI PROBABILITY VALUES WILL BE USED. SURCL
      *ASS(I) = 1.0/(NO. OF SURCLASSES)')
      GO TO 6070
6053 CONTINUE
      WRITE(6,6055)
6055 FORMAT(15,'DEFAULT APRIORI PROBABILITY VALUES FOR SUBCLASS(I) = 1.
      *0/(NO OF CATEGORIES)*(NO. OF SUBCLASSES IN CATEGORY(J))')
C      6070 APRFLG = NOSUB2
      WRITE(6,502)NFILE,NOFLD2,NOCLS2,NOSUB2,NOFET2
607 IF (BMFLG .LE. 0) GO TO 700
C      CALL WRBMB(BMATRX,BMCOMB,BMFEAT,FETVC2)
C      700 CONTINUE
C      IF (NOCHAN .NE. NOFET2) WRITE(6,800)
800 FORMAT(15,'NO. OF CHANNELS REQUESTED FOR DATA TAPE AND NO. OF CHANNE
      *LS ON STAT//: FILE MUST BE EQUAL')
      IF (NOCHAN .NE. NOFET2) CALL CMERR
      -----
      WRITE FIRST RECORD ON THE CLASSIFICATION RESULTS OUTPUT FILE.
      MAPTAP
      -----
      HEADER RECORD NO. 1 FOR MAPTAP
```

SET03050  
SET03060  
SET03070  
SET03080  
SET03090  
SET03100  
SET03110  
SET03120  
SET03130  
SET03140  
SET03150  
SET03160  
SET03170  
SET03180  
SET03190  
SET03200  
SET03210  
SET03220  
SET03230  
SET03240  
SET03250  
SET03260  
SET03270  
SET03280  
SET03290  
SET03300  
SET03310  
SET03320  
SET03330  
SET03340  
SET03350  
SET03360  
SET03370  
SET03380  
SET03390  
SET03400  
SET03410  
SET03420  
SET03430  
SET03440  
SET03450  
SET03460  
SET03470  
SET03480  
SET03490  
SET03500  
SET03510  
SET03520  
SET03530  
SET03540  
SET03550  
SET03560  
SET03570  
SET03580  
SET03590  
SET03600  
SET03610  
SET03620  
SET03630  
SET03640  
SET03650  
SET03660  
SET03670  
SET03680  
SET03690  
SET03700  
SET03710  
SET03720  
SET03730  
SET03740  
SET03750  
SET03760  
SET03770  
SET03780  
SET03790  
SET03800







FILE: THRESH

SUBROUTINE THRESH(NOCLS2,NOFET2,NPL1, APRIOR,AVEMTX,COVMTX,DET,  
1 VARSZ2,S1,S2,U1,U2,88,DIAG,THIJ)

SUBROUTINE THRESH COMPUTES THE CLASS-PAIR THRESHOLDS , AND RETURNS  
THEM IN 'SYMMETRIC' STORAGE ( THIJ) IN THE FOLLOWING MANNER:

2.1					
3.1	3.2				
4.1	4.2	4.3			
5.1	5.2	5.3	5.4		
6.1	6.2	6.3	6.4	6.5	

THRESH REQUIRES THE FUNCTION SUBPROGRAM, G ( WITH ALTERNATE ENTRY  
POINT, GG ), AND THE SUBROUTINE FALSY .

-----  
COMPUTATION OF THE CLASS-PAIR THRESHOLDS  
-----

FOR U1 = MEAN VECTOR, CLASS(1) OF THE CLASS-PAIR  
U2 = " " " " CLASS(2) " " " "  
COV1 = " " COVARIANCE MATRIX, CLASS(1) OF THE CLASS-PAIR  
COV2 = " " " " " " CLASS(2) " " " "  
C1 = 2 \* LOG APRIORI(CLASS 1) - LOG DETERMINANT(CLASS 1)  
C2 = 2 \* LOG APRIORI(CLASS 2) - LOG DETERMINANT(CLASS 2)  
AND

C = C2 - C1

- (1) IF C1 .LE. C2 - (U1-U2)\* X COV2\*\* -1 X (U1-U2), THIJ= C1
- (2) IF C2 .LE. C1 - (U1-U2)\* X COV1\*\* -1 X (U1-U2), THIJ= C2
- (3) IF NEITHER (1) NOR (2), COMPUTE THE CLASS-PAIR THRESHOLD  
ITERATIVELY AS FOLLOWS:

FIND A NUMBER, 0 .GE. X .LE. 1 , SO THAT THE SOLUTION VECTOR,  
H(X), OF THE SYSTEM OF EQUATIONS,

(3A) 
$$\begin{aligned} & ((1-X) * COV1** -1 + X * COV2** -1) * H(X) \\ & = (1-X) * COV1** -1 * U1 + X * COV2** -1 * U2 \end{aligned}$$

ALSO SATISFIES  $G( H(X) ) = C2 - C1$  , WHERE

(3B) 
$$G( H(X) ) = ( H(X) - U2) * * COV2** -1 * ( H(X) - U2) \\ - ( H(X) - U1) * * COV1** -1 * ( H(X) - U1 )$$

SUBROUTINE FALSY DETERMINES 3-POINT INTERVALS IN THE RANGE,  
0 .GE. X .LE. 1 , FITS A QUADRATIC Q(X) TO THE THREE POINTS AND  
OBTAINS THE ROOT, X , FOR Q(X) = C2 - C1. THE ROOT, X , OF THIS  
QUADRATIC APPROXIMATION OF G( H(X) ) IS SENT TO FUNCTION G.

FUNCTION G OBTAINS THE SOLUTION VECTOR , H(X), FOR THE SYSTEM  
OF EQUATIONS (3A) , AND APPLIES THE H(X) TO OBTAIN THE VALUE  
OF G( H(X) ) (EQN. 3B). FALSY TESTS THE VALUE OF G( H(X) )  
IN RELATION TO C2 - C1 , WITHIN A PRE-SET TOLERANCE ON THE RESIDUAL.  
IF G( H(X) ) .NE. C2 - C1 + TOLERANCE, ITERATION CONTINUES IN  
FALSY WITH NEW INTERVALS. QUADRATIC FIT OF THE INTERVALS. ROOT  
OF THE APPROXIMATING QUADRATIC SENT TO FUNCTION G. FALSY  
RETURNS TO THRESH WHEN A ROOT, X , OF THE APPROXIMATING QUADRATIC  
IS FOUND WHICH YIELDS AN H(X) THAT SATISFIES  $G( H(X) ) = C2 - C1$   
WITHIN THE PRE-SET TOLERANCE.

THRESH COMPUTES THE CLASS-PAIR THRESHOLD, THIJ, BY OBTAINING FROM  
G THE EVALUATION OF G( H(X) ) FOR THE X RETURNED BY FALSY :

$$THIJ = .5 * ( C1 - ( H(X) - U1) * * COV1** -1 * ( H(X) - U1) )$$

INTEGER VARSZ2  
-----

THR00010  
THR00020  
THR00030  
THR00040  
THR00050  
THR00060  
THR00070  
THR00080  
THR00090  
THR00100  
THR00110  
THR00120  
THR00130  
THR00140  
THR00150  
THR00160  
THR00170  
THR00180  
THR00190  
THR00200  
THR00210  
THR00220  
THR00230  
THR00240  
THR00250  
THR00260  
THR00270  
THR00280  
THR00290  
THR00300  
THR00310  
THR00320  
THR00330  
THR00340  
THR00350  
THR00360  
THR00370  
THR00380  
THR00390  
THR00400  
THR00410  
THR00420  
THR00430  
THR00440  
THR00450  
THR00460  
THR00470  
THR00480  
THR00490  
THR00500  
THR00510  
THR00520  
THR00530  
THR00540  
THR00550  
THR00560  
THR00570  
THR00580  
THR00590  
THR00600  
THR00610  
THR00620  
THR00630  
THR00640  
THR00650  
THR00660  
THR00670  
THR00680  
THR00690  
THR00700  
THR00710  
THR00720  
THR00730  
THR00740  
THR00750  
THR00760  
THR00770  
THR00780  
THR00790



12. DISPLAY PROCESSOR

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE DSPLAY

```

SUBROUTINE DSPLAY (ARRAY, TOP)                                DSP00010
IMPLICIT INTEGER (A-H, O-Z)                                  DSP00020
                                                                DSP00030
                                                                DSP00040
                                                                DSP00050
                                                                DSP00060
                                                                DSP00070
                                                                DSP00080
                                                                DSP00090
----- DSP00100
----- DSP00110
CALL.. CALL DSPLAY (ARRAY, TOP)                               DSP00120
                                                                DSP00130
ARGS.. ARRAY = SFE 'MONTOR'                                   DSP00140
      TOP = SFE 'MONTOR'                                       DSP00150
                                                                DSP00160
PURPOSE.. COORDINATES ROUTINES FOR DISPLAYING CLASSIFICATION DSP00170
          MAP AND PERFORMANCE TABLES.                         DSP00180
----- DSP00190
----- DSP00200
                                                                DSP00210
                                                                DSP00220
                                                                DSP00230
                                                                DSP00240
INCLUDE CMK10, LIST                                           DSP00250
INCLUDE COM10                                                 DSP00260
COMMON/DISPL/ CATFLG, CATNAM (61), CLSNAM (61), SUBNAM (61), SUBNO (60),
*   SURCAT (60), CLSSUB (60), NOMAP, TOTVT3, NOSUB3,          DSP00270
*   PCFDKY, TSTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,  DSP00280
*   PLTKY, BMFLG, BMCOMB, BMFEAT, CDATE (2),                 DSP00290
*   FLDSV2, FIELD2, VERTX2, FLDSV3, FIELD3, VERTX3, PCTID3,  DSP00300
*   THRS (60), SYMPTX (66), HIGH (60), CON (60)              DSP00310
*   , FLTKY, NOFLD2, NOFLD3, NOFET2, FETVC2 (30)             DSP00320
*   , NOSUP2, NOTRFD, TOTVT2, NOCLS2                         DSP00330
*   , KATNO (60), NOCAT, FILTER, MAPFMT                      DSP00340
*   , DESKEY, DESUNI, DESOTH, CROP, ACROP, AOTHER, ATOTAL   DSP00350
*   , SITE (6), ANALYS (5), CAM (15), CRPKY, KEPPTS (60)     DSP00360
*   , DOTKEY, DOTERR                                         DSP00370
                                                                DSP00380
COMMON BLOCK DISPL IS USED ONLY IN THE DISPLAY PROCESSOR    DSP00390
                                                                DSP00400
                                                                DSP00410
                                                                DSP00420
DEFINITIONS                                                    DSP00430
CATFLG - FLAG INDICATING WHETHER OR NOT CATEGORY PERFORMANCE DSP00440
          REPORTS MUST BE GENERATED.                         DSP00450
CATNAM - NAMES OF CATEGORIES. READ FROM MAPTAP.              DSP00460
CLSNAM - NAMES OF CLASSES. READ FROM MAPTAP.                DSP00470
SUBNAM - NAMES OF SUBCLASSES. READ FROM MAPTAP.            DSP00480
SURCAT - SURCLASS-CATEGORY CORRESPONDENCE VECTOR           DSP00490
          (SURCAT(I)=M MEANS SUBCLASS I BELONGS TO CATEGORY M)
CLSSUB - SURCLASS-CLASS CORRESPONDENCE VECTOR.              DSP00500
          (CLSSUB(I)=M MEANS SUBCLASS I BELONGS TO CLASS M)
NOMAP - TRIGGER INDICATING WHETHER OR NOT A MAP IS TO BE PRINTED DSP00510
TOTVT3 - TOTAL NO. OF VERTICES IN INPUT TEST FIELDS.       DSP00520
NOSUB3 - NO. OF SUBCLASSES USED IN CLASSIFY PLUS ONE, FOR THE DSP00530
          THRESHOLD CLASS.
PCFDKY - KEY INDICATING WHETHER OR NOT GROUND TRUTH PERFORMANCE DSP00540
          REPORTS ARE TO BE PRINTED ON A PER FIELD BASIS.
TSTKEY - KEY INDICATING WHETHER OR NOT TEST FIELDS WERE INPUT. DSP00550
TRNKEY - KEY INDICATING WHETHER OR NOT TRAINING FIELDS ARE TO DSP00560
          BE OUTLINED.
CONTINUE                                                    DSP00570
THRSKY - THRESHOLD KEY                                       DSP00580
          =1 APPLY CHI-SQUARE THRESHOLDS                      DSP00590
          =2 APPLY EMPIRICAL THRESHOLDS                       DSP00600
          =3 APPLY USER-INPUT THRESHOLDS                     DSP00610
          =4 APPLY FISHER DISTRIBUTION THRESHOLD              DSP00620
          =0 NO THRESHOLDING                                  DSP00630
STATKY - KEY FOR PRINTING STATS FROM MAPTAP                 DSP00640
EMPTRS - EMPIRICAL THRESHOLDING FLAG                        DSP00650
THRSVA - USER-INPUT THRESHOLD VALUE FLAG                   DSP00660
PLTKY - FLAG FOR PRINTING CUMMULATIVE HISTOGRAMS OF QUADRATIC DSP00670
          FORM.
BMFLG - FLAG INDICATING WHETHER OR NOT A B-MATRIX WAS      DSP00680
          APPLIED IN CLASSIFY.                                DSP00690
BMCOMB - NO. OF LINEAR COMBINATIONS IN H-MATRIX            DSP00700
BMFEAT - NO. OF CHANNELS USED IN COMPUTING B-MATRIX         DSP00710
                                                                DSP00720
                                                                DSP00730
                                                                DSP00740
                                                                DSP00750
                                                                DSP00760

```

FILE DISPLAY

```

C*      COATE - DATE OF CLASSIFICATION                                DSP00770
C*      FLOSV2 - ADDRESS IN 'ARRAY' FOR TRAINING FIELD INFORMATION.  DSP00780
C*              FOR EACH TRAINING FIELD 4 PIECES OF INFORMATION ARE DSP00790
C*              STORED - 1=FIELD NAME                                DSP00800
C*                      2=CLASS NO.                                DSP00810
C*                      3=SUBCLASS NO.                             DSP00820
C*                      4=NO. OF VERTICES                           DSP00830
C*      FIELD2 - ADDRESS IN 'ARRAY' FOR RECTANGULAR AREA SURROUNDING DSP00840
C*              EACH TRAINING FIELD. FOR EACH TRAINING FIELD 5 PIECES DSP00850
C*              OF INFORMATION ARE STORED.                           DSP00860
C*              1=LINE START                                        DSP00870
C*              2=LINE END                                        DSP00880
C*      CONTINUE                                                    DSP00890
C*              3=SAMPLE START                                        DSP00900
C*              4=SAMPLE END                                        DSP00910
C*              5=POINTER INTO VERTEX ARRAY FOR VERTICES          DSP00920
C*              OF THIS FIELD.                                       DSP00930
C*      VERTX2 - ADDRESS IN 'ARRAY' FOR TRAINING FIELD VERTICES.    DSP00940
C*      FLOSV3 - SAME AS FLOSV2 FOR TEST FIELDS                      DSP00950
C*      FIELD3 - SAME AS FIELD2 FOR TEST FIELDS                      DSP00960
C*      VERTX3 - SAME AS VERTX2 FOR TEST FIELDS                      DSP00970
C*      PCTIO3 - ADDRESS IN 'ARRAY' FOR PERFORMANCE TABLE.         DSP00980
C*      THRES - THRESHOLD VALUES                                    DSP00990
C*      SYMPTX - SYMBOLS FOR EACH SUBCLASS, PLUS THRESHOLD SYMBOL    DSP01000
C*              AND OUTLINE SYMBOLS.                                  DSP01010
C*      HIGH - THRESHOLD REJECTION PERCENTAGE - EMPIRICAL OPTION    DSP01020
C*      CON - CONSTANT FACTOR FROM PROBABILITY DENSITY FUNCTION     DSP01030
C*              FROM CLASSIFY. ONE FOR EACH SUBCLASS.              DSP01040
C*      FLDKEY - KEY INDICATING WHETHER GROUND TRUTH FIELDS ARE    DSP01050
C*              ASSOCIATED WITH CLASSES OR SUBCLASSES.             DSP01060
C*      NOFLD2 - NO. OF TRAINING FIELDS                               DSP01070
C*      NOFLD3 - NO. OF TEST FIELDS                                  DSP01080
C*      NOFET2 - NO. OF CHANNELS USED IN CLASSIFICATION.            DSP01090
C*      FETVC2 - CHANNELS USED IN CLASSIFICATION.                   DSP01100
C*      NOSUB2 - NO. OF SUBCLASSES USED IN CLASSIFICATION.         DSP01110
C*      NOTRFD - NO. OF GROUND TRUTH FIELDS FOR WHICH PERFORMANCE  DSP01120
C*              TABLES WILL BE MADE. EQUALS NOFLD3 OR NOFLD2.     DSP01130
C*      TOTVT2 - TOTAL NO. OF VERTICES FOR TRAINING FIELDS.        DSP01140
C*      NUCLS2 - NO. OF CLASSES USED IN CLASSIFICATION.            DSP01150
C*      KATNO - CLASS - CATEGORY CORRESPONDENCE VECTOR              DSP01160
C*      CONTINUE                                                    DSP01170
C*              (KATNO(I)=M MEANS CLASS I IS IN CATEGORY M)         DSP01180
C*      NOCAT - NO. OF CATEGORIES.                                    DSP01190
C*      FILTER - FLAG FOR SPATIAL FILTERING OPTION.                 DSP01200
C*      MAPFMT - FORMAT FOR OUTPUT MAP TAPE                          DSP01210
C*      DESKEY - KEY INDICATING WHETHER OR NOT DESIGNATED FIELDS WERE IN DSP01220
C*      DESUNI - NO. FOR DESIGNATED UNIDENTIFIABLE (NOSUB2+5)      DSP01230
C*      DESOTH - NO. FOR DESIGNATED OTHER (NOSUB2+6)                DSP01240
C*      CROP - NAME OF CROP FOR WHICH INTENSIVE TEST SITE SUMMARY  DSP01250
C*              REPORT IS TO BE PRINTED. CROP IS TO BE COMPARED WITH OT DSP01260
C*      ACROP - ACRES OF 'CROP' - USER INPUT                        DSP01270
C*      AOTHER - ACRES OF 'OTHER' - USER INPUT                      DSP01280
C*      ATOTAL - TOTAL ACRES IN CLASSIFIED SEGMENT                  DSP01290
C*      SITE - NAME OF SITE (CLASSIFIED SEGMENT)                    DSP01300
C*      ANALYS - NAME OF ANALYST PERFORMING STUDY                   DSP01310
C*      CAMS - NAME OF PROCEDURE CONFIGURATION USED IN STUDY       DSP01320
C*      CRPKEY - KEY FOR GENERATING INTENSIVE TEST SITE SUMMARY REPORT DSP01330
C*      CRPPTS - TOTAL NUMBER PIXELS IN EACH SUBCLASS              DSP01340
C*      DOTKEY - KEY INDICATING WHETHER OR NOT DOT DATA CLASSIFICATION DSP01350
C*      PERFORMANCE SUMMARIES ARE TO BE PROCESSED: DOTKEY = 0 , NO DOT DSP01360
C*      DATA PROCESSING ; DOTKEY .GT. 0 , DOT PERFORMANCE SUMMARIES DSP01370
C*      ARE PROVIDED (CHANGED TO INDICATE LIST PROCESSING           DSP01380
C*      INSTEAD OF DOT PROCESSING ON MAY 1979)                      DSP01390
C*      DOTERR USE OF THIS FLAG REMOVED MAY 1979                    DSP01400
C*      FLAG NOT NEEDED WHEN LIST SUBSTITUTED FOR DOT PROCESSING   DSP01410
C*      CONTINUE                                                    DSP01420
C*      DSP01430
C*      DSP01440
C*      DSP01450
C*      DSP01460
C*      DSP01470
C*      DSP01480
C*      DSP01490
C*      DSP01500
C*      DSP01510
C*      DSP01520
C*      SETUP3 WILL READ FIRST 2 RECORDS FROM MAPTAP, AND CALL REDIF3
C*      TO READ IN CONTROL CARDS. ALL OF THE PARAMETERS IN COMMON BLOCK
C*      DISPL ARE INITIALIZE BEFORE RETURNING TO THIS ROUTINE IN ADDITION
C*      TRAINING AND/OR TEST FIELD DEFINITIONS WILL BE STORED IN 'ARRAY'

```

FILE DISPLAY

```

C*
C**
C*** THIS ADDED OR CHANGED NOV.13,1978 TO INCLUDE LIST PROS.
C
DIMENSION DESSAV(4,50),DESFLD(5,50),DESVER(1100)
REAL ALP(2)
STOP=0
CALL SETUP3(ARRAY, TOP, GTUNIT, GTFILE, AIUNIT, AIFILE,
* PPUNIT, PPFIL, NAMECT, ALP, DESSAV, DESFLD, DESVER, NOFLD4, STOP)
IF (STOP.FQ.0) GO TO 5
6100 WRITE(6,6100)
FORMAT(//,1X,'NO MAP AVAILABLE FROM CLASSIFICATION PROCESSOR.')
```

DSP01530  
 DSP01540  
 DSP01550  
 DSP01560  
 DSP01570  
 DSP01580  
 DSP01590  
 DSP01600  
 DSP01610  
 DSP01620  
 DSP01630  
 DSP01640  
 DSP01650  
 DSP01660  
 DSP01670  
 DSP01680  
 DSP01690  
 DSP01700  
 DSP01710  
 DSP01720  
 DSP01730  
 DSP01740  
 DSP01750  
 DSP01760  
 DSP01770  
 DSP01780  
 DSP01790  
 DSP01800  
 DSP01810  
 DSP01820  
 DSP01830  
 DSP01840  
 DSP01850  
 DSP01860  
 DSP01870  
 DSP01880  
 DSP01890  
 DSP01900  
 DSP01910  
 DSP01920  
 DSP01930  
 DSP01940  
 DSP01950  
 DSP01960  
 DSP01970  
 DSP01980  
 DSP01990  
 DSP02000  
 DSP02010  
 DSP02020  
 DSP02030  
 DSP02040  
 DSP02050  
 DSP02060  
 DSP02070  
 DSP02080  
 DSP02090  
 DSP02100  
 DSP02110  
 DSP02120  
 DSP02130  
 DSP02140  
 DSP02150  
 DSP02160  
 DSP02170  
 DSP02180  
 DSP02190  
 DSP02200  
 DSP02210  
 DSP02220  
 DSP02230  
 DSP02240

```

C*
C** DSPLY1 WILL READ NEXT 2 RECORDS FROM MAPTAP AND PRINT THE
C** STATISTICS IF REQUESTED.
C**
C** CALL DSPLY1
C***
C**** CODE ADDED NOV. 13,1978 TO INCLUDE LIST PROCESSING
C**** CODE CHANGED MAY 1979 TO SUBSTITUTE LIST FOR DOTS
C
IF (DOTKEY.NE.0) GO TO 30
IF (EMPTRS.NE.2.AND. PLTKEY.NE.1) GO TO 30
C*
C** EMTHRS COMPUTES AND PLOTS THE HISTOGRAM OF THE QUADRATIC FORM
C** FOR THE CORRECTLY CLASSIFIED PIXELS WITHIN THE TRAINING OR TEST
C** FIELDS.
C*
IF (FLDKEY.EQ.1) GO TO 10
WRITE(6,100)
GO TO 30
10 CONTINUE
IF (TSTKEY.EQ.1) CALL EMTHRS (ARRAY (FLDSV3), ARRAY (FIELD3),
* ARRAY (VERTX3), NOFLD3)
IF (TSTKEY.NE.1) CALL EMTHRS (ARRAY (FLDSV2), ARRAY (FIELD2),
* ARRAY (VERTX2), NOFLD2)
30 CONTINUE
C-
C- TEST THRSKY = 4 FOR FISHER F-DISTRIBUTION THRESHOLDS
C- CALL FDIST TO COMPUTE AND STORE THRESHOLDS
C-
IF (THRSKY.EQ.4) CALL FDIST
C-
C*
C** DSPLY2 PRINTS THE MAP AND CALLS PCT TO BUILD PERFORMANCE TABLES.
C*
CALL DSPLY2 (ARRAY (FLDSV2), ARRAY (FIELD2), ARRAY (VERTX2),
* ARRAY (FLDSV3), ARRAY (FIELD3), ARRAY (VERTX3),
* ARRAY (PCTID3), GTUNIT, GTFILE,
* AIUNIT, AIFILE, PPUNIT, PPFIL, NAMECT, ALP,
* DESSAV, DESFLD, DESVER, NOFLD4)
C*
C** IF DOT DATA PROCESSING WAS REQUESTED, THE PERFORMANCE
C** TABLES WERE PERFORMED IN DSPLY2
C**
C***
C**** CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
C**** CODE CHANGED MAY 1979 TO SUBSTITUTE LIST FOR DOT PROCESSING
C
IF (DOTKEY.GT.0) GO TO 99
C*
C** PRTPCT PRINTS THE PERFORMANCE TABLES
C*
IF (TSTKEY.NE.1) CALL PRTPCT (ARRAY (FLDSV2), ARRAY (PCTID3), NOFLD2)
99 IF (TSTKEY.EQ.1) CALL PRTPCT (ARRAY (FLDSV3), ARRAY (PCTID3), NOFLD3)
WRITE(6,6000)
6000 FORMAT(1H1X,' ***** DISPLAY COMPLETED *****')
100 FORMAT(//,1X,' **DISPLAY** - FIELDS MUST BE DEFINED FOR SUBCLASSES FOR
* EMPIRICAL THRESHOLDS')
999 RETURN
END
```

FILE: CHI

```
REAL FUNCTION CHI(X,N,IFLAG)                                CHI00010
-----
C TO COMPUTE THE VALUE OF THE CHI-SQUARED DISTRIBUTION WITH N-D.F. CHI00020
C-----
C IF(X.GT.0.0) GO TO 5                                       CHI00040
  CHI=0.0                                                    CHI00050
  RETURN                                                    CHI00060
C-----
C CHFK TO SEE IF THE DEGREES OF FREEDOM IS EVEN             CHI00070
C-----
C 5 IF(MOD(N,2).EQ.0) GO TO 1                                CHI00080
C-----
C CALCULATION OF CHI FOR 1 DEGREE OF FREEDOM                CHI00090
C-----
  G=SQRT(X)                                                 CHI00100
  CHI=2.0*RNORM(G)-1.0                                       CHI00110
  G=G/1.25331414                                            CHI00120
  IN=3                                                       CHI00130
  GO TO 2                                                    CHI00140
C-----
C CALCULATION OF CHI FOR 2 DEGREES OF FREEDOM              CHI00150
C-----
  1 IN=4                                                     CHI00160
  G=X/2.0                                                    CHI00170
  IF(ABS(G).GT.88.027) GOTO 4                               CHI00180
  CHI=1.0-EXP(-G)                                           CHI00190
  2 IF(N.LT.3) RETURN                                       CHI00200
  IF(ABS(X/2.0).GT.88.027) GOTO 4                          CHI00210
  G=G*EXP(-X/2.0)                                           CHI00220
C-----
C CALCULATION OF CHI FOR N-GT-2 DEGREES OF FREEDOM        CHI00230
C-----
  DO 3 I=IN,N,2                                             CHI00240
  CHI=CHI-G                                                 CHI00250
  G=G*X/I                                                   CHI00260
  CALL OVERFL(INDCT)                                        CHI00270
  IF(INDCT.EQ.1) GOTO 4                                    CHI00280
  3 CONTINUE                                               CHI00290
  RETURN                                                    CHI00300
  4 IFLAG=1                                                 CHI00310
  RETURN                                                    CHI00320
  END                                                       CHI00330
                                                         CHI00340
                                                         CHI00350
                                                         CHI00360
                                                         CHI00370
                                                         CHI00380
                                                         CHI00390
                                                         CHI00400
                                                         CHI00410
                                                         CHI00420
```

FILE: CHIN

ORIGINAL PAGE IS  
OF POOR QUALITY

```
FUNCTION CHIN(ALPHA,N,IFLAG)
DIMENSION H(7), D(15)
EQUIVALENCE (H(7), D(15))
DATA D/-.7000,-.4020,-.1900,-.0600,.0060,.0360,.0360,
1 .0120,-.0100,-.0360,-.0300,.0120,.1020,.2580,.4920/
IFLAG=0
IF(N.EQ.1) CHIN=(TINORM((1.-ALPHA/2.),IFLAG))
IF(IFLAG.EQ.1) GOTO 10
CHIN=CHIN**2
IF(N.EQ.2) CHIN = -2. * ALOG(ALPHA)
IF(N.LE.2) RETURN
X=TINORM(1.-ALPHA,IFLAG)
IF(IFLAG.EQ.1) GOTO 10
I=2.*X
IF(I.LT.-7) I=-7
IF(I.GT.6) I=6
Y=(H(I)+(2.*X-I)*(H(I+1)-H(I)))/N
CHIN=2./(9.*N)
X=N*(1.-CHIN+(X-Y)*SQRT(CHIN))**3
IF((N.LE.55.AND.X.LT.0.) ) X=N*(1.-CHIN+X*SQRT(CHIN))**3
IF(N.LE.55) GO TO 1
IF(X.GT.176.16) GOTO 10
CHIN=X
4 RETURN
1 IC=0
IE=N-2
IR=3
G=2.5066283
IF(MOD(N,2).EQ.1) GO TO 2
IR=2
G=2.
2 IF(IE.LE.1) GO TO 11
DO 3 I=IR,IE,2
3 G=G*I
11 N2 = (N-2)/2
N3 = N-2-N2
SQX = SQRT(X)
CHA = ((1.-CHI(X,N,IFLAG)-ALPHA)*G)/(SQX**N2)
IF(IFLAG.EQ.1) GOTO 10
CHR = EXP(X/2.)/(SQX**N3)
CHIN = X + CHA*CHR
IF(ABS(X-CHIN)/AMAX1(X,CHIN).LT.5.E-06) GO TO 4
IF(IC.GT.200) RETURN
IC=IC+1
X=CHIN
IF(X.GT.176.16) GOTO 10
GO TO 11
10 IFLAG=1
RETURN
END
```

CHI00010  
CHI00020  
CHI00030  
CHI00040  
CHI00050  
CHI00060  
CHI00070  
CHI00080  
CHI00090  
CHI00100  
CHI00110  
CHI00120  
CHI00130  
CHI00140  
CHI00150  
CHI00160  
CHI00170  
CHI00180  
CHI00190  
CHI00200  
CHI00210  
CHI00220  
CHI00230  
CHI00240  
CHI00250  
CHI00260  
CHI00270  
CHI00280  
CHI00290  
CHI00300  
CHI00310  
CHI00320  
CHI00330  
CHI00340  
CHI00350  
CHI00360  
CHI00370  
CHI00380  
CHI00390  
CHI00400  
CHI00410  
CHI00420  
CHI00430  
CHI00440  
CHI00450  
CHI00460  
CHI00470  
CHI00480  
CHI00490  
CHI00500



FILE: DESIG

```

      SURROUTINE DESIG(LINE,IR,FLDSAV,FIELD ,VERTEX,NOFLD,
      SAMSTR,SAMEND,SAMINC)
      IMPLICIT INTEGER(A-Z)
      DIMENSION IR(1),FIELD(5,NOFLD), FLDSAV(4,NOFLD),VERTEX(1),FL(22)
C*
C* THIS ROUTINE SETS THE IR ARRAY FOR DESIGNATED FIELDS
C*
      DO 50 I=1,NOFLD
      IF(LINE .LT.FIELD(1,I))GO TO 50
      IF(LINE .GT.FIELD(2,I))GO TO 50
      IF(FIELD(3,I).GT.SAMEND)GO TO 50
      IF(FIELD(4,I).LT.SAMSTR)GO TO 50
C*
C* FOUND A DESIGNATED FIELD ON THIS LINE
C*
      NV=FLDSAV(4,I)
      IPT=FIELD(5,I)
      ID = FLDSAV(2,I)
      CALL FDLINT(VERTEX(IPT),NV,FL,LINE,SAMPS,NI)
      DO 20 J=1,NI,2
      IB = (FL(J)-SAMSTR)/SAMINC+1
      IE = (FL(J+1)-SAMSTR)/SAMINC+1
      IF(MOD(SAMSTR,SAMINC) .NE. MOD(FL(J),SAMINC)) IB=IB+1
      IF(IR.GT.IE)GO TO 20
      DO 10 K=IB,IE
10  IR(K)=ID
20  CONTINUE
50  CONTINUE
      RETURN
      END
```

~~12-6~~  
185

FILE: DISTCV

```

SURROUTINE DISTCV(DSFUNC,TOTPTS,RANGE)                                DIS00010
IMPLICIT INTEGER(A-Z)                                              DIS00020
.....C.....DIS00030
DISTCV PLOTS THE DISTRIBUTION AND CHI SQUARE CURVES AND COMPUTES  DIS00040
THE EMPIRICAL THRESHOLD VALUES                                   DIS00050
.....C.....DIS00060
REAL Q, DSFUNC, DISTVL, THRESH, REJECT, PCTREJ, THRES, CHISQ1      DIS00070
REAL RFJPCT, CHISO, INC, CHIN                                       DIS00080
DIMENSION DSFUNC(RANGE,60), TOTPTS(1), THRES(60), MINM(60),      DIS00090
SYMBOLS(100), FIELDS(2), FIELD1(2), PCTREJ(60)                   DIS00100
DIMENSION CHISO(100)                                               DIS00110
INCLUDE CMRKN.LIST                                                DIS00120
COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60),  DIS00130
SURCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,                       DIS00140
PCFOKY,ISTKEY,TRNKEY,THRSKY,STATKY,EMPTR5,THRSVA,                DIS00150
PLTKFY,RMFLG,RMCONH,RMFFAT,CUATE(2),                             DIS00160
FLOSV2,FIELD2,VERTX2,FLOSV3,FIELD3,VERTX3,PCTID3,                DIS00170
THRES(60),SYMTX(66),HIGH(60),CON(60)                             DIS00180
FLOKEY,NOFLD2,NOFLD3,NOFET2,FETVC2(30)                           DIS00190
NOSUB2,NOTHFD,TOTVT2,NOCLS2                                     DIS00200
KATNO(60),NOCAT,FILTER,MAPFMT                                    DIS00210
DESKEY,DESUNI,DESOTH,CROP,ACROP,AOTHER,ATOTAL                    DIS00220
SITE(6),ANALYS(5),CAM(15),CRPKY,KEPPTS(60)                       DIS00230
DOTKEY,DOTERR                                                    DIS00240
CSEND                                                             DIS00250
DATA FIELD /'TEST'/. FIELDS/'TRAI','NING'/.                       DIS00260
DATA BLANK/' ' /'.MXQUAD/20/.                                     DIS00270
DATA ASTK /'%' /'.DOLLAR /'$/'.                                  DIS00280
EQUIVALENCE ( HIGH(1),PCTREJ(1) )                                DIS00290
DO 5 I=1,2                                                         DIS00300
5 FLD1(I) = BLANK                                                 DIS00310
DO 10 L=1,99                                                       DIS00320
REJECT = 1. - FLOAT(L) / 100.0                                     DIS00330
CHISQ1 = CHIN(REJPCT,NOFET2,FLAG)                                 DIS00340
IF (FLAG.EQ.1) GO TO 13                                           DIS00350
CHISQ(L) = CHISQ1 + 0.05                                         DIS00360
GO TO 10                                                           DIS00370
13 WRITE(6,140) REJPCT,CHISQ1                                     DIS00380
140 FORMAT(' OVERFLOW',2X,F5.2,3X,F15.5)                          DIS00390
10 CONTINUE                                                        DIS00400
CALL SETMRG(66,0.66)                                              DIS00410
DO 200 J=1,NOSUB2                                                 DIS00420
TOTPTS(J)=0                                                       DIS00430
DO 20 M=1,RANGE                                                   DIS00440
TOTPTS(J) = DSFUNC(M,J) + TOTPTS(J)                               DIS00450
CONTINUE                                                           DIS00460
20 CONTINUE                                                        DIS00470
IF (TOTPTS(J) .EQ. 0) GO TO 200                                   DIS00480
DO 40 M=1,RANGE                                                   DIS00490
MM = M - 1                                                         DIS00500
IF (MM .EQ. 0) DSFUNC(MM+1,J) = (DSFUNC(MM+1,J) / TOTPTS(J)) * 100 DIS00510
IF (MM .EQ. 0) GO TO 40                                           DIS00520
DSFUNC(M,J) = DSFUNC(MM,J) + (DSFUNC(M,J) / TOTPTS(J)) * 100    DIS00530
40 CONTINUE                                                        DIS00540
.....C.....DIS00550
FIND MINIMUM M FOR WHICH DSFUNC(M,J) .GT. 1 - 100*PCTREJ        DIS00560
.....C.....DIS00570
DISTVL = (1. - PCTREJ(J)) * 100                                    DIS00580
MXREJT = DISTVL + 0.5                                             DIS00590
DO 50 M=1,RANGE                                                   DIS00600
IF (DISTVL .LE. DSFUNC(M,J) ) MINM(J) = M                         DIS00610
IF (DISTVL .LE. DSFUNC(M,J)) GO TO 55                             DIS00620
50 CONTINUE                                                        DIS00630
55 CONTINUE                                                        DIS00640
.....C.....DIS00650
THRESHOLD                                                         DIS00660
.....C.....DIS00670
REJECT = PCTREJ(J) * 100                                         DIS00680
IF (ISTKEY .EQ. 1) FIELD1(1) = FIELD                               DIS00690
IF (ISTKEY .EQ. 0) FIELD1(1) = FIELDS(1)                         DIS00700
IF (ISTKEY .EQ. 0) FIELD1(2) = FIELDS(2)                         DIS00710
THRESH(J) = (0.1 * MINM(J)) * 2                                   DIS00720
.....C.....DIS00730
PRINT HEADING                                                     DIS00740
.....C.....DIS00750
CLASNO=J                                                           DIS00760
.....C.....DIS00770
.....C.....DIS00780
.....C.....DIS00790

```

FILE: DISTCV

```
WRITE(6,100)SUBNAM(J)
100 FORMAT(1H1,///T52,'DISTRIBUTION CURVE FOR SURCLASS ',A4)
WRITE(6,105) THRES(CLASNO),REJECT,(FIELD1(I),I=1,2),THRESH(J)
105 FORMAT(///T10,'CHI SQUARE THRESHOLD = ',F5.2,T53,'EMPIRICAL THRESHOLD FROM ',T91,'USER REJECTION PERCENTAGE = ',F4.1/T53,
* 2A4,' FIELDS = ',F5.2)
WRITE(6,110)
110 FORMAT(///T15,10(1H0,9X),1H1,/T15,'0',9X,'1',9X,'2',9X,'3',9X,
* '4',9X,'5',9X,'6',9X,'7',9X,'8',9X,'9',9X,'0'//
* T15,11(1H0,9X)/
* T11,'0.0',1X,101(1H*))
INC = FLOAT(MXQUAD)/FLOAT(RANGE)
II = 1
N = 0
DO 90 L=1,RANGE
Q = FLOAT(MXQUAD*L) / RANGE
N = N + 1
DO 60 M=1,100
60 SYMBS(M) = BLANK

CCC
CHI SQUARE CURVE
65 IF ( II .GT. MXREJT ) GO TO 75
IF ( CHISQ(II) .GE. (Q+INC) ) GO TO 75
SYMBS(II) = ASTK
II = II + 1
GO TO 65
75 CONTINUE

CCC
DISTRIBUTION CURVE
PERCNT = DSFUNC(L,J)
IF (PERCNT .EQ. 0) GO TO 77
IF (SYMBS(PERCNT) .NE. BLANK) SYMBS(PERCNT) = DOLLAR
IF (SYMBS(PERCNT) .NE. BLANK) GO TO 77
76 SYMB=J
SYMBS(PERCNT) = SYMMTX(SYMB)
77 CONTINUE
IF (N .EQ. 5) GO TO 80
WRITE(6,120) (SYMBS(K),K=1,100)
120 FORMAT(T15,'+',100A1)
GO TO 90
80 WRITE(6,130) Q, (SYMBS(K),K=1,100)
130 FORMAT(T10,F4.1,1X,'+',100A1)
N = 0
90 CONTINUE
WRITE(6,150) SYMMTX(SYMB)
150 FORMAT(///T10,'NOTE : ',A1,' - CLASS DISTRIBUTION CURVE'//T18,
* '* - CHI SQUARE DISTRIBUTION CURVE'//T18,'S - INTERSECTION OF CURVE'//
* FS)
IF (EMPTRS .EQ. 0) GO TO 200
THRES(J)=THRESH(J)
200 CONTINUE
CALL SETMRG(66,4,62)
RETURN
END
DIS00800
DIS00810
DIS00820
DIS00830
DIS00840
DIS00850
DIS00860
DIS00870
DIS00880
DIS00890
DIS00900
DIS00910
DIS00920
DIS00930
DIS00940
DIS00950
DIS00960
DIS00970
DIS00980
DIS00990
DIS01000
DIS01010
DIS01020
DIS01030
DIS01040
DIS01050
DIS01060
DIS01070
DIS01080
DIS01090
DIS01100
DIS01110
DIS01120
DIS01130
DIS01140
DIS01150
DIS01160
DIS01170
DIS01180
DIS01190
DIS01200
DIS01210
DIS01220
DIS01230
DIS01240
DIS01250
DIS01260
DIS01270
DIS01280
DIS01290
DIS01300
DIS01310
DIS01320
DIS01330
DIS01340
DIS01350
```

FILE: DSPLY1

```

SUBROUTINE DSPLY1
IMPLICIT INTEGER (A-M,0-Z)
-----
PURPOSE.. READS THE STATISTICS FROM 'MAPTAP'
-----

INCLUDE COMRK6.LIST
REAL CON,DET(60)
INCLUDE CMHK10.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWD5,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTF1,SCTRUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CPDUNT,PRUNT,RANDIU
COMMON/DISPL/CATFLG,CATNAM(6),CLSNAM(61),SUBNAM(61),SUBNO(60),
* SURCAT(60),CLSSUR(60),NOMAP,TOTVT3,NOSUB3,
* PCFDKY,TSTKEY,TRNKEY,THRSKY,STATKY,EMPTRS,THRSVA,
* PLTKEY,BMFLG,BMCOMB,BMFEAT,CDATE(2),
* FLDSV2,FIELD2,VERTX2,FLDSV3,FIELD3,VERTX3,PCTID3,
* THRS(60),SYMMTX(66),HIGH(60),CON(60)
* ,FLOKEY,NOFLD2,NOFLD3,NOFET2,FETVC2(30)
* ,NOSUR2,NOTREFD,TOTVT2,NOCLS2
* ,KATNO(60),NOCAT,FILTER,MAPFMT
* ,DESKEY,DESUNI,DESOTH,CROP ,ACROP,AOTHER,ATOTAL
* ,SITE(6),ANALYS(5),CAM(15),CRPKEY,KEPPTS(60)
* ,DOTKEY,DOTERR

DIMENSION STORAGE(9500)
DATA SIZE/9500/
DATA ACOFOR/'4'//,BCDWO/'2'//,DASH/'----'//
-----
RETRIEVE AND PRINT THE COVARIANCE AND MEAN
-----
200 IF( BMFLG .LE. 0) GO TO 202
NOFET2 = BMCOMB
202 CV=1
VARSZ2=NOFET2*(NOFET2+1)/2
MN=CV + VARSZ2*NOSUR2
IF(MN + NOSUR2*NOFET2 .GT. SIZE)GO TO 180
GO TO 190
170 CONTINUE
CALL DSPLIA(STORAGE(CV),STORAGE(MN),VARSZ2,NOFET2,NOSUB2)
RETURN
190 WRITE(6,1A1)
181 FORMAT(' NOT ENOUGH STORAGE FOR COVARIANCE MATRICES - DSPLY1')
CALL CMERR
CONTINUE
RETURN

END --- SUBROUTINE DSPLY1
-----

INTERNAL SUBROUTINE USPLIA
SUBROUTINE USPLIA(COVMTX,AVEMTTX,VARSZ2,NOFET2,NOCLS2)
COVMTX(1,1)=STORAGE(1)
AVEMTX(I,J)=STORAGE(MN-1+I+(J-1)*NOFET2)
190 CONTINUE

READ ORIGINAL COVARIANCE AND MEANS MATRIX FOR EACH CLASS
( H-TRANSFORMED IF H-MATRIX WAS APPLIED IN SCLASSIFY -
BMFLG .GT. 0 IF SO )

```

12-9  
188

FILE: DSPLY1

```
DUMMY=NOCLS2
NOCLS2=NOCL52
NS = VARS22 * NOCLS2
NSS = NOFFT2 * NOCLS2
READ(MAPTAP) (STORAG(CV+I-1), I=1, NS), (STORAG(MN+I-1), I=1, NSS)
C
IF (STATKY.EQ.0) GO TO 290
C
CNT = 7*(5+3*2*NOFET2)*((NOFET2+11)/12)
CNT = PAGESZ/CNT
INC = CNT
C
DO 280 I=1, NOCLS2
IF (INC.LT.CNT) GO TO 210
WRITE (6, HEAD)
INC = 0
210 WRITE (6, 270) SUBNAM(ICLAS), SYMPTX(ICLAS), (DASH, I=1, 5)
220 FORMAT(//, 'SUBCLASS', A4, ' REPRESENTED BY SYMBOL - ', A1 /
      * 1X, 3A4, A1, T44, A3/)
DO 230 LOC=1, NOFET2, 12
STOP = LOC+1
IF (STOP.GT. NOFET2) STOP = NOFET2
NS=MN-1+(ICLAS-1)*NOFET2
WRITE (6, 240) (STORAG(NS+I), I=LOC, STOP)
230 CONTINUE
260 FORMAT('0 MEAN:', 3X, 12F9.2)
WRITE (6, 240) DASH
2601 FORMAT(1X, A4/)
C
IF (RMFLG.GT. 0) GO TO 271
WRITE (6, 270) (DASH, I=1, 5)
GO TO 272
270 FORMAT('0 COVARIANCE MATRIX: ' / 1X, 5A4)
271 WRITE (6, 271) (DASH, I=1, 9)
2711 FORMAT('0 COVARIANCE MATRIX (B-TRANSFORMED) ' / 1X, 8A4, A3)
272 NS=1+(ICLAS-1)*VARS22
CALL WRMTX(STORAG(NS), NOFET2, BCDTWO)
INC = INC+1
280 CONTINUE
C
READ COVARIANCE MATRIX ( AFTER CHOLESKY FACTORIZATION),
PROBABILITY DENSITY FUNCTION CONSTANTS, CON, AND COVARIANCE
MATRIX DETERMINANT, DET, FOR EACH CLASS
-----
290 CONTINUE
NS = VARS22 * NOCLS2
READ(MAPTAP) (STORAG(CV+I-1), I=1, NS), (CON(I), I=1, NOCLS2), (DET(I),
* I=1, NOCLS2)
IF (STATKY.EQ.0) GO TO 330
C
CNT = 11*(3+2*NOFET2)*((NOFET2+11)/12)
CNT = PAGESZ/CNT
INC = CNT
DO 310 I=1, NOCLS2
IF (INC.LT.CNT) GO TO 300
WRITE (6, HEAD)
INC = 0
300 WRITE (6, 320) SUBNAM(I), SYMPTX(I), DET(I), CON(I)
NS = 1+(I-1)*VARS22
CALL WRMTX(STORAG(NS), NOFET2, BCDFOR)
INC = INC+1
310 CONTINUE
320 FORMAT(//1X, 'MULTSPECTRAL CHARACTERISTICS FOR SUBCLASS ', A4, '
1 REPRESENTED BY SYMBOL ', A1/1H0, ' DETERMINANT = ', F18.8/1H0, ' CONSTATNDS
2 T TERM = ', F10.4//1H0, ' COVARIANCE MATRIX (CHOLESKY DECOMPOSITION)
1 : ' )
C
GO HOME
-----
330 CONTINUE
NOCLS2=DUMMY
GO TO 170
END
```

DSP00800  
DSP00810  
DSP00820  
DSP00830  
DSP00840  
DSP00850  
DSP00860  
DSP00870  
DSP00880  
DSP00890  
DSP00900  
DSP00910  
DSP00920  
DSP00930  
DSP00940  
DSP00950  
DSP00960  
DSP00970  
DSP00980  
DSP00990  
DSP01000  
DSP01010  
DSP01020  
DSP01030  
DSP01040  
DSP01050  
DSP01060  
DSP01070  
DSP01080  
DSP01090  
DSP01100  
DSP01110  
DSP01120  
DSP01130  
DSP01140  
DSP01150  
DSP01160  
DSP01170  
DSP01180  
DSP01190  
DSP0200  
DSP0210  
DSP0220  
DSP0230  
DSP0240  
DSP0250  
DSP0260  
DSP0270  
DSP0280  
DSP0290  
DSP0300  
DSP0310  
DSP0320  
DSP0330  
DSP0340  
DSP0350  
DSP0360  
DSP0370  
DSP0380  
DSP0390  
DSP0400  
DSP0410  
DSP0420  
DSP0430  
DSP0440  
DSP0450  
DSP0460  
DSP0470  
DSP0480  
DSP0490  
DSP0500  
DSP0510  
DSP0520  
DSP0530  
DSP0540  
DSP0550

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE DSPLY2

```

SUBROUTINE DSPLY2 (TRNSAV, TRNFLD, TRNVER, TSTSAV, TSTFLD,
* TSTVER, PCTAB, GTUNIT, GFILE, ATUNIT, AIFILE,
* PPUNIT, PPFIL, NAMECT, ALP, DESSAV, DESFLD, DESVER, NOFLO4)
  IMPLICIT INTEGER (A-Z)
C*** CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING
      REAL ALP(2)
-----
      DSPLY2 CAN PERFORM SPATIAL FILTERING ON THE CLASSIFIED DATA.
      (I.E. - THE FOUR NEAREST NEIGHBORS OF EACH PIXEL ARE
      TESTED FOR 'SAMENESS'. IF THE FOUR NEIGHBORS ARE
      CLASSIFIED AS ONE TYPE AND THE PIXEL IN QUESTION
      WAS CLASSIFIED AS A DIFFERENT TYPE, THE CLASSIFICATION
      IS CHANGED TO THE SAME AS THE NEIGHBORS.)
      EXAMPLE:
      LINE N = C CXC = X IS CHANGED TO C
      N+1 = CXC =
      N+2 = C =
-----
      DSPLY2 ALSO PERFORMS THRESHOLDING, PRINTS THE CLASSIFICATION
      MAP, AND CALLS THE APPROPRIATE ROUTINES TO BUILD AND THEN PRINT
      THE CLASSIFICATION PERFORMANCE TABLES.

      LIST PROCESSING ADDED NOV 13, 1978 DOTKEY IS THE
      NAME OF THE SWITCH (CHANGED FROM LISTSW MAY 1979)
-----
      REAL CON
      DIMENSION DESSAV(4,50), DESFLD(5,50), DESVER(1100), PCTAB(500,1)
      DIMENSION TRNSAV(4,NOFLD2), TRNFLD(5,NOFLD2), TRNVER(2,TOTVT2)
      * TSTSAV(4,NOFLD3), TSTFLD(5,NOFLD3), TSTVER(2,TOTVT3)
      * COL(3,110), SCRAT(330), FLDINF(6)
      * IR(1000,3), VR(1000), OUT(1000), ILINE(3), BUF(110,20)
      * JSTAT(20), PCTAB(NOTRFD,NOSUB3)
      INCLUDE CM9K10,LIST
      INCLUDE CM4K6
      COMMON/GLOBAL/HEAD(63), MAPTAB, DATAE, SAVTAB, BMFILE, BMKEY,
      * HISFIL, HISKEY, TRFORM, ERIP, ERKEY, MAPUNT, NOFILE,
      * DRUMAD, DRUMDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
      * NHSTUN, NHSTFI, SCTRIN, MAPFIL
      * DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
      * CRDUNT, PRTUNT, RANDIO
      COMMON/DISPL/CATFLG, CATNAM(61), CLSNAM(61), SUBNAM(61), SUBNO(60),
      * SUBCAT(60), CLSSUP(60), NOMAP, TOTVT3, NOSUB3,
      * PCFDKY, TSTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,
      * PLTKEY, BMFLG, RMCOMB, BMFEAT, CDAT(2),
      * FLDSV2, FIELD2, VERTX2, FLDSV3, FIELD3, VERTX3, PCTID3,
      * THRS(60), SYMTX(66), HIGH(60), CON(60)
      * FLDKEY, NOFLD2, NOFLD3, NOFET2, FETVC2(30)
      * NOSUB2, NOTRFD, TOTVT2, NOCLS2
      * KATNO(60), NOCAT, FILTER, MAPFMT
      * DESKEY, DESUNT, DESOTH, CHOP, ACROP, AOTHER, ATOTAL
      * SITE(6), ANALYS(5), CAM(15), CRPKEY, KEPPYS(60)
      * DOTKEY, DOTERR
      CSEND
      LOGICAL START, FULL
      DATA AST/'****'/
      DATA THRESH/'THRE'/.BLANK/'/'
      EQUIVALENCE (FLDINF(1), LINST), (FLDINF(2), LINEND),
      1 (FLDINF(3), LININC), (FLDINF(4), SAMSTR),
      2 (FLDINF(5), SAMEND), (FLDINF(6), SAMINC),
      3 (COL, SCRAT), (IR, BUF)
-----
      DIMENSION FORMAT(3,2)
-----

```

FILE DSPLY2 :

```

DATA FORMAT/UNIV,VERSA,IL  'LARS',YS I,II  '
REAL TOTALS(66),VR,THRES
DIMENSION TTOL(66)
-----
TRNNO=NOSUR3+1
TSTNO=NOSUR3+2
SET POINTERS FOR SYMBOLS ARRAY AND TOTALS ARRAY
DUPNO = NOSUR3+3
DESUNI= NOSUR3+4
DESOTH= NOSUR3+5
FLAG USED IN DOTPCT TO INITIALIZE PCTAB=0
PCTKEY=0
CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
      FLD CNT = 0
      IF (NOTKEY.EQ.0) GO TO 17
      DO 16 I = 1,209
      LR = (I - 1)/19
      LR = (LR + 1)*10
      LS = (LR - 1)/10
      LS = 10*(I - (LS*19))
      TRNVER(1,I) = LS
      TRNVER(2,I) = LR
16 CONTINUE
17 CONTINUE
DO 10 I=1,NOSUR3
DO 10 J=1,NOTRFD
10 PCTAB(J,I) = 0
-----
PRINT OUT HEADING
-----
20 READ(MAPTAP)FLDINF,PTS,LINES,FLDESC
IF (PTS.GT.1000) WRITE(6,22)
IF (PTS.GT.1000) STOP
22 FORMAT(' DISPLAY WILL ACCEPT ONLY 1000 PTS/SCAN LINE')
I STRT=SAMSTR
I END=SAMEND
IF (PTS.EQ.0) GO TO 310
MDREC = 1
DO 25 I=1,DESOTH
25 TTOL(I)=0
TOTALS(I)=0.0
-----
PRINT OUT THE COLUMN NUMBERS
-----
30 J = 0
CALL SETMRG(68,0,68)
IF (NOMAP.EQ.0) GO TO 85
  WPTFG=1
  GO TO 370
31 PPTFG=1
  GO TO 510
85 CONTINUE
SPKNT=0
COUNTR=0
DRUMLN=0
LAST=0
START=.FALSE.
FULL=.FALSE.
I1=1
I2=2
I3=3
J=1
ADRES=DRUMAD

```

DSP00770  
 DSP00780  
 DSP00790  
 DSP00800  
 DSP00810  
 DSP00820  
 DSP00830  
 DSP00840  
 DSP00850  
 DSP00860  
 DSP00870  
 DSP00880  
 DSP00890  
 DSP00900  
 DSP00910  
 DSP00920  
 DSP00930  
 DSP00940  
 DSP00950  
 DSP00960  
 DSP00970  
 DSP00980  
 DSP00990  
 DSP01000  
 DSP01010  
 DSP01020  
 DSP01030  
 DSP01040  
 DSP01050  
 DSP01060  
 DSP01070  
 DSP01080  
 DSP01090  
 DSP01100  
 DSP01110  
 DSP01120  
 DSP01130  
 DSP01140  
 DSP01150  
 DSP01160  
 DSP01170  
 DSP01180  
 DSP01190  
 DSP01200  
 DSP01210  
 DSP01220  
 DSP01230  
 DSP01240  
 DSP01250  
 DSP01260  
 DSP01270  
 DSP01280  
 DSP01290  
 DSP01300  
 DSP01310  
 DSP01320  
 DSP01330  
 DSP01340  
 DSP01350  
 DSP01360  
 DSP01370  
 DSP01380  
 DSP01390  
 DSP01400  
 DSP01410  
 DSP01420  
 DSP01430  
 DSP01440  
 DSP01450  
 DSP01460  
 DSP01470  
 DSP01480  
 DSP01490  
 DSP01500  
 DSP01510  
 DSP01520

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE DSPLY2

```
91 READ(MAPTAP) ILINE(J),(IR(I,J),I=1,PTS),(VR(I),I=1,PTS) DSP01530
IF(ILINE(J).EQ.0)GO TO 105 DSP01540
C** DSP01550
IF DESIGNATED FIELDS HAVE BEEN INPUT, SET THE IR ARRAY FOR PIXELS DSP01560
IN THOSE FIELDS. DSP01570
C** DSP01580
IF(DESKEY.EQ.1)CALL DESIG(ILINE(J),IR(I,J),DESSAV,DESFLD,DESVER, DSP01590
* NOFLD4,SAMSTR,SAMEND,SAMINC) DSP01600
DO 100 I=1,PTS DSP01610
L = IR(I,J) DSP01620
IF(L.EQ.0)GO TO 100 DSP01630
IF(L.EQ.NOSUB3)GO TO 92 DSP01640
IF(L.GT.NOSUB3)GO TO 95 DSP01650
C** DSP01660
THRESHOLDING DSP01670
C** DSP01680
THE VALUE OF THE QUADRATIC FORM MUST BE EXTRACTED FROM VR FOR THRE DSP01690
C** DSP01700
Q = -2*VR - CON DSP01710
C** DSP01720
IF(THRSKY.EQ.0)GO TO 95 DSP01730
IF((-2.*VR(I)-CON(L)).LT. THRES(L))GO TO 95 DSP01740
C** DSP01750
92 CONTINUE DSP01760
C** DSP01770
TTOL = TOTAL PIXELS THRESHOLDED, BY SUBCLASS DSP01780
C** DSP01790
TTOL(L) = TTOL(L) + 1 DSP01800
IR(I,J) = NOSUB3 DSP01810
L = NOSUB3 DSP01820
C** DSP01830
TOTALS = TOTAL NO. PIXELS CLASSIFIED INTO EACH SUBCLASS, DSP01840
INCLUDING THRESHOLDED AND DESIGNATED OTHER, DESIGNATED UNIDENT DSP01850
C** DSP01860
95 TOTALS(L) = TOTALS(L) + 1 DSP01870
100 CONTINUE DSP01880
C** DSP01890
HAVE 3 LINES BEEN READ DSP01900
C** DSP01910
IF(START)GO TO 105 DSP01920
J=J+1 DSP01930
IF(J.LT.3)GO TO 91 DSP01940
START=.TRUE. DSP01950
GO TO 91 DSP01960
C** DSP01970
SPATIAL FILTERING DSP01980
C** DSP01990
105 IF(FILTER.EQ.0)GO TO 115 DSP02000
I=2 DSP02010
106 IF(IR(I-1,I2) .NE. IR(I+1,I2)) GO TO 110 DSP02020
IF(IR(I,I1) .NE. IR(I,I3)) GO TO 110 DSP02030
IF(IR(I,I1) .NE. IR(I-1,I2)) GO TO 110 DSP02040
IF(IR(I,I2) .EQ. IR(I,I1))GO TO 110 DSP02050
IF(IR(I,I1) .EQ. NOSUB3)GO TO 110 DSP02060
ICC=IR(I,I1) DSP02070
ICK=IR(I,I2) DSP02080
IF(ICK.EQ.NOSUB3)GO TO 110 DSP02090
TOTALS(ICC)=TOTALS(ICC)+1 DSP02100
TOTALS(ICK)=TOTALS(ICK)-1 DSP02110
SPKNT=SPKNT+1 DSP02120
IR(I,I2) = IR(I,I1) DSP02130
I=I+1 DSP02140
110 IF(I.LE.PTS-1)GO TO 106 DSP02150
C** DSP02160
GET PERFORMANCE FOR LINE II DSP02170
C** DSP02180
CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING DSP02190
C** DSP02200
IF (DOTKEY.EQ.0) GO TO 40 DSP02210
115 CONTINUE DSP02220
116 DSP02230
C** DSP02240
TEST TO SEE IF THE CURRENT LINE CONTAINS ANY DOTS DSP02250
IF (ILINE(II).GT.TRNVER(2,NOFLD2)) GO TO 114 DSP02260
BCNT = 0 DSP02270
ECNT = 0 DSP02280
```

ORIGINAL PAGE IS  
OF POOR QUALITY

12-13  
192



FILE DSPLY2

```

DO 41 I=1,NOFLD2
IF (ILINE(I1).NE.TRNVER(2,I)) GO TO 42
IF (RCNT.EQ.0) RCNT = I
GO TO 41
42 IF (RCNT.EQ.0) GO TO 41
ECNT = I-1
GO TO 43
41 CONTINUE
IF (ILINE(I1).NE.TRNVER(2,NOFLD2)) GO TO 114
ECNT = NOFLD2
C*
C* FOR DOT DATA PROCESSING, CALL THE INTERNAL SUBROUTINE DOTPCT TO
C* BUILD THE CLASSIFICATION PERFORMANCE TABLE (PCTABD)
C*
43 GO TO 432
40 IF (TSTKEY.EQ.0)CALL PCT(ILINE(I1),IR(1,I1),TPNFLD,TRNVER,
* TRNSAV,PCTAB,NOFLD2,SAMSTR,SAMEND,SAMINC)
114 IF (TSTKEY.EQ.1)CALL PCT(ILINE(I1),IR(1,I1),TSTFLD,ISTVER,
* TSTSAV,PCTAB,NOFLD3,SAMSTR,SAMEND,SAMINC)
IF (NOMAP.EQ.0) GO TO 135
C*
C** CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
C*
C* IF (DOTKEY.NE.0) GO TO 117
C*
C* OUTLINE TRAINING AND/OR TEST FIELDS
C*
* IF (TRNKEY.EQ.1)CALL FLDBOR(TRNNO,ILINE(I1),IR(1,I1),NOFLD2,
* TRNFLD,TRNSAV,TRNVER,NOSUB3,SAMSTR,SAMEND,
* SAMINC,LININC)
117 IF (TSTKEY.EQ.1)CALL FLDBOR(TSTNO,ILINE(I1),IR(1,I1),NOFLD3,
* TSTFLD,TSTSAV,TSTVER,NOSUB3,SAMSTR,SAMEND,
* SAMINC,LININC)
C* SET UP SYMBOLS FOR THIS LINE. FIRST MAKE SURE I/O FROM LAST LINE
C* IS COMPLETED.
DO 120 I=1,PTS
L = IR(I,I1)
IF (L.EQ.0)OUT(I)=BLANK
IF (L.NE.0)OUT(I)=SYMMTX(L)
120 CONTINUE
C*
C* WRITE FIRST 110 SAMPLES ON LINE PRINTER AND THE REST ON DRUM
C*
IPTS=PTS
IF (IPTS.GT.110)IPTS=110
IF (IPTS.LE.110)GO TO 125
IPD=PTS-110
IF (FULL)GO TO 125
CALL RWRITE(ADRES,OUT(111),IPD,LSTAT)
ADRES=ADRES+IPD
DRUMLN=DRUMLN+1
IF (ADRES+IPD.LE.DRUMAD+DRMWDS)GO TO 125
FULL=.TRUE.
125 WRITE(6,240) ILINE(I1),(OUT(I),I=1,IPTS)
C*
C* IS CLASSIFICATION MAP TO BE OUTPUT IN UNIVERSAL OR LARSYS
C* FORMAT
C*
135 IF (MAPFMT .LE. 0) GO TO 200
C*
C* CHECK TO SEE IF LAST WRITE IS COMPLETED
C*
GO TO (155,160), HDREC
C*
C* WRITE HEADER RECORD
C*
155 NC = 1
LINES = 0
FEAT = 1
LSTLIN = 0
HDREC = 2
NOFILE = NOFILE + 1
CALL WRTHED(NC,FEAT,PTS,MAPFMT,MAPUNT)
C*

```

DSP02290  
 DSP02300  
 DSP02310  
 DSP02320  
 DSP02330  
 DSP02340  
 DSP02350  
 DSP02360  
 DSP02370  
 DSP02380  
 DSP02390  
 DSP02400  
 DSP02410  
 DSP02420  
 DSP02430  
 DSP02440  
 DSP02450  
 DSP02460  
 DSP02470  
 DSP02480  
 DSP02490  
 DSP02500  
 DSP02510  
 DSP02520  
 DSP02530  
 DSP02540  
 DSP02550  
 DSP02560  
 DSP02570  
 DSP02580  
 DSP02590  
 DSP02600  
 DSP02610  
 DSP02620  
 DSP02630  
 DSP02640  
 DSP02650  
 DSP02660  
 DSP02670  
 DSP02680  
 DSP02690  
 DSP02700  
 DSP02710  
 DSP02720  
 DSP02730  
 DSP02740  
 DSP02750  
 DSP02760  
 DSP02770  
 DSP02780  
 DSP02790  
 DSP02800  
 DSP02810  
 DSP02820  
 DSP02830  
 DSP02840  
 DSP02850  
 DSP02860  
 DSP02870  
 DSP02880  
 DSP02890  
 DSP02900  
 DSP02910  
 DSP02920  
 DSP02930  
 DSP02940  
 DSP02950  
 DSP02960  
 DSP02970  
 DSP02980  
 DSP02990  
 DSP03000  
 DSP03010  
 DSP03020  
 DSP03030  
 DSP03040

FILE DSPLY2

```

C*   WRITE DATA RECORD
C*
C*   LINES = LINES + 1
C*   IF (LINES.EQ. LINES) LSTLIN = -1
C*   CALL WRTLN(IR(1,1),LSTLIN)
200  CONTINUE
C*   IF (ILINE(I3).EQ.0)GO TO 201
C*
C*   SET INDICES AND GO READ NEXT LINE
C*
      J=1
      I1=1
      I2=1
      I3=J
      GO TO 91
C*
C*   LAST LINE IN THIS FIELD HAS BEEN READ. MAKE SURE LAST 2 LINES
C*   ARE PRINTED.
C*
201  LAST=LAST+1
      IF (LAST.EQ.2)GO TO 203
      I1=1
      I2=1
      GO TO 115
C*
C*   NOW FINISH PRINTING MAP FOR THIS FIELD.
C*
203  CONTINUE
      IF (MAPFMT.GT. 0) WRITE(6,220)NOFILE,FLDESC,(FORMAT(I,MAPFMT),
      * I=1,3),LINES
2200  FORMAT(///T55,'FILE NO.      - ',I6,/T55,'FIELD NAME  - ',A4,/
      * T55,'FORMAT      - ',I3A4,/T55,'NO. RECORDS - ',I6)
      IF (NOMAP.LE. 0 ) GO TO 230
C
      IF (PTS.LE.110)GO TO 230
      AD=0
      NBUFS=20
      IF (NBUFS.GT.LINES)NBUFS=LINES
      KPTS=PTS-110
      LPTS=110
      NWDTHS=KPTS/110
      IF (MOD(KPTS,110).NE.0)NWDTHS=NWDTHS+1
      LASTRC=MOD(KPTS,110)
      IF (LASTRC.EQ.0)LASTRC=110
      I=0
219  I=I+1
      IF (I.GT.NWDTHS) GO TO 220
      LINCNT=0
      WRTFG=2
      GO TO 370
221  PRTFG=2
      GO TO 510
222  CONTINUE
      IF (I.EQ.NWDTHS)LPTS=LASTRC
      ADRES=DRUMAD + AD
      DO 205 J=1,NBUFS
      CALL PREAD(ADRES,BUF(1,J),LPTS,JSTAT(J))
      ADRES=ADRES+KPTS
      LINCNT=LINCNT + 1
205  CONTINUE
      LINE=LINSTR
      IBUF=1
C*   FINISHED READING
210  IF (JSTAT(IBUF).EQ.1)GO TO 210
      WRITE(6,240)LINE,(BUF(IK,IBUF),IK-1,LPTS)
      LINE = LINE + LININC
      LINCNT=LINCNT+1
      IF (LINCNT.GT. DRUMLN)GO TO 215
      CALL PREAD(ADRES,BUF(1,IBUF),LPTS,JSTAT(IBUF))
      ADRES=ADRES + KPTS
215  IBUF=IBUF+1
      IF (IBUF.GT.NBUFS)IBUF=1
      IF (LINE.LE.LINEND)GO TO 210
      AD=AD + LPTS
      GO TO 219
220  CONTINUE

```

DSP03050  
 DSP03060  
 DSP03070  
 DSP03080  
 DSP03090  
 DSP03100  
 DSP03110  
 DSP03120  
 DSP03130  
 DSP03140  
 DSP03150  
 DSP03160  
 DSP03170  
 DSP03180  
 DSP03190  
 DSP03200  
 DSP03210  
 DSP03220  
 DSP03230  
 DSP03240  
 DSP03250  
 DSP03260  
 DSP03270  
 DSP03280  
 DSP03290  
 DSP03300  
 DSP03310  
 DSP03320  
 DSP03330  
 DSP03340  
 DSP03350  
 DSP03360  
 DSP03370  
 DSP03380  
 DSP03390  
 DSP03400  
 DSP03410  
 DSP03420  
 DSP03430  
 DSP03440  
 DSP03450  
 DSP03460  
 DSP03470  
 DSP03480  
 DSP03490  
 DSP03500  
 DSP03510  
 DSP03520  
 DSP03530  
 DSP03540  
 DSP03550  
 DSP03560  
 DSP03570  
 DSP03580  
 DSP03590  
 DSP03600  
 DSP03610  
 DSP03620  
 DSP03630  
 DSP03640  
 DSP03650  
 DSP03660  
 DSP03670  
 DSP03680  
 DSP03690  
 DSP03700  
 DSP03710  
 DSP03720  
 DSP03730  
 DSP03740  
 DSP03750  
 DSP03760  
 DSP03770  
 DSP03780  
 DSP03790  
 DSP03800

FILE DSPLY2

```

240 FORMAT(' ',15,2X,110A1) DSP03810
    IF (FILTER.EQ.0) GO TO 230 DSP03820
    WRITE (6,305) SPKNT DSP03830
305 FORMAT('/ THE CLASSIFICATION OF ',I7,' PIXELS WAS CHANGED AS A RESU DSP03840
    *LT OF SPATIAL FILTERING'// DSP03850
230 CONTINUE DSP03860
250 CONTINUE DSP03870
C** DSP03880
C** PRINT CLASSIFICATION SUMMARY FOR THIS FIELD DSP03890
C** DSP03900
    CALL SETMRG(68,4,62) DSP03910
    CALL PRSUM(TOTALS,TTOL,FLDESC) DSP03920
    IF (DOTKEY.EQ.0) GO TO 500 DSP03930
C** DSP03940
C** DOTS IN THE DESIGNATED AREA OR NOT IN THE CLASSIFIED AREA WILL DSP03950
C** HAVE PCTABD=0 DSP03960
C** DSP03970
C*** CODE ADDED NOV 13 ,1978 TO INCLUDE LIST PROCESSING DSP03980
C** DSP03990
255 CONTINUE DSP04000
C** DSP04010
C** DSP04020
    WRITE (6,5) DSP04030
    5 FORMAT ('H1) DSP04040
    DO 7 CHPCT=1,NOFLD2 DSP04050
    IF (PCTABD(CHPCT,1).LE.NOSUB3) GO TO 8 DSP04060
    WRITE (6,4) TRNVER(1,CHPCT),TRNVER(2,CHPCT) DSP04070
    4 FORMAT (/3X, 'DOT (' , I4, ' , ' , I4, ' ) IS IN THE DESIGNATED AR DSP04080
    *EA' ) DSP04090
    PCTABD(CHPCT,1) = 0 DSP04100
    GO TO 7 DSP04110
    8 F (PCTABD(CHPCT,1).GT.0) GO TO 7 DSP04120
    WRITE (6,2) TRNVER(1,CHPCT),TRNVER(2,CHPCT) DSP04130
    2 FORMAT (/3X, 'DOT (' , I4, ' , ' , I4, ' ) IS NOT IN THE CLASSIFI DSP04140
    *ED AREA' ) DSP04150
    7 CONTINUE DSP04160
    WRITE (6,5) DSP04170
C** DSP04180
C*** CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING DSP04190
C** DSP04200
    FLDCNT = FLDCNT + 1 DSP04210
    CALL LISTSM(TOTALS,TTOL, DSP04220
    * PCTABD,GTUNIT,GTFILE,AIUNIT,AIFILE, DSP04230
    * PPUNIT,PPFILE,NAMECT,ALP,FLDCNT, DSP04240
    * NOCAT,CATNAM,SUBCAT,NOFLD2,NOSUB2,SUBNAM) DSP04250
C** DSP04260
C** DSP04270
500 CONTINUE DSP04280
    GO TO 20 DSP04290
310 CONTINUE DSP04300
    RETURN DSP04310
C** DSP04320
C** DSP04330
    SELF-CONTAINED SUBROUTINE TO PRINT HEADERS DSP04340
    ----- DSP04350
370 WRITE (6,HEAD) DSP04360
    WRITE (6,380) FLDESC,COATE DSP04370
380 FORMAT(' DISPLAY OF CLASSIFIED FIELD.....',A4/ DSP04380
    * CLASSIFICATION DATE.....',2A4) DSP04390
    IF (RMFLG.GT.0) WRITE (6,390) RMCOMB,(FETVC2(L),L=1,8MFET) DSP04400
    IF (RMFLG.EQ.0) WRITE (6,400) (FETVC2(L),L=1,NOFET2) DSP04410
    CALL MAPHD (NOCAT,SYMMTX,CATNAM,KATNO,CLSNAM,SUBNO,SUBNAM, DSP04420
    * CLSSUB,NOCLS2,NOSUB2,THRSKY,THRES) DSP04430
    IF (DFSKEY.EQ.1) WRITE (6,410) SYMMTX(DESUNI) DSP04440
    410 FORMAT(/5X, 'DESIGNATED FIELDS SYMBOL IS ',A1/) DSP04450
    390 FORMAT(12X, 'CLASSIFICATION CHANNELS... ',I2, ' LINEAR COMBINATIONS DSP04460
    * CHANNELS' /38X, 30I3) DSP04470
    400 FORMAT(12X, 'CLASSIFICATION CHANNELS... ',30I3) DSP04480
    GO TO (31,221),WRTEG DSP04490
    ----- DSP04500
C** DSP04510
C** DSP04520
C** DSP04530
C** INTERNAL ROUTINE TO PRINT COLUMN NUMBERS DSP04540
C** DSP04550
510 J=0 DSP04560

```

FILE DSPLY2

```

DO 50 IJ=ISTR, IEND, SAMINC
J=J+1
COL(1,J) = IJ/100
COL(2,J)=MOD(IJ,100)/10
COL(3,J)=MOD(IJ,10)
IF (J, EQ, 110) GO TO 60
50 CONTINUE
60 SAMEN=IJ
ISTR=SAMEN + SAMINC
IPTS=J
WRITE(6,96)
DO 80 IJ=1,3
80 WRITE(6,90)(COL(IJ,J), J=1, IPTS)
90 FORMAT(9X, 11011)
WRITE(6,96)
96 FORMAT(/)
GO TO (85,222), PRTFG
*****
INTERNAL SUBROUTINE DOTPCT
PURPOSE - TO SET PCTAB TO EACH DOT'S RESPECTIVE SUBCLASS
(OOR THRESHOLD) NUMBER WHEN THE DOTKEY FLAG IS ON
*****
TEST FOR THE POSITION OF THE DOT IN THE CLASSIFICATION
RECORD
IF THE DOT'S POSITION IS NOT COMPATIBLE WITH THE CLASSIFICATION
RECORD, PCTAB(JJ,1) = 0
432 IF (PCTKEY.NE.0) GO TO 430
DO 431 I=1,NOFLD2
431 PCTAB(I,1)=0
PCTKEY = 1
430 DO 411 K=HCNT, ECNT
SAMDF=TRNVER(1,K) - SAMSTR
SAMPS=SAMDF/SAMINC + 1
CHSPS=(SAMPS-1)*SAMINC
SAMDF = SAMDF + SAMSTR
CHSPS = CHSPS + SAMSTR
IF (CHSPS.NE.SAMDF) GO TO 421
PCTABD(K,1) = IR(SAMPS,11)
GO TO 411
421 PCTABD(K,1) = 0
411 CONTINUE
GO TO 114
END

```

DSP04570  
DSP04580  
DSP04590  
DSP04600  
DSP04610  
DSP04620  
DSP04630  
DSP04640  
DSP04650  
DSP04660  
DSP04670  
DSP04680  
DSP04690  
DSP04700  
DSP04710  
DSP04720  
DSP04730  
DSP04740  
DSP04750  
DSP04760  
DSP04770  
DSP04780  
DSP04790  
DSP04800  
DSP04810  
DSP04820  
DSP04830  
DSP04840  
DSP04850  
DSP04860  
DSP04870  
DSP04880  
DSP04890  
DSP04900  
DSP04910  
DSP04920  
DSP04930  
DSP04940  
DSP04950  
DSP04960  
DSP04970  
DSP04980  
DSP04990  
DSP05000  
DSP05010  
DSP05020  
DSP05030  
DSP05040  
DSP05050  
DSP05060  
DSP05070  
DSP05080

FILE: EMTHRS

```
SUBROUTINE EMTHRS(FLDSAV, FIELD, VERTEX, NOFLD) EMT00010
----- EMT00020
C THIS SUBROUTINE IS USED ONLY WHEN THE EMPIRICAL THRESHOLD OPTION EMT00030
C REQUESTED EMT00040
C OR IF HISTOGRAMS OF THE QUADRATIC FORM WERE REQUESTED WITH SOME EMT00050
C OTHER THRESHOLDS EMT00060
----- EMT00070
C EMT00080
C EMT00090
C EMT00100
C EMT00110
C EMT00120
C EMT00130
C EMT00140
C EMT00150
C EMT00160
C EMT00170
C EMT00180
C EMT00190
C EMT00200
C EMT00210
C EMT00220
C EMT00230
C EMT00240
C EMT00250
C EMT00260
C EMT00270
C EMT00280
C EMT00290
C EMT00300
C EMT00310
C EMT00320
C EMT00330
C EMT00340
C EMT00350
C EMT00360
C EMT00370
C EMT00380
C EMT00390
C EMT00400
C EMT00410
C EMT00420
C EMT00430
C EMT00440
C EMT00450
C EMT00460
C EMT00470
C EMT00480
C EMT00490
C EMT00500
C EMT00510
C EMT00520
C EMT00530
C EMT00540
C EMT00550
C EMT00560
C EMT00570
C EMT00580
C EMT00590
C EMT00600
C EMT00610
C EMT00620
C EMT00630
C EMT00640

IMPLICIT INTEGER (A-Z)
REAL VR, DSFUNC, THRES, CON
INCLUDE CMHKK6.LIST
INCLUDE CMHKK10.LIST
COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, RMFILE, RMKEY,
* HISFIL, HISKEY, TRFORM, ERIPPT, ERPKEY, MAPUNT, NOFILE,
* DRUMAD, DRUMWDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVL
* .NHSTUN, NHSTFI, SCTRUN, MAPFIL
* .DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
* CRDUNT, PRTUNT, HANDIO
COMMON/DISPL./CATFLG, CATNAM(61), CLSNAM(61), SURNAM(61), SURNO(60),
* SUBCAT(60), CLSSUB(60), NOMAP, TOTVT3, NOSUR3,
* PCFDKY, ISTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,
* PLTKEY, RMFLG, HMCOMB, RMFEAT, CDATE(2),
* FLDSV2, FIELD2, VERTX2, FLDSV3, FIELD3, VERTX3, PCTID3,
* THRES(60), SYMMTX(66), HIGH(60), CON(60)
* FLDKEY, NOFLD2, NOFLD3, NOFET2, FETVC2(30)
* .NOSUH2, NOTREFD, TOTVT2, NOCLS2
* .KATNO(60), NOCAT, FILTER, MAPFMT
* .DESKY, DESUNI, DESOth, CROP, ACROP, AOTHER, ATOTAL
* .SITE(6), ANALYS(5), CAM(15), CRPKEY, KEPPTS(60)
* .DOTKEY, DOTERR
C$END
DATA RANGE/100/
DIMENSION DSFUNC(100,60), IR(1000), VR(1000), FLDINF(6), SCRAT(500)
EQUIVALENCE (SCRAT, IR)
10 READ(MAPTAP) FLDINF, PTS, LINES, FLDESC
IF (PTS.EQ.0) GO TO 40
DO 30 I=1, LINES
READ(MAPTAP) ILINE, (IR(J), J=1, PTS), (VR(J), J=1, PTS)
C HISTOGRAM VR WITHIN GROUND TRUTH FIELDS EMT00400
CALL PCTT (ILINE, IR, VR, FIELD, VERTEX, FLDSAV, DSFUNC, NOFLD,
* FLDINF(4), FLDINF(5), FLDINF(6), CON, RANGE) EMT00410
30 CONTINUE EMT00420
C READ END OF FIELD RECORD AND GO SEE IF THERE IS ANOTHER FIELD EMT00430
READ(MAPTAP) ILINE EMT00440
GO TO 10 EMT00450
C* EMT00460
C* EMT00470
C* EMT00480
C* EMT00490
C* EMT00500
C* EMT00510
C* EMT00520
C* EMT00530
C* EMT00540
C* EMT00550
C* EMT00560
C* EMT00570
C* EMT00580
C* EMT00590
C* EMT00600
C* EMT00610
C* EMT00620
C* EMT00630
C* EMT00640
ALL FIELDS ON THIS FILE HAVE BEEN PROCESSED NOW PLOT THE HISTOGRAM
40 CONTINUE
CALL DISTCV(DSFUNC, SCRAT, RANGE)
C*
C* NOW GO BACK TO BEGINNING OF THIS FILE AND POSITION TAPE OVER THE
C* FOUR HEADER RECORDS - GETTING IT READY D$PLY2
C*
REWIND MAPTAP
CALL F$HSEL(MAPTAP, 4, ISTAT)
IF (ISTAT.GT. 0) WRITE(6,500) ISTAT
500 FORMAT(' ERROR BACK SPACING MAPTAP', ISTAT = ', IS)
IF (ISTAT.GT. 0) CALL CMERR
DO 210 J=1, 4
210 READ(MAPTAP)
RETURN
END
```



FILE: FISH

```
FUNCTION FISH(F,N1,N2)
LOGICAL E1,E2,E3
E1=.FALSE.
E2=.FALSE.
E3=.FALSE.
IF (MOD(N1,2).EQ.0) E1=.TRUE.
IF (MOD(N2,2).EQ.0) E2=.TRUE.
X=N2/(N2*N1*F)
IF (.NOT.(E1.OR.E2)) GO TO 5
IF (E1.AND..NOT.E2) GO TO 1
IF (.NOT.E1.AND.E2) GO TO 2
IF (N1.LE.N2) GO TO 1
2
J=N1
N1=N2
N2=J
X=1.0-X
E3=.TRUE.
1
Y=1.0-X
FISH=0.0
H=SQRT(X**N2)
M=N1/2-1
I=-1
3
I=I+1
IF (I.GT.M) GO TO 10
FISH=FISH+H
H=(H*Y*(N2+2.*I))/(2.*(I+1.))
GO TO 3
10
IF (E3) GO TO 4
FISH=1.0-FISH
RETURN
4
J=N1
N1=N2
N2=J
RETURN
5
Y=1.0-X
H=.63661977*SQRT(X*Y)
FISH=.63661977*ARCOS(SQRT(X))
IF (N2.EQ.1) GO TO 6
M=N2-2
DO 6 I=1,M,2
FISH=FISH+H
H=H*X*(I+1)/(I+2)
6
IF (N1.EQ.1) RETURN
H=H*N2
M=N1-2
DO 7 I=1,M,2
FISH=FISH-H
H=H*Y*(N2*I)/(I+2)
7
RETURN
END
```

```
FIS00010
FIS00020
FIS00030
FIS00040
FIS00050
FIS00060
FIS00070
FIS00080
FIS00090
FIS00100
FIS00110
FIS00120
FIS00130
FIS00140
FIS00150
FIS00160
FIS00170
FIS00180
FIS00190
FIS00200
FIS00210
FIS00220
FIS00230
FIS00240
FIS00250
FIS00260
FIS00270
FIS00280
FIS00290
FIS00300
FIS00310
FIS00320
FIS00330
FIS00340
FIS00350
FIS00360
FIS00370
FIS00380
FIS00390
FIS00400
FIS00410
FIS00420
FIS00430
FIS00440
FIS00450
FIS00460
FIS00470
FIS00480
FIS00490
FIS00500
```







FILE: LISTPR

```
C THIS SUBROUTINE PRINTS LABEL LABELS
SUBROUTINE LISTPR(ISIT,DOTLAB,ITYPE,SUBLAB)
IMPLICIT INTEGER (4-7)
DATA SLASH/'/' ,BLANK/' '
DIMENSION LINE(57),SUBLAB(19,11),LINE2(19)
      DIMENSION DOTLAB(19,11,4)
      I1 = ISIT
      I2 = 4
      NPRT = 6
      WRITE(NPRT,10)
10    FORMAT(1H)
      IF ( ISIT.EQ.1) WRITE(NPRT,120)
120   FORMAT(//.50X,'APTC VS CLASSIFIED LABELS ')
      IF (ISIT.EQ.2) WRITE(NPRT,130)
130   FORMAT(//.50X,'GROUND TRUTH VS CLASSIFIED LABELS')
      IF (ISIT.EQ.3) WRITE(NPRT,230)
230   FORMAT(//.50X,'A. I. LABELS VS CLASSIFIED LABELS')
      WRITE(NPRT,20) ITYPE
20    FORMAT(//.50X,'TYPE = 1) : DOT CLASSIFICATION')
      WRITE(NPRT,30) (I,I=10,19,10)
30    FORMAT(//.12X,19(15,1X))
      DO 200 J=1,11
        DO 215 J = 1,57
          LINE(J) = BLANK
215   DO 216 J=1,19
          LINE2(J)=BLANK
216   DO 220 J=1,19
          LL=3*(J-1)
          LINE(LL+1)=DOTLAB(J,I,11)
          LINE(LL+3)=DOTLAB(J,I,12)
          IF (DOTLAB(J,I,11).NE.BLANK) LINE(LL+2) = SLASH
          LL2 = LINE(LL+2)
          IF (LL2.NF.SLASH) LINE(LL+3) = BLANK
          IF (LL2.NF.SLASH) GO TO 220
          LINE2(J)=SUBLAB(J,I)
220   CONTINUE
          I10=I*10
          WRITE(NPRT,300) I10,(LINE(KK),KK=1,57),(LINE2(KK),KK=1,19)
300   FORMAT(//.9X,13,19(1X,1A2,1A1,1A2),/.12X,19(1X,1A4,1X))
200   CONTINUE
100   CONTINUE
      RETURN
      END
```

LS00010  
LS00020  
LS00030  
LS00040  
LS00050  
LS00060  
LS00070  
LS00080  
LS00090  
LS00100  
LS00110  
LS00120  
LS00130  
LS00140  
LS00150  
LS00160  
LS00170  
LS00180  
LS00190  
LS00200  
LS00210  
LS00220  
LS00230  
LS00240  
LS00250  
LS00260  
LS00270  
LS00280  
LS00290  
LS00300  
LS00310  
LS00320  
LS00330  
LS00340  
LS00350  
LS00360  
LS00370  
LS00380  
LS00390  
LS00400  
LS00410  
LS00420  
LS00430

12-23  
302

FILE LISTSM

```

SUBROUTINE LISTSM(TOTALS,TTOL,PCTAB,GTUNIT,
* GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE,NAMECT,ALP,
* FLDINT,NCAT,CATNM,SUBCAT,NFLD2,NSUB2,SUBNAM)
  IMPLICIT INTEGER (A-Z)

C*****
C***** SUBROUTINE WRITTEN NOV 1978 TO INCLUDE LIST PROCESSING
C***** M AND R ARE BOUNDARY DOT NAMES
C***** D IS THE DESIGNATED NAME
C***** NAMECT IS THE NAME OF THE 'SMALL GRAINS' CATEGORY
C***** THE CATEGORY CLASSIFIER MUST HAVE BEEN INVOKED
C*****

COMMON /LISTMM/ NPGA(3,2),NAMPGA(209,3,2),LINPGA(209,3,2),
* SAMPGA(209,3,2),DOTLAB(209,4,2),VPGA(3),IPGA

C2345678
C***** INCLUDE CMBK14
C***** INCLUDE COMRK1
COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VAR22,TOTVT2,NOFLD2,
* AVAR2,COVAR2,CLSID2,SURNO2,SURDS2,FLDSV2,VERTX2,
* FETVC2(30),SUPVC2(75),SUHPTR(75),CLSVC2(60),
* KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
COMMON/DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),
* PRTKEY,SIZE,LACIE

CSEND
  DIMENSION ALPMSG(3),SURNAM(1),SUBLAB(209)
  DATA SYMTHR/'001',SYNDES/'1',SYMOUT/'1',
* MRCO/'M',RBCO/'R',DHCO/'D',BLANK/' ',
* ALPMSG/'PPC','GT','A1'

C*****
C***** DIMENSION TTOL(1),PCTAB(1),CATNM(1),SUBCAT(1),
* FIELDS(4,250),VERTEX(1000),INFUNT(3),INFFIL(3)

C*****
C***** REAL TOTALS(1),ALP(2)

C*****
C***** CODE ADDED TO PRINT DOT PERFORMANCE SUMMARY

REAL RS,VRS,RC,VRC,TRM1,TRM3,ALPSUM,PG,CLP
REAL SUM,BASE,DUTTOT(60),LABTOT(60),TOTCAT(60)
REAL ALPHA(15),TRM2

C*****
C***** THIS CODE ADDED TO PRINT CONFUSION MATRIX N*O*N

DIMENSION CONF(60,60),DOTNUM(209,2)

REAL PHACH(2),PIXTOT,P11,P12,PB1,PB2,P,CATTOT(61)
CONTINUE
IF (FLDINT.GT.1) GO TO 400

C*****
C***** INITIALIZE IF FIRST FIELD TO BE SUMMARIZED

DO 10 I = 1,3
DO 10 II = 1,2
10  NPGA(I,II) = 0
DO 20 I = 1,2
DO 20 II = 1,3
DO 20 III = 1,209
NAMPGA(III,II,I) = BLANK
DOTLAB(III,II,I) = BLANK
20  CONTINUE

C*****
C***** READ IN PPC GT AI FILES
C***** ASSUME TYPE 1 AND 2 ON SAME UNIT BACK-TO-BACK

IPGA = 0
IF (PPUNIT.EQ.0) GO TO 25
IPGA = IPGA + 1
VPGA(IPGA) = 1
25  IF (GTUNIT.EQ.0) GO TO 30
IPGA = IPGA + 1
VPGA(IPGA) = 2
30  IF (AIUNIT.EQ.0) GO TO 35
IPGA = IPGA + 1
VPGA(IPGA) = 3

```

000010  
000020  
000030  
000040  
000050  
000060  
000070  
000080  
000090  
000100  
000110  
000120  
000130  
000140  
000150  
000160  
000170  
000180  
000190  
000200  
000210  
000220  
000230  
000240  
000250  
000260  
000270  
000280  
000290  
000300  
000310  
000320  
000330  
000340  
000350  
000360  
000370  
000380  
000390  
000400  
000410  
000420  
000430  
000440  
000450  
000460  
000470  
000480  
000490  
000500  
000510  
000520  
000530  
000540  
000550  
000560  
000570  
000580  
000590  
000600  
000610  
000620  
000630  
000640  
000650  
000660  
000670  
000680  
000690  
000700  
000710  
000720  
000730  
000740  
000750  
000760



FILE LISTSM

```

22  CONTINUE
    ISIT = VPGA(I)
    IPT = 0
    STAMNT = 1
    TYPE = 1
    NOCAT = 0
    NOFLD2 = 0
    TOTVT2 = 0
    TOTVEC = 0
    SWCHG = 0
    INIT = 0
110  CALL LISTLC(FIELDS,STAMNT,&130,&140,&150,SWCHG,
    *  INIT,INFUNT(ISIT),INFFIL(ISIT),IPT,VERTEX)
C
C 130  NPGA(ISIT,TYPE) = NOFLD2
    NAMPGA(NOFLD2,ISIT,TYPE) = FIELDS(1,NOFLD2)
    LINPGA(NOFLD2,ISIT,TYPE) = FLDINF(1)
    SAMPGA(NOFLD2,ISIT,TYPE) = FLDINF(4)
    I10 = FLDINF(1)/10
    I110 = FLDINF(4)/10
    J = (I10 - 1)*19 + I110
    DOTLAB(J,ISIT,TYPE) = FIELDS(1,NOFLD2)
C
C  CODE ADDED TO PRINT CONFUSION MATRIX N*N
    IF (TYPE.EQ.1) GO TO 110
    DO 135 JJ=1,NOCAT
    IF (DOTLAB(J,ISIT,TYPE).EQ.CATNAM(JJ)) DOTNUM(J,1)=JJ
135  CONTINUE
    IF (DOTLAB(J,ISIT,TYPE).EQ.BLANK) GO TO 110
    SUM=SUM+1
    IF (DOTNUM(J,1).EQ.0) DOTNUM(J,1)=NOCAT+1
    GO TO 110
C
C ***   DOT TYPE CHANGE
140  NOFLD2 = 0
    NOCAT=0
    GO TO 110
C
C ***   SEND CARD IMAGE DETECTED
150  CONTINUE
C
C  CODE ADDED TO FIND CATEGORY TOTALS FOR DOT REPORT
    DO 155 JJ=1,NOCAT
    TOTCAT(JJ)=0
    DO 155 JK=1,NOCAT
    IF (CATNAM(JJ).EQ.CATNM(JK)) TOTCAT(JJ)=CATTOT(JK)
155  CONTINUE
C
C  CODE ADDED TO FIND DOT LABEL CATEGORY NUMBER
    DO 160 JJ=1,209
    IF (PCTAB(JJ).EQ.NSUB3) GO TO 160
    IF (PCTAB(JJ).EQ.NSUB6) GO TO 160
    IF (PCTAB(JJ).EQ.NSUB7) GO TO 160
    DO 160 KK=1,NOCAT
    IF (DOTLAB(JJ,4,2).EQ.CATNAM(KK)) DOTNUM(JJ,2)=KK
160  CONTINUE
    IF (NPGA(ISIT,1).EQ.0) GO TO 505
    ITYPE = 1
    CALL LISTPR(ISIT,DOTLAB,ITYPE,SUBLAB)
    505  IF (NPGA(ISIT,2).EQ.0) GO TO 600
    ITYPE = 2
    CALL LISTPR(ISIT,DOTLAB(1,1,2),ITYPE,SUBLAB)
C
C *** COMPUTE N11,N12,N21,N22,NR1,NR2 FOR TYPE 2 DOTS
    N11 = 0
    N12 = 0
    N21 = 0
    N22 = 0
    NR1 = 0

```

LIS01530  
 LIS01540  
 LIS01550  
 LIS01560  
 LIS01570  
 LIS01580  
 LIS01590  
 LIS01600  
 LIS01610  
 LIS01620  
 LIS01630  
 LIS01640  
 LIS01650  
 LIS01660  
 LIS01670  
 LIS01680  
 LIS01690  
 LIS01700  
 LIS01710  
 LIS01720  
 LIS01730  
 LIS01740  
 LIS01750  
 LIS01760  
 LIS01770  
 LIS01780  
 LIS01790  
 LIS01800  
 LIS01810  
 LIS01820  
 LIS01830  
 LIS01840  
 LIS01850  
 LIS01860  
 LIS01870  
 LIS01880  
 LIS01890  
 LIS01900  
 LIS01910  
 LIS01920  
 LIS01930  
 LIS01940  
 LIS01950  
 LIS01960  
 LIS01970  
 LIS01980  
 LIS01990  
 LIS02000  
 LIS02010  
 LIS02020  
 LIS02030  
 LIS02040  
 LIS02050  
 LIS02060  
 LIS02070  
 LIS02080  
 LIS02090  
 LIS02100  
 LIS02110  
 LIS02120  
 LIS02130  
 LIS02140  
 LIS02150  
 LIS02160  
 LIS02170  
 LIS02180  
 LIS02190  
 LIS02200  
 LIS02210  
 LIS02220  
 LIS02230  
 LIS02240  
 LIS02250  
 LIS02260  
 LIS02270  
 LIS02280

~~12-26~~  
 225

ORIGINAL PAGE  
OF POOR QUALITY

FILE LISTSM

```

      NB2 = 0
C
C
C      CODE ADDED TO PRINT CONFUSION MATRIX N*N
      DO 510 IJ=1,60
      DO 510 IK=1,60
      CONF(IK,IJ)=0
510    CONTINUE
      DO 512 IJ=1,60
      DOTTOT(IJ)=0
      LARTOT(IJ)=0
512    CONTINUE
C
C
C      IF DOT PROCESSING SKIP LIST REPORTS
      IF (NAMECT.EQ.BLANK) GO TO 582
C
      DO 580 II = 1,209
      DUM = NAMPGA(II,ISIT,2)
      IF (DUM.NE.NAMECT) GO TO 530
C
C*** DOT LABEL IS PREFERRED CATEGORY
C
      DUMS = SAMPGA(II,ISIT,2)/10
      DUML = LINPGA(II,ISIT,2)/10
      DO 515 III = 1,11
      DO 514 IIII = 1,19
      IF (IIII.NE.DUMS) GO TO 514
      IF (III.NE.DUML) GO TO 514
C*** FOUND MACHINE CLASSIFIED DOT
      J = (III - 1)*19 + IIII
      DUMA = DOTLAR(J,4,2)
      IF (DUM.EQ.DUMA) N11 = N11 + 1
      IF (DUMA.EQ.SYMPDES) GO TO 514
C23456789
      IF (DUMA.EQ.SYMTHR) GO TO 514
      IF (DUMA.EQ.SYMOUT) GO TO 514
      IF (DUM.NE.DUMA) N12 = N12 + 1
      514 CONTINUE
      515 CONTINUE
      GO TO 580
C
C
C***
C
C
C      530 IF (DUM.NE.MBCD.AND.DUM.NE.RBCD) GO TO 550
C
C
C*** THIS PIXEL WAS LABELED BOUNDARY
C
      DUMS = SAMPGA(II,ISIT,2)/10
      DUML = LINPGA(II,ISIT,2)/10
      DO 535 III = 1,11
      DO 534 IIII = 1,19
      IF (IIII.NE.DUMS) GO TO 534
      IF (III.NE.DUML) GO TO 534
      J = (III - 1)*19 + IIII
      DUMA = DOTLAR(J,4,2)
      IF (DUMA.EQ.NAMECT) N11 = N11 + 1
      IF (DUMA.EQ.SYMPDES) GO TO 534
      IF (DUMA.EQ.SYMTHR) GO TO 534
      IF (DUMA.EQ.SYMOUT) GO TO 534
      IF (DUMA.NE.NAMECT) N12 = N12 + 1
      534 CONTINUE
      535 CONTINUE
      GO TO 580
C
C
C*** THIS PIXEL IS LABELED DESIGNATED OR IS IN THE OTHER CATEGORY
C
      550 IF (DUM.EQ.DHCD) GO TO 580
C*** IT'S IN THE OTHER CATEGORY
      DUMS = SAMPGA(II,ISIT,2)/10
      DUML = LINPGA(II,ISIT,2)/10
      DO 555 III = 1,11
      DO 554 IIII = 1,19
      IF (IIII.NE.DUMS) GO TO 554
      IF (III.NE.DUML) GO TO 554
      J = (III - 1)*19 + IIII

```

LI S02290  
LI S02300  
LI S02310  
LI S02320  
LI S02330  
LI S02340  
LI S02350  
LI S02360  
LI S02370  
LI S02380  
LI S02390  
LI S02400  
LI S02410  
LI S02420  
LI S02430  
LI S02440  
LI S02450  
LI S02460  
LI S02470  
LI S02480  
LI S02490  
LI S02500  
LI S02510  
LI S02520  
LI S02530  
LI S02540  
LI S02550  
LI S02560  
LI S02570  
LI S02580  
LI S02590  
LI S02600  
LI S02610  
LI S02620  
LI S02630  
LI S02640  
LI S02650  
LI S02660  
LI S02670  
LI S02680  
LI S02690  
LI S02700  
LI S02710  
LI S02720  
LI S02730  
LI S02740  
LI S02750  
LI S02760  
LI S02770  
LI S02780  
LI S02790  
LI S02800  
LI S02810  
LI S02820  
LI S02830  
LI S02840  
LI S02850  
LI S02860  
LI S02870  
LI S02880  
LI S02890  
LI S02900  
LI S02910  
LI S02920  
LI S02930  
LI S02940  
LI S02950  
LI S02960  
LI S02970  
LI S02980  
LI S02990  
LI S03000  
LI S03010  
LI S03020  
LI S03030  
LI S03040

FILE LISTSM .

```

DUMA = DOTLAB(J,4,2)
IF (DUMA.EQ.NAMECT) N21 = N21 + 1
IF (DUMA.EQ.SYMDER) GO TO 554
IF (DUMA.EQ.SYMTHR) GO TO 554
IF (DUMA.EQ.SYMTOUT) GO TO 554
IF (DUMA.NE.NAMECT) N22 = N22 + 1
554 CONTINUE
555 CONTINUE
580 CONTINUE
C
C*** COMPUTE PROPORTION OF PREFERRED CLASS
C23456789
P11 = FLOAT(N11)/FLOAT(N11 + N21 + NB1)
P12 = FLOAT(N12)/FLOAT(N12 + N22 + NB2)
PB1 = FLOAT(NB1)/FLOAT(N11 + N21 + NB1)
PB2 = FLOAT(NB2)/FLOAT(N12 + N22 + NB2)
P = PMACH(1)*(P11 + ALP(1)*PB1)
P = P + PMACH(2)*(P12 + ALP(2)*PB2)
C
WRITE(6,990) ALPMSG(ISIT)
WRITE(6,1000)NAMECT
WRITE(6,1050) PMACH(1),PMACH(2)
WRITE(6,1005)
WRITE(6,1010)N11,N12,N21,N22
WRITE(6,1020)NB1,NB2
WRITE(6,1030)
WRITE(6,1040) P11,P12,PB1,PB2,P,ALP(1),ALP(2)
GO TO 675
C
990 FORMAT(1H1,' TYPE II DOT REPORTS FOR LIST PROCESSING', ' ',A4,
1 ' VS MACHINE CLASS')
1000 FORMAT(1H0,' PROPORTION SUMMARY FOR CATEGORY 1 = ',1A4)
1005 FORMAT(2X,'CLASS',5X,'1',5X,'2',/,2X,'LABEL')
1010 FORMAT(4X,'1',2X,2(1X,I5),/,4X,'2',2X,2(1X,I5))
1020 FORMAT(4X,'B',2X,2(1X,I5))
1030 FORMAT(1H0,' P11 P12 PB1 PB2 P ALP ')
1040 FORMAT(1H0,7(2X,F6.4))
1050 FORMAT(1H0,' P(1) = ',F7.4,' P(2) = ',F7.4)
C
C CODE ADDED TO PRINT CONFUSION MATRIX N*N
582 TOP=NOCAT+1
DO 590 JJ=1,209
LAB=DOTNUM(JJ,1)
IF(LAB.EQ.0)GO TO 590
IF(LAB.EQ.TOP)GO TO 585
LABTOT(LAB)=LABTOT(LAB)+1
585 CLS=DOTNUM(JJ,2)
IF(CLS.EQ.0)GO TO 590
DOTTOT(CLS)=DOTTOT(CLS)+1
IF(LAB.EQ.TOP)GO TO 590
CONF(LAB,CLS)=CONF(LAB,CLS)+1
590 CONTINUE
C
C WRITE CONFUSION MATRIX
C
2000 WRITE(6,2000)
FORMAT(1H1,2X,'TYPE II DOT REPORTS',/,/,2X,'CONFUSION MATRIX')
STCAT=1
ENDCAT=NOCAT
IF(ENDCAT.GT.15)ENDCAT=15
TIMES=NOCAT/15
IF(MOD(NOCAT,15).NE.0)TIMES=TIMES+1
DO 595 JJ=1,TIMES
WRITE(6,2010) (CATNAM(KK),KK=STCAT,ENDCAT)
2010 FORMAT(/,3X,'CLASS',5X,1A4,14(2X,1A4))
WRITE(6,2012)ALPMSG(ISIT)
2012 FORMAT(3X,'-----',/,/,3X,1A4,'LABEL')
DO 610 KK=1,NOCAT
WRITE(6,2020)CATNAM(KK), (CONF(KK,LL),LL=STCAT,ENDCAT)
2020 FORMAT(4X,1A4,15I6)
610 CONTINUE
STCAT=STCAT+15
ENDCAT=NOCAT
IF(ENDCAT.GT.STCAT+14)ENDCAT=STCAT+14

```

LIS03050  
LIS03060  
LIS03070  
LIS03080  
LIS03090  
LIS03100  
LIS03110  
LIS03120  
LIS03130  
LIS03140  
LIS03150  
LIS03160  
LIS03170  
LIS03180  
LIS03190  
LIS03200  
LIS03210  
LIS03220  
LIS03230  
LIS03240  
LIS03250  
LIS03260  
LIS03270  
LIS03280  
LIS03290  
LIS03300  
LIS03310  
LIS03320  
LIS03330  
LIS03340  
LIS03350  
LIS03360  
LIS03370  
LIS03380  
LIS03390  
LIS03400  
LIS03410  
LIS03420  
LIS03430  
LIS03440  
LIS03450  
LIS03460  
LIS03470  
LIS03480  
LIS03490  
LIS03500  
LIS03510  
LIS03520  
LIS03530  
LIS03540  
LIS03550  
LIS03560  
LIS03570  
LIS03580  
LIS03590  
LIS03600  
LIS03610  
LIS03620  
LIS03630  
LIS03640  
LIS03650  
LIS03660  
LIS03670  
LIS03680  
LIS03690  
LIS03700  
LIS03710  
LIS03720  
LIS03730  
LIS03740  
LIS03750  
LIS03760  
LIS03770  
LIS03780  
LIS03790  
LIS03800

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE LISTSM

```
595 CONTINUE LIS03810
C LIS03820
C CODE ADDED TO PRINT ALPHA VALUE MATRIX LIS03830
C LIS03840
WRITE(6,3000) LIS03850
3000 FORMAT(//,2X,'ALPHA VALUE MATRIX') LIS03860
STCAT=1 LIS03870
ENDCAT=NOCAT LIS03880
IF(ENDCAT.GT.15)ENDCAT=15 LIS03890
DO 720 JJ=1,TIMES LIS03900
WRITE(6,2010)(CATNAM(KK),KK=STCAT,ENDCAT) LIS03910
WRITE(6,2012)ALPMSG(ISIT) LIS03920
DO 730 KK=1,NOCAT LIS03930
DO 740 II=STCAT,ENDCAT LIS03940
IF(DOTTOT(II).EQ.0)ALPHA(II-STCAT+1)=0 LIS03950
IF(DOTTOT(II).EQ.0)GO TO 740 LIS03960
ALPHA(II-STCAT+1)=CONF(KK,II)/DOTTOT(II) LIS03970
740 CONTINUE LIS03980
L=ENDCAT-STCAT+1 LIS03990
WRITE(6,3020)CATNAM(KK), (ALPHA(II),II=1,L) LIS04000
3020 FORMAT(4X,1A,2X,15F6.3) LIS04010
730 CONTINUE LIS04020
STCAT=STCAT+15 LIS04030
ENDCAT=NOCAT LIS04040
IF(ENDCAT.GT.STCAT+14)ENDCAT=STCAT+14 LIS04050
720 CONTINUE LIS04060
C LIS04070
C CODE ADDED TO PRINT DOT PERFORMANCE SUMMARY LIS04080
C LIS04090
WRITE(6,2030) LIS04100
2030 FORMAT(//,2X,'DOT DATA PERFORMANCE SUMMARY') LIS04110
WRITE(6,2035) LIS04120
2035 FORMAT(/,5X,'CATEGORY',9X,'CLASSIFIED',10X,'BIAS CORRECTED ', LIS04130
1 'PROPORTION',12X,'RANDOM SAMPLE PROPORTION',11X, LIS04140
2 'VARIANCE') LIS04150
WRITE(6,2038) LIS04160
2038 FORMAT(7X,'NAME',12X,'ESTIMATE',10X,'ESTIMATE',10X,'VARIANCE', LIS04170
1 10X,'ESTIMATE',10X,'VARIANCE',11X,'RATIO') LIS04180
C LIS04190
BASE=PIXTOT*TOTALS(NSUB7) LIS04200
C LIS04210
C LOOP TO CALCULATE & PRINT SUMMARY BY CATEGORY LIS04220
C LIS04230
DO 650 KK=1,NOCAT LIS04240
ALPSUM=0 LIS04250
DO 625 II=1,NOCAT LIS04260
IF(DOTTOT(II).EQ.0)TRM1=0 LIS04270
IF(DOTTOT(II).EQ.0)GO TO 630 LIS04280
TRM1=CONF(KK,II)/DOTTOT(II)*TOTCAT(II) LIS04290
630 ALPSUM=ALPSUM+TRM1 LIS04300
625 CONTINUE LIS04310
RS=(LABTOT(KK)/SUM)*(PIXTOT/BASE)*100 LIS04320
VRS=RS*((PIXTOT/BASE)*100-RS)/(SUM-1) LIS04330
RC=(ALPSUM/BASE)*100 LIS04340
VRC=0 LIS04350
DO 635 LL=1,NOCAT LIS04360
TRM1=((TOTCAT(LL)/BASE)*100)**2 LIS04370
IF(DOTTOT(LL).EQ.0)TRM3=0 LIS04380
IF(DOTTOT(LL).EQ.0)GO TO 640 LIS04390
TRM2=CONF(KK,LL)/DOTTOT(LL) LIS04400
TRM3=(TRM2*(1-TRM2))/(DOTTOT(LL)-1) LIS04410
640 VRC=VRC+TRM1*TRM3 LIS04420
635 CONTINUE LIS04430
PG=VRC/VRS LIS04440
CLP=TOTCAT(KK)/PIXTOT*100 LIS04450
WRITE(6,2040)CATNAM(KK),CLP,BC,VBC,RS,VRS,PG LIS04460
2040 FORMAT(9X,1A,4X,6(6X,F8.4,4X)) LIS04470
650 CONTINUE LIS04480
675 WRITE(6,2050) LIS04490
2050 FORMAT(1H1) LIS04500
600 CONTINUE LIS04510
RETURN LIS04520
END LIS04530
```



FILE: MAPHD

```

SUBROUTINE MAPHD(NOCAT,CLSSYM,CATNAM,KATNO,CLSMTX,SURNO,
* SUBDES,CLSV2,NOCLS2,NOSUR2,THRSKY,THRES)
MAP00010
MAP00020
MAP00030
MAP00040
MAP00050
MAP00060
MAP00070
MAP00080
MAP00090
MAP00100
MAP00110
MAP00120
MAP00130
MAP00140
MAP00150
MAP00160
MAP00170
MAP00180
MAP00190
MAP00200
MAP00210
MAP00220
MAP00230
MAP00240
MAP00250
MAP00260
MAP00270
MAP00280
MAP00290
MAP00300
MAP00310
MAP00320
MAP00330
MAP00340
MAP00350
MAP00360
MAP00370
MAP00380
MAP00390
MAP00400
MAP00410
MAP00420
MAP00430
MAP00440
MAP00450
MAP00460
MAP00470
MAP00480
MAP00490
MAP00500
MAP00510
MAP00520
MAP00530
MAP00540
MAP00550
MAP00560
MAP00570
MAP00580
MAP00590
MAP00600
MAP00610
MAP00620
MAP00630
MAP00640
MAP00650
MAP00660
MAP00670
MAP00680
MAP00690
MAP00700
MAP00710
MAP00720
MAP00730
MAP00740
MAP00750
MAP00760
MAP00770
MAP00780
MAP00790

THIS ROUTINE PRINTS THE HEADER INFORMATION FOR THE CLASSIFICATION
MAP IN DISPLAY

      NOCAT -- NO. OF CATEGORIES
      CLSSYM -- SYMBOLS FOR CATEGORIES OR SURCLASSES
      CATNAM -- CATEGORY NAMES
      KATNO -- CATEGORY EACH CLASS WAS ASSIGNED TO
      CLSMTX -- CLASS NAMES
      SURNO -- NO. OF SURCLASSES IN EACH CLASS
      SUBDES -- SURCLASS NAMES
      CLSV2 -- CLASS EACH SUBCLASS WAS ASSIGNED TO (IN COMMON
      BLOCK INFORM)

      IMPLICIT INTEGER (A-Z)
      DIMENSION CLSV2(1)
      REAL THRES(1)

      LOGICAL ISWTH
      DIMENSION CLSSYM(1),CATNAM(1),KATNO(1),CLSMTX(1),SUBNO(1),
      SUBDES(1)

      PRINTS CATEGORY CLASSIFIER INFORMATION

      IF (NOCAT .LE. 0) GO TO 82
      WRITE(6,200)
200  FORMAT(// T42,'MAP OF CATEGORY CLASSIFIER CLASSIFICATION RESULTS',
* // T32,'CATEGORY',T62,'CLASS',T93,'SURCLASS',T31,
* 'NO.',T37,'NAME',
* T60,'NO.',T66,'NAME',
* T84,'NO.',T94,'NAME',T101,'SYMBOL')
      IF (THRSKY.NE.0) WRITE(6,205)
205  FORMAT(1H+,T108,'THRES.')
```

C

PRINTS STANDARD CLASSIFIER INFORMATION

FILE: MAPHD

```

R2 CONTINUE
C
  WRITE(A,260)
  260 FORMAT( // T42, 'MAP OF STANDARD CLASSIFIER CLASSIFICATION RESULT'
    *S' ///T45,'CLASS',T77,'SURCLASS' / T42,'NO.',T50,'NAME',T72,'NO.',
    *T78, 'NAME',T85,'SYMBOL')
  IF (THRSKY.NE.0) WRITE(6,265)
  265 FORMAT(1H+,T93,'THRES.')
C
  CLSNUM = 1
  ISWTH = .TRUE.
  DO A9 I=1,NOSIHK2
    IF (CLSNUM.F3. CLSVC2(I)) GO TO 85
    CLSNUM = CLSNUM + 1
    GO TO R7
  R5 IF (ISWTH) GO TO 87
  WRITE(6,270)I,SURDES(I),CLSSYM(I)
  270 FORMAT(T72,I2,T78,A4,T87,A1)
  GO TO R6
  R7 WRITE(6,280)CLSNUM,CLSMIX(CLSNUM),I,SURDES(I),CLSSYM(I)
  280 FORMAT(/ T42,I2,T50,A4,T72,I2,T78,A4,T87,A1)
  ISWTH = .FALSE.
  R8 IF (THRSKY.NE.0) WRITE(6,285)THRES(I)
  285 FORMAT(1H+,T93,F6.3)
  89 CONTINUE
  RETURN
  END
MAP00800
MAP00810
MAP00820
MAP00830
MAP00840
MAP00850
MAP00860
MAP00870
MAP00880
MAP00890
MAP00900
MAP00910
MAP00920
MAP00930
MAP00940
MAP00950
MAP00960
MAP00970
MAP00980
MAP00990
MAP01000
MAP01010
MAP01020
MAP01030
MAP01040
MAP01050
MAP01060

```

FILE: PCT

```
SUBROUTINE PCT(LINUM,IR,FIELD,VERTEX,FLOSAV,PCTAB,NOFLD,
              SAMSTR,SAMEND,SAMINC)
  IMPLICIT INTEGER(A-Z)
```

THIS SUBROUTINE BUILDS THE PERFORMANCE TABLE FOR DISPLAY, OR THE HISTOGRAM OF THE QUADRATIC FORM FOR EMPIRICAL THRESHOLDS.

ARGUMENTS:

```

  LINUM - LINE NO. BEING TESTED.
  PTS   - NO. OF POINTS IN IR ARRAY
  FIELD - RECTANGULAR COORDINATES OF FIELDS (TRAINING OR TEST)
         (5,NOFLD)
         1-LINE START
         2-LINE END
         3-SAMPLE START
         4-SAMPLE END
         5-POINTER TO VERTEX ARRAY FOR VERTICES OF THIS FIELD
  VERTEX - ARRAY CONTAINING VERTICES FOR ALL FIELDS
  FLOSAV - FIELD INFORMATION
         (4,NOFLD)
         1-FIELD NAME
         2-CLASS NO.
         3-SURCLASS NO.
         4-NO. OF VERTICES
  PCTAB - PERFORMANCE TABLE
         (NOFLD,NOFLD,NOFLD)
  NOFLD - NO. OF FIELDS TO TEST
  IR     - ARRAY CONTAINING THE SUBCLASS NUMBERS FOR POINTS
         ON THIS LINE.
  SAMSTR - BEGINNING SAMPLE NO. OF CLASSIFIED FIELD.
  SAMEND - LAST SAMPLE NO. OF CLASSIFIED FIELD.
  SAMINC - SAMPLE INCREMENT USED IN CLASSIFY
```

```

  DIMENSION FIELD(5,NOFLD),VERTEX(1),FLOSAV(4,NOFLD),
             PCTAB(NOFLD,1),IR(1)
  DIMENSION VFC(50),FL(22)
  OPT=1
  GO TO 5
  ENTRY PCTT(LINUM,IR,VR,FIELD,VERTEX,FLOSAV,DSFUNC,NOFLD,
            SAMSTR,SAMEND,SAMINC,CON,RANGE)
  PEAL DSFUNC(RANGE,50),VR(1),CON(1)
  OPT=2
```

5 CONTINUE

FIND NUMBER OF FIELDS THAT THIS LINE INTERSECTS

-----PCTT0620  
PCTT0630

```

  II=0
  DO 10 I=1,NOFLD
  IF (LINUM.LT.FIELD(1,I)) GO TO 10
  IF (LINUM.GT.FIELD(2,I)) GO TO 10
  IF (FIELD(3,I).GT.SAMEND) GO TO 10
  IF (FIELD(4,I).LT.SAMSTR) GO TO 10
  II = II+1
  VFC(II) = I
10 CONTINUE
```

PCTT0670  
PCTT0680  
PCTT0690  
PCTT0700  
PCTT0710  
PCTT0720  
PCTT0730  
PCTT0740  
PCTT0750  
PCTT0760  
PCTT0770

NOW CHECK THE FIELDS OF INTEREST ( GIVEN BY 'VFC' )

IF (II .EQ. 0) GO TO 35

PCTT0790  
PCTT0800

```

  DO 30 J=1,II
  JJ = VFC(J)
  NV=FLOSAV(4,JJ)
  IPT=FIELD(5,JJ)
```

FIND INTERCEPTS FOR THIS FIELD  
CALL FOLINT(VERTEX(IPT),NV,FL,LINUM,SAMPS,NI)

```

  DO 30 I=1,1,2
  JB = (FL(I)-SAMSTR)/SAMINC + 1
  JE = (FL(I+1)-SAMSTR)/SAMINC + 1
  IF(MOD(SAMSTR,SAMINC).NE.MOD(FL(I),SAMINC))JB=JB+1
  IF(JB.GT.JE)GO TO 30
```

PCTT0870

```

  DO 20 K=JB,JE
  K2 = IR(K)
  IF(K2.EQ.0)GO TO 20
  GO TO (15,18),OPT
```

15 PCTAB(JJ,K2) = PCTAB(JJ,K2) + 1

GO TO 20  
18 CONTINUE

FILE: PCT

```
C*
C*
C*
      DIVSN = FLDSAV(3,JJ)
      WAS THIS PIXEL CLASSIFIED CORRECTLY
21  IF(DIVSN.NE.K?)GO TO 20
      L=10*(-.5*CON(K2)-VR(K)) + 0.5
      IF (L.LE.0) L = 1
      IF (L.GT.RANGE) L = RANGE
      DSFUNC(L,DIVSN) = DSFUNC(L,DIVSN) + 1.0
20  CONTINUE
30  CONTINUE
35  CONTINUE
      RETURN
      GO HOME
      END
```

PCTT0900  
PCTT0910  
PCTT0920  
PCTT0930  
PCTT0940  
PCTT0950  
PCTT0970

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: PRTPCT

```

SUBROUTINE PRTPCT (FLDSAV, PCTAB, NOFLD)
PRTPCT PRINTS THE FOLLOWING CLASSIFICATION PERFORMANCE TABLES
1. FIELD BY SUBCLASS - IF PCFDKY=1
2. FIELD BY CLASS - IF PCFDKY=1
3. FIELD BY CATEGORY - IF PCFDKY=1 AND CATFLG=1
4. CLASS BY SURCLASS
5. CLASS BY CLASS
6. CLASS BY CATEGORY - IF CATFLG=1

IMPLICIT INTEGER (A-Z)
REAL PCTT
REAL PCT
DIMENSION FLDSAV (4, NOFLD), PCTAB (NOFLD, NOSUR3), TOTSAM (200), BUF (60)
INCLUDE CMRK6.LIST
INCLUDE CMRK10.LIST
COMMON/GLOBAL/HEAD (63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
* HISFIL, HISKEY, TRFORM, ERPTP, ERPKEY, MAPUNT, NOFILE,
* DRIMAD, DRMWDIS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
* NHSTUN, NHSTFI, SCTRUN, MAPFIL
* DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
* CRDUNT, PRTUNT, RANDIO
COMMON/DISPL/CATFLG, CATNAM (61), CLSNAM (61), SURNAM (61), SURNO (60),
* SUBCAT (60), CLSSUH (60), NOMAP, TOTVT3, NOSUR3,
* PCFDKY, ISTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,
* PLTKY, BMFLG, RMCOMB, BMFEAT, CDATE (2),
* FLDSV2, FIELD2, VERTX2, FLDSV3, FIELD3, VERTX3, PCTID3,
* THRES (60), SYMPTX (66), HIGH (60), CON (60)
* , FLDKEY, NOFLD2, NOFLD3, NOFET2, FETVC2 (30)
* , NOSUH2, NOTKFD, TOTVT2, NOCLS2
* , KATNO (60), NOCAT, FILTER, MAPFMT
* , DESKEY, DESUNI, DESOTH, CROP, ACROP, AOTHER, ATOTAL
* , SITE (6), ANALYS (5), CAM (15), CRPKEY, KEPPTS (60)
* , DOTKEY, DOTERR

CSEND
DIMENSION TABLE (60, 61)
DATA THR/'THRE'/'
SURNAM (NOSUR3) = THR
CLSNAM (NOCLS2+1) = THR
CATNAM (NOCAT+1) = THR
IF (PCFDKY.NE.1) GO TO #1
CLASSIFICATION SUMMARIES BY FIELD
SURCLASS PERFORMANCE
IR=1
IE=14
5 IF (IE.GT.NOSUR3) IE=NOSUR3
WRITE (6, HEAD)
IF (ISTKEY.EQ.1) WRITE (6, 100)
IF (ITKEY.NE.1) WRITE (6, 200)
IF (FLDKEY.EQ.1) WRITE (6, 350)
IF (FLDKEY.NE.1) WRITE (6, 351)
WRITE (6, 300) (SURNAM (I), I=IR, IE)
WRITE (6, 351)
PCTT=0.0
DO 20 J=1, NOFLD
IS=FLDSAV (3, J)
IC=FLDSAV (2, J)
IF (IS.NE.0) NAME=SURNAM (IS)
IF (IC.EQ.0) NAME=CLSNAM (IC)
TOTSAM (J)=0
DO 10 K=1, NOSUR3
10 TOTSAM (J) = TOTSAM (J) + PCTAB (J, K)
IF (FLDKEY.NE.1) GO TO 15
PCT=FLOAT (PCTAB (J, IS))/FLOAT (TOTSAM (J))*100.
PCTT=PCTT+PCT
WRITE (6, 400) FLDSAV (1, J), NAME, TOTSAM (J), PCT, (PCTAB (J, K), K=IR, IF)
GO TO 20
15 WRITE (6, 500) FLDSAV (1, J), NAME, TOTSAM (J), (PCTAB (J, K), K=IR, IE)
20 CONTINUE
IF (IE.EQ.NOSUR3) GO TO 30
IR=IR+1
IE=IE+14
PCTT=0.0
GO TO 5
30 CONTINUE
PCTT=PCTT/NOFLD

```

FILE: PRTPCT

```

IF (FLOKEY.EQ.1) WRITE(6,860) PCTT
PCTT=0.0
IR=1
IE=14
C* NOW FIELD BY CLASS
IN = NOSUR2+1
32 IF (IE.GT.NOCLS2+1) IE=NOCLS2+1
WRITE(6,HEAD)
IF (TSTKEY.NF.1) WRITE(6,600)
IF (TSTKEY.EQ.1) WRITE(6,650)
WRITE(6,351)
WRITE(6,300) (CLSNAM(I), I=IR, IE)
WRITE(6,351)
C* DO 50 J=1,NOFLD
CONDENSE A LINE
DO 33 K=1,IN
33 RUF(K)=0
DO 35 K=1,NOSUR2
IK=CLSSUR(K)
35 RUF(IK) = RUF(IK) + PCTAB(J,K)
IC=FLDSAV(2,J)
RUF(NOCLS2+1) = PCTAB(J,NOSUB3)
PCT = FLOAT(RUF(IC))/FLOAT(TOTSAM(J)) * 100.
PCTT=PCTT+PCT
WRITE(6,400) FLDSAV(1,J),CLSNAM(IC),TOTSAM(J),PCT,
(HUF(K),K=IB,IE)
50 CONTINUE
IF (IF.FQ.NOCLS2+1) GO TO 60
IR=IF+1
IE=IF+14
PCTT=0.0
GO TO 32
60 CONTINUE
PCTT=PCTT/NOFLD
WRITE(6,460) PCTT
PCTT=0.0
C*
C* NOW FIELD BY CATEGORY
C*
IF (CATFLG.EQ.0) GO TO A1
IR=1
IE=IE+14
62 IF (IE.GT.NOCAT+1) IE=NOCAT + 1
WRITE(6,HEAD)
IF (TSTKEY.NF.1) WRITE(6,700)
IF (TSTKEY.FJ.1) WRITE(6,750)
WRITE(6,352) (CATNAM(I), I=IR, IE)
WRITE(6,351)
DO 70 J=1,NOFLD
DO 63 K=1,IN
63 RUF(K)=0
DO 65 K=1,NOSUR2
IK = SURCAT(K)
65 RUF(IK) = RUF(IK) + PCTAB(J,K)
IC=FLDSAV(2,J)
ICAT=KATNO(IC)
PCT = FLOAT(RUF(ICAT))/FLOAT(TOTSAM(J)) * 100.
PCTT=PCTT+PCT
RUF(NOCAT+1) = PCTAB(J,NOSUB3)
WRITE(6,400) FLDSAV(1,J),CATNAM(ICAT),TOTSAM(J),PCT,
(RUF(K),K=IB,IF)
70 CONTINUE
IF (IE.EQ.NOCAT+1) GO TO A0
IR=IF+1
IE=IF+14
PCTT=0.0
GO TO 62
A0 CONTINUE
PCTT=PCTT/NOFLD
WRITE(6,860) PCTT
C*
C* NOW COMPRESS PCTAB TO CLASS BY SUBCLASS
C*
A1 ZERO TABLE
CONTINUE
DO 85 I=1,NOCLS2
TOTSAM(I)=0
DO 85 J=1,NOSUB3
85 TABLE(I,J)=0

```

PRT00800  
PRT00810  
PRT00820  
PRT00830  
PRT00840  
PRT00850  
PRT00860  
PRT00870  
PRT00880  
PRT00890  
PRT00900  
PRT00910  
PRT00920  
PRT00930  
PRT00940  
PRT00950  
PRT00960  
PRT00970  
PRT00980  
PRT00990  
PRT01000  
PRT01010  
PRT01020  
PRT01030  
PRT01040  
PRT01050  
PRT01060  
PRT01070  
PRT01080  
PRT01090  
PRT01100  
PRT01110  
PRT01120  
PRT01130  
PRT01140  
PRT01150  
PRT01160  
PRT01170  
PRT01180  
PRT01190  
PRT01200  
PRT01210  
PRT01220  
PRT01230  
PRT01240  
PRT01250  
PRT01260  
PRT01270  
PRT01280  
PRT01290  
PRT01300  
PRT01310  
PRT01320  
PRT01330  
PRT01340  
PRT01350  
PRT01360  
PRT01370  
PRT01380  
PRT01390  
PRT01400  
PRT01410  
PRT01420  
PRT01430  
PRT01440  
PRT01450  
PRT01460  
PRT01470  
PRT01480  
PRT01490  
PRT01500  
PRT01510  
PRT01520  
PRT01530  
PRT01540  
PRT01550  
PRT01560  
PRT01570  
PRT01580

FILE: PRTPCT

```
DO 90 I=1,NOFLD
IC=FLDSAY(2,I)
DO 90 J=1,NOSUR3
TABLE(IC,J)=TABLE(IC,J)*PCTAB(I,J)
TOTSAM(IC)=TOTSAM(IC)+PCTAB(I,J)
C* 90 CONTINUE
CLASS BY SURCLASS
IR=1
IE=14
91 IF (IF.GT.NOSUB3) IE=NOSUB3
WRITE(6,HEAD)
IF (TSTKEY.NE.1) WRITE(6,800)
IF (TSTKEY.EQ.1) WRITE(6,85)
WRITE(6,A10) (SURNAM(I),I=IR,IE)
WRITE(6,351)
DO 92 I=1,NOCLS2
IF (TOTSAM(I).EQ.0) GO TO 92
WRITE(6,820) CLSNAM(I),TOTSAM(I),(TABLE(I,J),J=IR,IE)
92 CONTINUE
IR=IR+14
IE=IE+14
IF (IE-14 .LT. NOSUR3) GO TO 91
C*
C* CLASS BY CLASS
DO 93 I=1,NOCLS2
DO 97 J=1,NOCLS2
97 RUF(J)=0
DO 94 J=1,NOSUR2
IC=CLASSU9(J)
94 RUF(IC)=RUF(IC) + TABLE(I,J)
DO 95 J=1,NOCLS2
95 TABLE(I,J)=RUF(J)
TABLE(I,NOCLS2+1)=TABLE(I,NOSUB3)
93 CONTINUE
PCTT=0
IR=1
IE=14
96 IF (IF.GT.NOCLS2+1) IE=NOCLS2+1
WRITE(6,HEAD)
IF (TSTKEY.EQ.1) WRITE(6,830)
IF (TSTKEY.NE.1) WRITE(6,835)
WRITE(6,A10) (CLSNAM(I),I=IR,IE)
WRITE(6,351)
NC=0
DO 107 I=1,NOCLS2
IF (TOTSAM(I).EQ.0) GO TO 107
NC=NC+1
PCT=(FLOAT(TABLE(I,I))/FLOAT(TOTSAM(I)))*100.
PCTT=PCTT + PCT
WRITE(6,850) CLSNAM(I),TOTSAM(I),PCT,(TABLE(I,J),J=IR,IE)
107 CONTINUE
IF (IF.EQ.NOCLS2+1) GO TO 10A
IR=IR+14
IE=IE+14
PCTT=0.0
GO TO 96
10A CONTINUE
PCTT=PCTT/NC
WRITE(6,860) PCTT
C*
C* NOW CLASS BY CATEGORY
IF (CATFLG.EQ.0) RETURN
DO 116 J=1,NOCLS2
DO 112 I=1,NOCAT
112 RUF(I)=0
DO 114 K=1,NOCLS2
IC=KATNO(K)
114 RUF(IC)=RUF(IC) + TABLE(J,K)
DO 115 K=1,NOCAT
115 TABLE(J,K)=RUF(K)
TABLE(J,NOCAT+1)=TABLE(J,NOCLS2+1)
116 CONTINUE
PCTT=0.0
IR=1
IE=14
117 IF (IF.GT.NOCAT+1) IE=NOCAT+1
WRITE(6,HEAD)
```

PRT01590  
PRT01600  
PRT01610  
PRT01620  
PRT01630  
PRT01640  
PRT01650  
PRT01660  
PRT01670  
PRT01680  
PRT01690  
PRT01700  
PRT01710  
PRT01720  
PRT01730  
PRT01740  
PRT01750  
PRT01760  
PRT01770  
PRT01780  
PRT01790  
PRT01800  
PRT01810  
PRT01820  
PRT01830  
PRT01840  
PRT01850  
PRT01860  
PRT01870  
PRT01880  
PRT01890  
PRT01900  
PRT01910  
PRT01920  
PRT01930  
PRT01940  
PRT01950  
PRT01960  
PRT01970  
PRT01980  
PRT01990  
PRT02000  
PRT02010  
PRT02020  
PRT02030  
PRT02040  
PRT02050  
PRT02060  
PRT02070  
PRT02080  
PRT02090  
PRT02100  
PRT02110  
PRT02120  
PRT02130  
PRT02140  
PRT02150  
PRT02160  
PRT02170  
PRT02180  
PRT02190  
PRT02200  
PRT02210  
PRT02220  
PRT02230  
PRT02240  
PRT02250  
PRT02260  
PRT02270  
PRT02280  
PRT02290  
PRT02300  
PRT02310  
PRT02320  
PRT02330  
PRT02340  
PRT02350  
PRT02360  
PRT02370

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: PRTPCT

```
IF (TSTKEY.EQ.1)WRITE(6,870)
IF (TSTKEY.NE.1)WRITE(6,875)
WRITE(6,910) (CATNAM(I),I=1B,IE)
WRITE(6,351)
NC=0
DO 11A I=1,NOCLS2
IF (TOTSAM(I).EQ.0)GO TO 118
NC=NC+1
IC=KATNO(I)
PCT=(FLOAT(TABLE(I,IC))/FLOAT(TOTSAM(I)))*100.
PCTT=PCTT+PCT
WRITE(6,850) CLSNAM(I),TOTSAM(I),PCT,(TABLE(I,J),J=1B,IE)
118 CONTINUE
IF (IE.GE.NOCAT+1)GO TO 119
IB=IB+14
IE=IE+14
PCTT=0.0
GO TO 117
119 CONTINUE
PCTT=PCTT/NC
WRITE(6,860) PCTT
100 FORMAT(10X,'CLASSIFICATION SUMMARY - TEST FIELDS BY SUBCLASS')
200 FORMAT(10X,'CLASSIFICATION SUMMARY - TRAINING FIELDS BY SUBCLASS')
300 FORMAT(1X,'FIELD',T17,'TOTAL',T24,'PCT',/
'1X,'NAME',T10,'CLASS',T17,'PTS',T24,'CORCT',.2X,14(1X,A6)/)
351 FORMAT(/)
352 FORMAT(1X,'FIELD',T17,'TOTAL',T24,'PCT',/
'1X,'NAME',T10,'CAT',T17,'PTS',T24,'CORCT',.2X,14(1X,A6))
350 FORMAT(1X,'SUR')
400 FORMAT(3X,A4,3X,A4,15,2X,F7.2,2X,14(15,2X))
500 FORMAT(3X,A4,3X,A4,15,12X,14(15,2X))
400 FORMAT(10X,'CLASSIFICATION SUMMARY - TRAINING FIELDS BY CLASS')
450 FORMAT(10X,'CLASSIFICATION SUMMARY - TEST FIELDS BY CLASS')
450 FORMAT(10X,'CLASSIFICATION SUMMARY - TRAINING FIELDS BY CATEGORY')
700 FORMAT(10X,'CLASSIFICATION SUMMARY - TEST FIELDS BY CATEGORY')
750 FORMAT(10X,'CLASSIFICATION SUMMARY - TRAINING CLASS BY SUBCLASS')
800 FORMAT(10X,'CLASSIFICATION SUMMARY - TRAINING CLASS BY SUBCLASS')
810 FORMAT(10X,'TOTAL PCT',/1X,'CLASS PTS. CORCT',
'.5X,14(3X,A4)/)
820 FORMAT(3X,A4,3X,15,15X,14(15,2X))
830 FORMAT(10X,'CLASSIFICATION SUMMARY - TEST CLASS BY CLASS')
830 FORMAT(10X,'CLASSIFICATION SUMMARY - TRAINING CLASS BY CLASS')
835 FORMAT(10X,'CLASSIFICATION SUMMARY - TEST CLASS BY SURCLASS')
850 FORMAT(3X,A4,3X,15,3X,F7.2,5X,14(15,2X))
851 FORMAT(10X,'CLASSIFICATION SUMMARY - TEST CLASS BY SURCLASS')
860 FORMAT(10X,'OVERALL PERFORMANCE =',F10.2)
870 FORMAT(10X,'CLASSIFICATION SUMMARY - TEST CLASS BY CATEGORY')
875 FORMAT(10X,'CLASSIFICATION SUMMARY - TRAINING CLASS BY CATEGORY')
RETURN
END
```

PRT023A0  
PRT02390  
PRT02400  
PRT02410  
PRT02420  
PRT02430  
PRT02440  
PRT02450  
PRT02460  
PRT02470  
PRT02480  
PRT02490  
PRT02500  
PRT02510  
PRT02520  
PRT02530  
PRT02540  
PRT02550  
PRT02560  
PRT02570  
PRT02580  
PRT02590  
PRT02600  
PRT02610  
PRT02620  
PRT02630  
PRT02640  
PRT02650  
PRT02660  
PRT02670  
PRT02680  
PRT02690  
PRT02700  
PRT02710  
PRT02720  
PRT02730  
PRT02740  
PRT02750  
PRT02760  
PRT02770  
PRT02780  
PRT02790  
PRT02800  
PRT02810  
PRT02820  
PRT02830  
PRT02840  
PRT02850



FILE: PRISUM

```

SUBROUTINE PRISUM(TOTALS,TTOL,FLDESC)
INCLUDE CMRK10.LIST
INCLUDE COMRKA.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
    HTSFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE,
    DRUMAD,DRMWOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
    .NMSTLN,NMSTFI,SCRUN,MAPFIL
    .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCMUNT,
    CRJUNT,PRJUNT,RANDIO
COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SURN0(60),
    SURCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,
    PCFKY,ISTKEY,TRNKEY,TRMSKY,STATKY,EMPTRS,TRMSVA,
    PLTKEY,BMFLG,BMCOMB,BMFEAT,COATE(2),
    FLOSV2,FIELD2,VEHTX2,FLOSV3,FIELD3,VEHTX3,PCTID3,
    THRS(60),SYMTEX(66),HIGH(60),CON(60)
    .FLDKEY,NOFLD2,NOFLD3,NOFET2,FETVC2(30)
    .NOSUR2,NOTHFD,TOTVT2,NOCLS2
    .KAINO(60),NOCAT,FILTER,MAPFMT
    .DESKEY,DESUNI,DESOTH,CROP,ACROP,OTHER,ATOTAL
    .SIF(A),ANALYS(5),CAM(15),CRPKEY,KEPPTS(60)
    .NOTKEY,DOTERR
C$END
DIMENSION TOTALS(66),TTOL(66)
INTEGER G, CRPTYP,F,M,Z,Y
INTEGER TTOL,DESUNI,DESOTH
INTEGER TIME,CLSSUB,CATFLG,SUBCAT
INTEGER CRPKEY,CROP,CATNAM,CLSNAM,SUBNAM
C*
C* IF INTENSIVE TEST SITE SUMMARY REPORT IS TO BE PRINTED, FIND A
C* MATCH ON 'CROP' NAME.
C*
CRPTYP=0
IF(CRPKEY.NE.1)GO TO 10
DO 2 I=1,NOCAT
IF(CROP.EQ.CATNAM(I))GO TO 6
2 CONTINUE
DO 3 I=1,NOCLS2
IF(CROP.EQ.CLSNAM(I))GO TO 7
3 CONTINUE
DO 4 I=1,NOSUR2
IF(CROP.EQ.SURNAM(I))GO TO 8
4 CONTINUE
WRITE(A,490)CROP
490 FORMAT(' THE CROP NAME ',A4,' DOES NOT MATCH A CATEGORY,CLASS OR
SUBCLASS NAME.'/) THE INTENSIVE TEST SITE SUMMARY REPORT CANNOT BE
PRINTED')
CRPKEY=0
GO TO 10
C*
C* CROP IS A CATEGORY      CRPTYP=1
C* CROP IS A CLASS        CRPTYP=2
C* CROP IS A SUBCLASS     CRPTYP=3
C*
6 INDEX=1
CRPTYP=1
GO TO 10
7 INDEX=1
CRPTYP=2
GO TO 10
8 INDEX=1
CRPTYP=3
10 CONTINUE
G=0
M=0
I=0
J=0
JJ=0
JJ=0
DO 20 I=1,66
20 JJ=TOTALS(I) * JJ
C*
C* PRINT CLASSIFICATION SUMMARY FOR THIS FIELD
C*
CALL SFTMHG(68,4,62)
WRITE(A,HEAD)
WRITE(A,260)FLDESC,JJ
J=JJ - TOTALS(DESUNI)
KT = TOTALS(DESUNI)
IF(J.LT. JJ)WRITE(6,265) KT, J
MT=TOTALS(DESOTH)

```

PRT00010  
PRT00020  
PRT00030  
PRT00040  
PRT00050  
PRT00060  
PRT00070  
PRT00080  
PRT00090  
PRT00100  
PRT00110  
PRT00120  
PRT00130  
PRT00140  
PRT00150  
PRT00160  
PRT00170  
PRT00180  
PRT00190  
PRT00200  
PRT00210  
PRT00220  
PRT00230  
PRT00240  
PRT00250  
PRT00260  
PRT00270  
PRT00280  
PRT00290  
PRT00300  
PRT00310  
PRT00320  
PRT00330  
PRT00340  
PRT00350  
PRT00360  
PRT00370  
PRT00380  
PRT00390  
PRT00400  
PRT00410  
PRT00420  
PRT00430  
SPRT00440  
PRT00450  
PRT00460  
PRT00470  
PRT00480  
PRT00490  
PRT00500  
PRT00510  
PRT00520  
PRT00530  
PRT00540  
PRT00550  
PRT00560  
PRT00570  
PRT00580  
PRT00590  
PRT00600  
PRT00610  
PRT00620  
PRT00630  
PRT00640  
PRT00650  
PRT00660  
PRT00670  
PRT00680  
PRT00690  
PRT00700  
PRT00710  
PRT00720  
PRT00730  
PRT00740  
PRT00750  
PRT00760  
PRT00770  
PRT00780  
PRT00790

FILE: PRTSUM

ORIGINAL PAGE IS  
OF POOR QUALITY

```
IF (MT.GT.0)WRITE(6,266)MT
WRITE(6,270)
WRITE(6,275)
WRITE(6,288)
WRITE(6,276)
DO 290 I=1,NOSUR2
ITTL=TOTALS(I)+TTOL(I)
PCTT=FLOAT(ITTL)/FLOAT(J)*100.
IT=TOTALS(I)
PIT=TOTALS(I)/FLOAT(J)*100.
PT=TTOL(I)
PC=PT/FLOAT(ITTL)*100.
PPIT=100.-PC
PT=PT/FLOAT(J)*100.
WRITE(6,280)SURNAM(I),ITTL,PCTT,IT,PIT,PPIT,TTOL(I),PT,PC
IF (CRPTYP.NE.3)GO TO 290
IF (INDEX.EQ.1)GO TO 80
M=M+ITTL
J=J+IT
GO TO 290
290 G=G+ITTL
I=I+1
CONTINUE
KTTL=TOTALS(NOSUR3)
JTTL = TOTALS(NOSUR3)-TTOL(NOSUR3)
PPCTT=TOTALS(NOSUR3)/FLOAT(J)*100.
WRITE(6,285)JTTL,TTOL(NOSUR3),KTTL,PPCTT
TIME=1
IC=NOCLS2
295 CONTINUE
WRITE(6,HEAD)
WRITE(6,260)FLDFSC,JJ
IF (J.LT.JJ)WRITE(6,265)KT,J
MT=TOTALS(NESOTH)
IF (MT.GT.0)WRITE(6,266) MT
WRITE(6,270)
IF (TIME.EQ.1)WRITE(6,286)
IF (TIME.EQ.2)WRITE(6,287)
WRITE(6,288)
WRITE(6,276)
DO 330 IJ=1,IC
IT=0
ITT=0
ITTL=0
DO 320 I=1,NOSUR2
IF (TIME.EQ.1)GO TO 315
IF (SUHCAT(I).NE.IJ)GO TO 320
GO TO 316
315 IF (CLASS(I).NE.IJ)GO TO 320
316 IT=TOTALS(I)+IT
ITTL=TTOL(I)+TOTALS(I)+ITTL
ITT=TTOL(I)+ITT
320 CONTINUE
PCTT=FLOAT(ITTL)/FLOAT(J)*100.
PIT=FLOAT(IT)/FLOAT(J)*100.
PT=FLOAT(ITT)/FLOAT(J)*100.
PPIT=FLOAT(IT)/FLOAT(ITTL)*100.
PC=100.-PPIT
IF (TIME.EQ.2)GO TO 325
WRITE(6,240)CLSNAM(IJ),ITTL,PCTT,IT,PIT,PPIT,ITT,PT,PC
IF (CRPTYP.NE.2)GO TO 326
IF (INDEX.EQ.1)GO TO 323
M=M+ITTL
J=J+IT
GO TO 326
323 G=G+ITTL
I=I+1
GO TO 324
325 WRITE(6,240)CATNAM(IJ),ITTL,PCTT,IT,PIT,PPIT,ITT,PT,PC
IF (CRPTYP.NE.1)GO TO 326
IF (INDEX.EQ.1)GO TO 324
M=M+ITTL
J=J+IT
GO TO 326
324 G=G+ITTL
I=I+1
326 CONTINUE
330 CONTINUE
WRITE(6,285)JTTL,TTOL(NOSUR3),KTTL,PPCTT
```

PRT00800  
PRT00810  
PRT00820  
PRT00830  
PRT00840  
PRT00850  
PRT00860  
PRT00870  
PRT00880  
PRT00890  
PRT00900  
PRT00910  
PRT00920  
PRT00930  
PRT00940  
PRT00950  
PRT00960  
PRT00970  
PRT00980  
PRT00990  
PRT01000  
PRT01010  
PRT01020  
PRT01030  
PRT01040  
PRT01050  
PRT01060  
PRT01070  
PRT01080  
PRT01090  
PRT01100  
PRT01110  
PRT01120  
PRT01130  
PRT01140  
PRT01150  
PRT01160  
PRT01170  
PRT01180  
PRT01190  
PRT01200  
PRT01210  
PRT01220  
PRT01230  
PRT01240  
PRT01250  
PRT01260  
PRT01270  
PRT01280  
PRT01290  
PRT01300  
PRT01310  
PRT01320  
PRT01330  
PRT01340  
PRT01350  
PRT01360  
PRT01370  
PRT01380  
PRT01390  
PRT01400  
PRT01410  
PRT01420  
PRT01430  
PRT01440  
PRT01450  
PRT01460  
PRT01470  
PRT01480  
PRT01490  
PRT01500  
PRT01510  
PRT01520  
PRT01530  
PRT01540  
PRT01550  
PRT01560  
PRT01570  
PRT01580

FILE: PRTSUM

```
IF (TIME.EQ.2) GO TO 340
IF (CATFLG.EQ.0) GO TO 340
IC=NOCAT
TIME=2
GO TO 295
340 CONTINUE
IF (CRPKEY.NE.1) RETURN
WRITE (6,500) CHOP,SITE,ANALYS
WRITE (6,505) CAM
D= ACROP/ATOTAL
F= AOTHER/ATOTAL
WRITE (6,510) CROP,ACROP,CROP,D, AOTHER,E, ATOTAL
H=H+TOTALS (DESOTH)
J1=J1+TOTALS (DESOTH)
Z=TOTALS (DESOTH)
Y=JJ
F=Y-Z
WRITE (6,520) Y,Z,F,CROP,G,H,CROP,I1,J1
K=G-I1
L=H-J1
M=K+L
WRITE (6,530) CROP,K,L,M
RN=FLOAT(K)/FLOAT(G)
O =FLOAT(L)/FLOAT(H)
P =FLOAT(K)/FLOAT(F)
Q =FLOAT(L)/FLOAT(F)
WRITE (6,540) CROP,RN,O,CROP,P,Q
R=FLOAT(K+L)/FLOAT(F)
S=FLOAT(G)/FLOAT(F)
T=FLOAT(H)/FLOAT(F)
U=FLOAT(I1)/FLOAT(F)
V=FLOAT(J1)/FLOAT(F)
W=FLOAT(H+K)/FLOAT(F)
WRITE (6,550) R,CROP,S,T,CROP,U,V,W
WRITE (6,610)
WRITE (6,610)
WRITE (6,560)
WRITE (6,575)
WRITE (6,570)
WRITE (6,580)
WRITE (6,570)
WRITE (6,585)
WRITE (6,570)
WRITE (6,590)
WRITE (6,570)
WRITE (6,560)
WRITE (6,595)
WRITE (6,570)
WRITE (6,611)
WRITE (6,570)
T1=S-O
T2=U-O
WRITE (6,600) CROP,G,S,T1,I1,U,P,T2
WRITE (6,570)
WRITE (6,560)
WRITE (6,605)
WRITE (6,570)
WRITE (6,615)
WRITE (6,570)
DATA NAME /'CROP'/
T1=T-E
T2=W-E
KH=H+K
WRITE (6,600) NAME,H,T,T1,KH,W,O,T2
WRITE (6,570)
WRITE (6,570)
WRITE (6,560)
260 FORMAT (/' CLASSIFICATION SUMMARY FOR FIELD ',A6//
* ' TOTAL NUMBER OF SAMPLED POINTS ',I10)
265 FORMAT (/' LESS DESIGNATED UNIDENTIFIABLE ',I10/T36,7('-'//
* T33,I1))
266 FORMAT (/' NO. OF PIXELS DESIGNATED OTHER',T33,I10//)
500 FORMAT (I1),T44,' INTENSIVE TEST SITE SUMMARY REPORT FOR ',A4 ///
* T20,' NAME OF INTENSIVE TEST SITE ',A4,T71,' NAME OF ANALYST ',
* SA4/T42,22('-' ),T86,18('-' ))
505 FORMAT (T20,' PROCEDURE CONFIGURATION',T48 , 15A4/T48,56('-' ))
510 FORMAT (T15,' GROUND TRUTH FOR INTENSIVE TEST SITE' /T15,36('-' ))
* T20,' ACHIEVE OF ',A4, ' A = ',F6.1,T62
* TRUE PROPORTION IN ',A4,5X,' A/C = D = ',F4.3 /
```

PRT01590  
PRT01600  
PRT01610  
PRT01620  
PRT01630  
PRT01640  
PRT01650  
PRT01660  
PRT01670  
PRT01680  
PRT01690  
PRT01700  
PRT01710  
PRT01720  
PRT01730  
PRT01740  
PRT01750  
PRT01760  
PRT01770  
PRT01780  
PRT01790  
PRT01800  
PRT01810  
PRT01820  
PRT01830  
PRT01840  
PRT01850  
PRT01860  
PRT01870  
PRT01880  
PRT01890  
PRT01900  
PRT01910  
PRT01920  
PRT01930  
PRT01940  
PRT01950  
PRT01960  
PRT01970  
PRT01980  
PRT01990  
PRT02000  
PRT02010  
PRT02020  
PRT02030  
PRT02040  
PRT02050  
PRT02060  
PRT02070  
PRT02080  
PRT02090  
PRT02100  
PRT02110  
PRT02120  
PRT02130  
PRT02140  
PRT02150  
PRT02160  
PRT02170  
PRT02180  
PRT02190  
PRT02200  
PRT02210  
PRT02220  
PRT02230  
PRT02240  
PRT02250  
PRT02260  
PRT02270  
PRT02280  
PRT02290  
PRT02300  
PRT02310  
PRT02320  
PRT02330  
PRT02340  
PRT02350  
PRT02360  
PRT02370

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: PRYSUM

```
* T20,'ACREAGE OF OTHER R = ',F6.1,T62 , PRT02380
* 'TRUE PROPORTION IN OTHER',4X,'R/C = E = ',F4.3 / PRT02390
* T20,'TOTAL ACREAGE',T39,'C = ',F6.1(//) PRT02400
520 FORMAT(T15,'RESULTS OF COMPUTATION FOR INTENSIVE TEST SITE',/ PRT02410
* T15.46('-'//) PRT02420
* T20,'TOTAL NUMBER OF PIXELS IN INTENSIVE TEST SITE',T96,'Y = ',I6/ PRT02430
* T20,'TOTAL NO. OF PIXELS IN EXCLUSION (UNIDENTIFIABLE) AREA',T96, PRT02440
* 'Z = ',I6 / PRT02450
* T20,'TOTAL NUMBER OF PIXELS LESS PIXELS IN EXCLUSION AREA',T92, PRT02460
* 'Y-Z=F = ',I6/ PRT02470
* T20,'NUMBER OF PIXELS CLASSIFIED AS ',A4,' BEFORE THRESHOLDING', PRT02480
* T96,'G = ',I6 / PRT02490
* T20,'NUMBER OF PIXELS CLASSIFIED AS OTHER BEFORE THRESHOLDING', PRT02500
* T96,'H = ',I6 / PRT02510
* T20,'NUMBER OF PIXELS CLASSIFIED AS ',A4,' AFTER THRESHOLDING', PRT02520
* T96,'I = ',I6 / PRT02530
* T20,'NUMBER OF PIXELS CLASSIFIED AS OTHER AFTER THRESHOLDING', PRT02540
* T96,'J = ',I6 ) PRT02550
530 FORMAT(T20,'NUMBER OF PIXELS CLASSIFIED AS ',A4,' WHICH WERE THRES PRT02560
* HOLDED',T92,'G-I=K = ',I6 / PRT02570
* T20,'NUMBER OF PIXELS CLASSIFIED AS OTHER WHICH WERE THRESHOLDED', PRT02580
* T92,'H-J=L = ',I6 / PRT02590
* T20,'TOTAL NUMBER OF PIXELS THRESHOLDED',T92,'K+L=M = ',I6) PRT02600
540 FORMAT(T20,'PROPORTION OF ',A4,' PIXELS THRESHOLDED',T92,'K/G=N = ', PRT02610
* F6.3 /T20,'PROPORTION OF OTHER PIXELS THRESHOLDED',T92,'L/H=O = ', PRT02620
* F6.3 /T20,'PROPORTION OF ',A4,' PIXELS THRESHOLDED (OF THE TEST SPRT02630
* IT)',T92,'K/F=P = ',F6.3 / PRT02640
* T20,'PROPORTION OF OTHER PIXELS THRESHOLDED (OF THE TEST SITE)', PRT02650
* T92,'L/F=Q = ',F6.3 ) PRT02660
550 FORMAT(T20,'PROPORTION OF PIXELS THRESHOLDED',T88,'(K+L)/F=R = ', PRT02670
* F6.3 /T20,'PROPORTION OF ',A4,' BEFORE THRESHOLDING',T92,'G/F=S = ', PRT02680
* F6.3 /T20,'PROPORTION OF OTHER BEFORE THRESHOLDING',T92,'H/F=T = ', PRT02690
* F6.3 /T20,'PROPORTION OF ',A4,' AFTER THRESHOLDING',T92,'I/F=U = ', PRT02700
* F6.3 /T20,'PROPORTION OF OTHER AFTER THRESHOLDING',T92,'J/F=V = ', PRT02710
* F6.3 /T20,'PROPORTION OF PIXELS TO BE CONSIDERED AS OTHER',T88, PRT02720
* '(G+K)/F = ',F6.3 ) PRT02730
560 FORMAT(T10,'101('-'//) PRT02740
570 FORMAT(1H+,T10,'-',T19,'-',T31,'-',T42,'-',T57,'-',T72,'-',T82, PRT02750
* '-',T94,'-',T110,'-'//) PRT02760
575 FORMAT(T44,'COMPUTED',T96,'COMPUTED') PRT02770
580 FORMAT(T33,'PROPOR.',T44,'PROPOR. LESS',T59,'NO. OF PX.',T74, PRT02780
* 'PROPOR.',T96,'PROPOR. LESS') PRT02790
585 FORMAT(T21,'NUMBER',T33,'BEFORE',T44,'TRUE PROPOR.',T59, PRT02800
* 'CLSFY. AFTER',T74,'AFTER',T84,'PROPOR.',T96,'TRUE PROPOR.)) PRT02810
590 FORMAT(T12,'CROP',T21,'OF PIXELS',T33,'THRSH.',T44, PRT02820
* 'BEFORE THRSH.',T59,'THRESHOLD',T74,'THRSH.',T84,'THRSH.',T96, PRT02830
* 'AFTER THRSH.)) PRT02840
595 FORMAT(T25,'(G)',T35,'(S)',T47,'(S-D)',T62,'(I)', PRT02850
* T76,'(U)',T86,'(P)',T99,'(U-D)') PRT02860
600 FORMAT(T12,'14,T23,'16,T33,'F6.3,T46,'F6.3,T61,'16,T74,'F6.3,T84,'F6.3, PRT02870
* T98,'F6.3') PRT02880
605 FORMAT(T25,'(H)',T35,'(T)',T47,'(T-E)',T61,'(H+K)', PRT02890
* T76,'(W)',T86,'(Q)',T99,'(W-E)') PRT02900
610 FORMAT(/) PRT02910
611 FORMAT(/) PRT02920
615 FORMAT(T12,'OTHER') PRT02930
270 FORMAT(/T15,'PTS. BEFORE',T29,'PCT. OF',T44,'PTS. AFTER', PRT02940
* T58,'PCT. OF',T70,'PCT. OF',T87,'PTS.',T99,'PCT. OF', PRT02950
* T111,'PCT. OF',T17,'THRES.',T29,'TOTAL',T46,'THRES.', PRT02960
* T58,'TOTAL',T87,'THRES.',T99,'TOTAL') PRT02970
275 FORMAT(1H+,T5,'SUBCLASS',T70,'SUBCLASS',T111,'SUBCLASS') PRT02980
276 FORMAT(/) PRT02990
280 FORMAT(T5,'A4,T17,'16,T29,'F6.2,T46,'16,T58, PRT03000
* F6.2,T70,'F6.2,T87,'16,T99,'F6.2,T111,'F6.2) PRT03010
285 FORMAT(/T15,'PTS. THRESHOLDED IN DISPLAY',1X,I10/ PRT03020
* T15,'PTS. THRESHOLDED IN CLASSIFY',I10/ PRT03030
* T35,'TOTAL',3X,I10,T60,'PCT. = ',F6.2) PRT03040
286 FORMAT(1H+,T5,'CLASS',T70,'CLASS',T111,'CLASS') PRT03050
287 FORMAT(1H+,T5,'CATEGORY',T70,'CATEGORY',T111,'CATEGORY') PRT03060
288 FORMAT(T29,'CLSF.FLD.',T58,'CLSF.FLD.',T99,'CLSF.FLD.', PRT03070
* T111,'THRES.') PRT03080
RETURN PRT03090
END PRT03100
```

FILE REDIF3

```

SURROUTINE REDIF3(TSTSAV,TSTFLD,TSTVER,VDIM,          RED00010
* GTUNIT,GTFILE,ATUNIT,AIFILE,PPUNIT,PPFILE,      RED00020
* NAMECT,ALP,DESSAV,DESFLD,DESVR,NOFLD4)          RED00030
IMPLICIT INTEGER (A-H,O-Z)                          RED00040
C*** CODE ADDED TO INCLUDE LIST PROCESSING          RED00050
C                                                    RED00060
REAL ALP(2)                                          RED00070
C-----IRED00080
C-----IRED00090
C-----IRED00100
C-----IRED00110
C-----IRED00120
C-----IRED00130
C-----IRED00140
C-----IRED00150
C-----IRED00160
C-----IRED00170
C-----IRED00180
C-----IRED00190
C-----IRED00200
C-----IRED00210
C-----IRED00220
C-----IRED00230
C-----IRED00240
C-----IRED00250
C-----IRED00260
C-----IRED00270
C-----IRED00280
C-----IRED00290
C-----IRED00300
C-----IRED00310
C-----IRED00320
C-----IRED00330
C-----IRED00340
C-----IRED00350
C-----IRED00360
C-----IRED00370
C-----IRED00380
C-----IRED00390
C-----IRED00400
C-----IRED00410
C-----IRED00420
C-----IRED00430
C-----IRED00440
C-----IRED00450
C-----IRED00460
C-----IRED00470
C-----IRED00480
C-----IRED00490
C-----IRED00500
C-----IRED00510
C-----IRED00520
C-----IRED00530
C-----IRED00540
C-----IRED00550
C-----IRED00560
C-----IRED00570
C-----IRED00580
C-----IRED00590
C-----IRED00600
C-----IRED00610
C-----IRED00620
C-----IRED00630
C-----IRED00640
C-----IRED00650
C-----IRED00660
C-----IRED00670
C-----IRED00680
C-----IRED00690
C-----IRED00700
C-----IRED00710
C-----IRED00720
C-----IRED00730
C-----IRED00740
C-----IRED00750
C-----IRED00760

```



FILE REDIF3

```

DO 20 I=1,20
IF (OPT(I).EQ.CODE)GO TO (100,200,300,400,600,710,720,730,
* 740,760,770,800,750,780,210,211,212,230,240), I
20 CONTINUE
GOTO 1500
C
C
C
GET SYMROLS
-----
100 IF ( SYMCNT .GE. SYMMAX ) GOTO 10
SYMCNT = SYMCNT+1
SYMMTX(SYMCNT) = BLANK
M = NXTCHR(CARD2,COL)
IF ( M .EQ. BLANK ) GO TO 10
IF ( M .EQ. COMMA ) GO TO 100
SYMMTX(SYMCNT) = CARD2(COL)
110 M = NXTCHR(CARD2,COL)
IF ( M .EQ. BLANK ) GO TO 10
IF ( M .NE. COMMA ) GO TO 110
GO TO 100
C*
C*
C*
SITE NAME
200 READ(30,201)SITE
201 FORMAT(10X,6A4)
REWIND RRUNIT
GO TO 10
C
C
C
C*** CODE ADDED NOV 13,1978. TO INCLUDE LIST PROCESSING
C*** READ GT AI OR PP UNIT AND FILE NUMBERS
210 IPAT = 16
GO TO 214
211 IPAT = 17
GO TO 214
212 IPAT = 18
214 M = NXTCHR(CARD2,COL)
IF (M.EQ.BLANK) GO TO 216
IF (M.NE.LU) GO TO 215
M = FIND12(CARD2,COL,EQUVEC)
IF(M.NE.2) GO TO 216
ISTART = 0
M = NUMBER(CARD2,COL,IPATT,ISTART)
M = FIND12(CARD2,COL,EQUVEC)
IF(M.NE.2) GO TO 216
ISTART = 0
M = NUMBER(CARD2,COL,IPATTT,ISTART)
GO TO 218
215 IF(M.NE.LF) GO TO 216
M = FIND12(CARD2,COL,EQUVEC)
IF (M.NE.2) GO TO 216
ISTART = 0
M = NUMBER(CARD2,COL,IPATTT,ISTART)
M = FIND12(CARD2,COL,EQUVEC)
IF(M.NE.2) GO TO 216
ISTART = 0
M = NUMBER(CARD2,COL,IPATT,ISTART)
GO TO 218
216 WRITE(6,217) OPT(IPAT)
217 FORMAT(' ERROR ON ',A4,' CONTROL CARD ')
GO TO 10
218 IF(IPAT.NE.16) GO TO 219
GTUNIT = IPATT
GTFIL = IPATTT
GO TO 221
219 IF(IPAT.NE.17) GO TO 220
AIUNIT = IPATT
AIFIL = IPATTT
GO TO 221
220 PPUNIT = IPATT
PPFIL = IPATTT
221 DOTKEY = 1
GO TO 10
C*** SELECTED LIST CLASS NAME
230 NAMECT = NXTCHR(CARD2,COL)
IF (NAMECT.NE.BLANK) GO TO 10

```

RED01530  
RED01540  
RED01550  
RED01560  
RED01570  
RED01580  
RED01590  
RED01600  
RED01610  
RED01620  
RED01630  
RED01640  
RED01650  
RED01660  
RED01670  
RED01680  
RED01690  
RED01700  
RED01710  
RED01720  
RED01730  
RED01740  
RED01750  
RED01760  
RED01770  
RED01780  
RED01790  
RED01800  
RED01810  
RED01820  
RED01830  
RED01840  
RED01850  
RED01860  
RED01870  
RED01880  
RED01890  
RED01900  
RED01910  
RED01920  
RED01930  
RED01940  
RED01950  
RED01960  
RED01970  
RED01980  
RED01990  
RED02000  
RED02010  
RED02020  
RED02030  
RED02040  
RED02050  
RED02060  
RED02070  
RED02080  
RED02090  
RED02100  
RED02110  
RED02120  
RED02130  
RED02140  
RED02150  
RED02160  
RED02170  
RED02180  
RED02190  
RED02200  
RED02210  
RED02220  
RED02230  
RED02240  
RED02250  
RED02260  
RED02270  
RED02280

FILE REDIF3

```

      WRITE(6,235)
235   FORMAT(' NO NAME APPEARS ON SELECTED CATEGORY CARD ')
      NAMECT = LS
      GO TO 10
C***  VALUES FOR ALPHA IN BIAS CORRECTION
C
240   M = FLTNUM(CARD2,COL,ALP,2)
      IF (M.EQ.2) GO TO 10
      WRITE(6,245)
245   FORMAT(' ERROR ON ALPHA CARD, DEFAULTING TO ZERO ')
      ALP(1) = 0.
      ALP(2) = 0.
      GO TO 10
C
C*****
C
C          ORIGINAL PAGE IS
C          OF POOR QUALITY
C
C          SCAN OPTION CARD
C-----
300  M = NXTCHR(CARD2,COL)
      DO 310 I=1,9
      IF (M.EQ.CODECD(I))GO TO (10,340,360,380,385,395,
      * 397,398,399),I
310  CONTINUE
315  WRITE(6,3152) CODE, CARD2
3152 FORMAT(//// 5X, '**** DSPLAY/REDIF3 --- ERROR IN 'OPTION' CAR
      *RD '////5X,2H',A4,6X,62A1,2H'////5X, '**** SCAN OF THIS CARD DIS
      2CONTINUED --- PROCEEDING TO NEXT CARD ***** //')
      GO TO 10
C-----
C
340  STATKY = 1
      GO TO 390
C-----
C
360  TRNKEY = 1
      GO TO 390
C-----
C
380  M=NXTCHR(CARD2,COL)
      IF (M.EQ.C)PCFDKY=1
      IF (M.EQ.L)PLTKY=1
      GO TO 390
C-----
C
385  NOMAP=0
C-----
C
390  M = FIND12(CARD2,COL,EQUCOM)
      IF (M.LE.0) GO TO 10
      GO TO 300
C
C-  -- SET THRESHOLD KEYS FOR EMPERICAL THRESHOLDING,TURN OTHERS OFF
395  THRSVA=0
      EMPTRS=2
      THRSKY=2
      GO TO 390
C-----
C
C-  -- SET THRESHOLD KEYS FOR CHI SQUARE THRESHOLDING,TURN OTHERS OFF
397  EMPTRS=0
      THRSVA=0
      THRSKY=1
      GO TO 390
C-----
C
C-  -- SET THRESHOLD KEYS FOR INPUT-VALUE THRESHOLDING,TURN OTHERS OFF
398  EMPTRS=0
      THRSVA=3
      THRSKY=3
      GO TO 390
C-  CHECK FOR FISHER OR FILTER
```



FILE REDIF3

```

399 COL=COL+1
M=NXTCHR(CARD2,COL)
IF(M.EQ.LL)GO TO 3990
IF(M.EQ.LS)GO TO 3991
GO TO 315
3990 FILTER=1
GO TO 390
C-
3991 IX=0
DO 3995 I=1,NOSUR2
IF(KEPPTS(I).GT.NOFET2)GO TO 3995
IF(IX.NE.0)GO TO 3993
WRITE(6,3992)
3992 FORMAT(1H0.////)
IX=1
3993 WRITE(6,3994) I , KEPPTS(I) , NOFET2
3994 FORMAT(' ***** FISHER THRESHOLD REQUESTED-NOT PERFORMED',/,BX,
1'... NO. SAMPLES FOR SURCLASS',I6,'(=',I6,') IS LESS THAN OR EQUAL
2 TO NUMBER OF CHANNELS (=',I6,')',/,)
3995 CONTINUE
IF(IX.EQ.0)GO TO 3996
WRITE(6,3992)
GO TO 390
C-
-- SET THRESHOLD KEYS FOR FISHER THRESHOLDING,TURN OTHERS OFF
3996 EMPTRS=0
THRSVA=0
THRSKY=4
C-
GO TO 390
C-
READ IN THRESHOLDS
-----
400 CONTINUE
I = SYMMAX-THSCNT
THSCNT = THSCNT+FLTNUM(CARD2,COL,THRES(THSCNT+1),I)
GO TO 10
C-
ANALYST NAME
600 READ(30,601)ANALYS
601 FORMAT(10X,5A4)
REWIND RRUNIT
GO TO 10
C-
COMMENT
-----
710 READ(30,999A) COMENT
999A FORMAT(10X,15A4)
REWIND RRUNIT
GOTO 10
C-
HED1
-----
720 READ(30,999B) HED1
999B FORMAT(10X,15A4)
REWIND RRUNIT
GOTO 10
C-
HED2
-----
730 READ(30,999C) HED2
999C FORMAT(10X,15A4)
REWIND RRUNIT
GOTO 10
C-
DATE
-----
740 READ(30,999D) DATE
999D FORMAT(10X,15A4)
REWIND RRUNIT
GO TO 10

```

```

RED03050
RED03060
RED03070
RED03080
RED03090
RED03100
RED03110
RED03120
RED03130
RED03140
RED03150
RED03160
RED03170
RED03180
RED03190
RED03200
RED03210
RED03220
RED03230
RED03240
RED03250
RED03260
RED03270
RED03280
RED03290
RED03300
RED03310
RED03320
RED03330
RED03340
RED03350
RED03360
RED03370
RED03380
RED03390
RED03400
RED03410
RED03420
RED03430
RED03440
RED03450
RED03460
RED03470
RED03480
RED03490
RED03500
RED03510
RED03520
RED03530
RED03540
RED03550
RED03560
RED03570
RED03580
RED03590
RED03600
RED03610
RED03620
RED03630
RED03640
RED03650
RED03660
RED03670
RED03680
RED03690
RED03700
RED03710
RED03720
RED03730
RED03740
RED03750
RED03760
RED03770
RED03780
RED03790
RED03800

```



FILE REDIF3

```

C*
  TOTVT3=0
  NOFLD3=1
  IPT=1
  NOFLD4=1
  PPT=1
840  ICK=LAREAD(TSTSAV(1,NOFLD3),TSTVER(IPT),INF,NV)
      IF(ICK.EQ.-3)GO TO 865
      IF(ICK.EQ.-2)GO TO 850
      IF(ICK.EQ.-1)GO TO 860
      IF(ICK.EQ.0)GO TO 870
      TSTSAV(2,NOFLD3)=CLSIND
      TSTSAV(3,NOFLD3)=SUBIND
      TSTSAV(4,NOFLD3)=NV
      TSTFLD(1,NOFLD3)=INF(1)
      TSTFLD(2,NOFLD3)=INF(2)
      TSTFLD(3,NOFLD3)=INF(4)
      TSTFLD(4,NOFLD3)=INF(5)
      TSTFLD(5,NOFLD3)=IPT
      IPT=IPT + 2*NV
      NOFLD3 = NOFLD3 + 1
      TOTVT3=TOTVT3+NV
      GO TO 840
C*
  SURCLASS NAME
C*
850  READ(30,851)NAME
      REWIND RRUNIT
851  FORMAT(10X,A4)
      DO 852 I=1,NOSUH2
      IF(NAME.EQ.SUBNAM(I))GO TO 854
852  CONTINUE
      WRITE(6,853)NAME
853  FORMAT(1 *ERROR ON SUBCLASS NAME CARD -.A4. DOES NOT MATCH A SUR
      *CLASS FROM THE MAPTAP FILE *)
      CALL CMERR
854  SUBIND=I
      CLSIND=CLSSUB(I)
      GO TO 840
C*
  CLASSNAME CARD
C*
860  READ(30,851)NAME
      REWIND RRUNIT
      DO 861 I=1,NOCLS2
      IF(NAME.EQ.CLSNAM(I))GO TO 863
861  CONTINUE
      WRITE(6,862)NAME
862  FORMAT(1 *ERROR ON CLASSNAME CARD -.A4. DOES NOT MATCH A CLASS NAME
      *FROM THE MAPTAP FILE *)
      CALL CMERR
863  CLSIND=I
      SUBIND=0
      GO TO 840
C*
  DESIGNATED FIELDS
C*
865  READ(30,851)TEST
      REWIND RRUNIT
      SURIND=1
      CLSIND = NOSUH3 + 4
      IF(TEST.EQ.OTHER)SUBIND=2
      IF(TEST.EQ.OTHER)CLSIND=NOSUH3+5
      IF(TEST.NE.OTHER.AND.TEST.NE.UNIDEN)GO TO 8066
8065  ICK=LAREAD(DESSAV(1,NOFLD4),DESVER(PPT),INF,NV)
      IF(ICK.EQ.-3)GO TO 865
      IF(ICK.EQ.-2)GO TO 850
      IF(ICK.EQ.-1)GO TO 860
      IF(ICK.EQ.0)GO TO 870
      DESSAV(2,NOFLD4)=CLSIND
      DESSAV(3,NOFLD4)=SUBIND
      DESSAV(4,NOFLD4)=NV
      DESFLD(1,NOFLD4)=INF(1)
      DESFLD(2,NOFLD4)=INF(2)
      DESFLD(3,NOFLD4)=INF(4)
      DESFLD(4,NOFLD4)=INF(5)
      DESFLD(5,NOFLD4)=PPT

```

RED04570  
 RED04580  
 RED04590  
 RED04600  
 RED04610  
 RED04620  
 RED04630  
 RED04640  
 RED04650  
 RED04660  
 RED04670  
 RED04680  
 RED04690  
 RED04700  
 RED04710  
 RED04720  
 RED04730  
 RED04740  
 RED04750  
 RED04760  
 RED04770  
 RED04780  
 RED04790  
 RED04800  
 RED04810  
 RED04820  
 RED04830  
 RED04840  
 RED04850  
 RED04860  
 RED04870  
 RED04880  
 RED04890  
 RED04900  
 RED04910  
 RED04920  
 RED04930  
 RED04940  
 RED04950  
 RED04960  
 RED04970  
 RED04980  
 RED04990  
 RED05000  
 RED05010  
 RED05020  
 RED05030  
 RED05040  
 RED05050  
 RED05060  
 RED05070  
 RED05080  
 RED05090  
 RED05100  
 RED05110  
 RED05120  
 RED05130  
 RED05140  
 RED05150  
 RED05160  
 RED05170  
 RED05180  
 RED05190  
 RED05200  
 RED05210  
 RED05220  
 RED05230  
 RED05240  
 RED05250  
 RED05260  
 RED05270  
 RED05280  
 RED05290  
 RED05300  
 RED05310  
 RED05320



FILE: RNORM FORTRAN A

```
FUNCTION RNORM(X)
DIMENSION A(7)
DATA A / .43063E-4, .276567E-3, .1520143E-3,
        .9270527E-2, .422820E-1, .70523E-1, 1.0 /
Y=ABS(X)/1.414213
RNORM=0.
DO 1 I=1,7
RNORM=RNORM*Y+A(I)
CALL OVERFL(INDCT) -
IF(INDCT.NE.2) GO TO 3
1 CONTINUE
RNORM=.5*(((1./RNORM)**2)**2)**2)
2 IF(X.GT.0.0) RNORM = 1.0-RNORM
RETURN
3 RNORM=0.0
GO TO 2
END
```

RN000010  
RN000020  
RN000030  
RN000040  
RN000050  
RN000060  
RN000070  
RN000080  
RN000090  
RN000100  
RN000110  
RN000120  
RN000130  
RN000140  
RN000150  
RN000160  
RN000170

FILE SETUP3

```

SUBROUTINE SETUP3 (ARRAY, TOP, GTUNIT, GTFIL,
* AIUNIT, AIFIL, PPUNIT, PPFILE, NAMECT, ALP, DESSAV,
* DESFLD, DESVER, NOFLD4, STOP)
IMPLICIT INTEGER (A-Z)
C*** CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING
C
REAL ALP (2)
LOGICAL OKAY
C
INCLUDE CUMMKA.LIST
INCLUDE CM-K10.LIST
COMMON/GLOBAL/HEAD (63), MAPTAP, DATAPE, SAVTAP, BMFILE, RMKEY,
* HISFIL, HISKEY, TRFORM, ER (PTP, EHPKEY, MAPUNT, NOFILE,
* DRUMAD, DRUMDS, PAGESZ, DATAFIL, STAFIL, ASAV, ASAVFL
* , NHSTUN, NHSTFI, SCTRUN, MAPFIL
* , DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
* CRDUNT, PNTUNT, WANDIO
COMMON/DISPL/CATFLG, CATNAM (61), CLSNAM (61), SUHNAM (61), SUBNO (60),
* SUHCAT (60), CLSSUR (60), NOMAP, TOTVT3, NOSUB3,
* PCFKEY, TSTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,
* PLTKEY, HMFLG, HMCOMR, RMFFAT, CDATE (2),
* FLDV2, FIELD2, VERTX2, FLDV3, FIELD3, VERTX3, PCTID3,
* THRS (60), SYMTX (66), HIGH (60), CON (60)
* , FLDKEY, NOFLD2, NOFLD3, NOFLT2, FETVC2 (30)
* , NOSUP2, NOTRFD, TOTVT2, NOCLS2
* , KATNO (60), NOCAT, FILTER, MAPFMT
* , DESKEY, DESUNI, DESOTH, CROP, ACROP, AOTHER, ATOTAL
* , SITE (6), ANALYS (5), CAM (15), CRPKEY, KEPPYS (60)
* , DOTKEY, DOTERR
CSEND
DATA A/'A'/.B/'NO' /
DATA BLANK/' ' /
-----
PURPOSE.. LOCATES FILE ON 'MAPTAP' AND COORDINATES
ROUTINES TO ANALYSE 'DISPLAY' CONTROL CARDS
-----
DIMENSION NUMVEC (30), FILVEC (2)
DIMENSION DESSAV (4,50), DESFLD (5,50), DESVER (1100)
DIMENSION CARD (62), ARRAY (1)
C
DIMENSION DOTCAT (62)
C
EQUIVALENCE ( DOTCAT (1) , CARD (1) )
C
DATA YBCD/ 'Y' /, NHCD/ 'N' /, FILVEC/ 1 , 'F' /
C
DIMENSION EQUVEC (2)
DATA DRUM/'DRUM' /
DATA UHCD/'U' /, FBGD/'F' /, IHCD/'I' /, EQUVEC/ 1, 'E' /
DIMENSION SLASH (2)
DATA SLASH/ 1, ' /' /
C
INIZ
-----
READ FIRST CONTROL CARD FOR MAPTAP UNIT AND FILE NUMBER

1 READ (CRDUNT, 1) CARD
  FORMAT (10X, 62A1)
  NFILE = 1
  COL = 0
  J = NATCHR (CARD, COL)
2 IF (J.EQ.BLANK) GO TO 6
  IF (J.NE.UHCD) GO TO 3
  J = FIND12 (CARD, COL, EQUVEC)

```

SET00010  
SET00020  
SET00030  
SET00040  
SET00050  
SET00060  
SET00070  
SET00080  
SET00090  
SET00100  
SET00110  
SET00120  
SET00130  
SET00140  
SET00150  
SET00160  
SET00170  
SET00180  
SET00190  
SET00200  
SET00210  
SET00220  
SET00230  
SET00240  
SET00250  
SET00260  
SET00270  
SET00280  
SET00290  
SET00300  
SET00310  
SET00320  
SET00330  
SET00340  
SET00350  
SET00360  
SET00370  
SET00380  
SET00390  
SET00400  
SET00410  
SET00420  
SET00430  
SET00440  
SET00450  
SET00460  
SET00470  
SET00480  
SET00490  
SET00500  
SET00510  
SET00520  
SET00530  
SET00540  
SET00550  
SET00560  
SET00570  
SET00580  
SET00590  
SET00600  
SET00610  
SET00620  
SET00630  
SET00640  
SET00650  
SET00660  
SET00670  
SET00680  
SET00690  
SET00700  
SET00710  
SET00720  
SET00730  
SET00740  
SET00750  
SET00760

FILE SETUP3

```

IF (J.NE.2) GO TO 6
I = NUMBER(CARD,COL,MAPTAP,ISTART)
J = FIND12(CARD,COL,EQUVEC)
IF (J.NE.2) GO TO 6
I = NUMBER(CARD,COL,NFILE,ISTART)
GO TO 10
3 IF (J.NE.IRCD) GO TO 4
J = FIND12(CARD,COL,SLASH)
IF (J.NE.2) GO TO 6
J = MATCHR(CARD,COL)
IF (J.EQ.FRCD) GO TO 5
IF (J.EQ.UBCD) GO TO 2
GO TO 6
4 IF (J.NE.FRCD) GO TO 6
5 J = FIND12(CARD,COL,EQUVEC)
IF (J.NE.2) GO TO 6
I = NUMBER(CARD,COL,NFILE,ISTART)
J = FIND12(CARD,COL,EQUVEC)
IF (J.NE.2) GO TO 6
I = NUMBER(CARD,COL,MAPTAP,ISTART)
GO TO 10
6 WRITE(6,7)
7 FORMAT(' ERROR ON MAPTAP CARD')
10 CONTINUE
IF (NFILE.LE.0) NFILE = 1
SERIAL = NFILE

COC
GET TAPE READY

40 REWIND MAPTAP
IF (NFILE.EQ.1) GO TO 50
NF = NFILE - 1
CALL FSHSFL(MAPTAP,NF,ISTAT)
IF (ISTAT.EQ.0) GO TO 50
WRITE(6,45) NF,ISTAT
45 FORMAT('/// 5X,***** DISPLAY/SETUP3 ... ERROR CONDITION ON ATTEMPT
IPT TO POSITION MAPTAP OVER',IA,3X,'FILES',// 5X,
2 '***** FSHSFL STATUS CODE =',I4,3X,'... ABORTING RUN *****/1M1)
REWIND MAPTAP
CALL CMERK

COCOC
READ MAPTAP
-----

50 CONTINUE
CATFLG=1
READ(MAPTAP) CDATE(1),CDATE(2),BMFLG,BMCOMB,BMFEAT,NOCLS2,
NOFLD2,NOSUB2,NOFT2,TOTVT2,NOCAT,VAR22,
(FETVC2(I),I=1,NOFT2)
NCAT=NOCAT
IF (NOCAT.GT.0) GO TO 55
CATFLG=0
NCAT=NOCLS2
C* SET BASE ADDRESSES FOR TRAINING FIELD INFORMATION
55 CONTINUE
NOSUB3=NOSUB2+1
FLDSV2=1
VERTX2=FLDSV2 * NOFLD2+4
FIELD2=VERTX2 * TOTVT2+2
TOP1=FIELD2 * NOFLD2+5
NV=TOTVT2+2
NF=NOFLD2+4
READ(MAPTAP,END=NO)(CATNAM(I),I=1,NCAT),(CLSNAM(I),I=1,NOCLS2),
(SUMNO(I),I=1,NOCLS2),(SUMNAM(I),I=1,NOSUB2),
(ARWAY(FLDSV2-1),I=1,NF),
(ARWAY(VERTX2-1),I=1,NV),
(SURCAT(I),I=1,NOSUB2),(CLSSUB(I),I=1,NOSUB2),
(KATNO(I),I=1,NOCLS2),(KFPPTS(I),I=1,NOSUB2)
GO TO 65
60 STOP=1
C*

```

000770  
000780  
000790  
000800  
000810  
000820  
000830  
000840  
000850  
000860  
000870  
000880  
000890  
000900  
000910  
000920  
000930  
000940  
000950  
000960  
000970  
000980  
000990  
001000  
001010  
001020  
001030  
001040  
001050  
001060  
001070  
001080  
001090  
001100  
001110  
001120  
001130  
001140  
001150  
001160  
001170  
001180  
001190  
001200  
001210  
001220  
001230  
001240  
001250  
001260  
001270  
001280  
001290  
001300  
001310  
001320  
001330  
001340  
001350  
001360  
001370  
001380  
001390  
001400  
001410  
001420  
001430  
001440  
001450  
001460  
001470  
001480  
001490  
001500  
001510  
001520

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE SETUP3

```
C* GO READ CONTROL CARDS AND TEST FIELDS
C*
C* FLOSV3=TOP1
C* FIELD3= FLOSV3 * 800
C* RESERVE ROOM FOR 200 TEST FIELDS
C* VERTX3=FIELD3 * 1000
C* VDIM = TOP - VERTX3
C* CALL REIF3 (ARRAY(FLOSV3),ARRAY(FIELD3),ARRAY(VERTX3),VDIM,
C* GTUNIT,GTFILE,ATUNIT,AIFILE,PPUNIT,PPFILE,
C* NAMECT,ALP,DESSAV,DESFLD,DESVER,NOFLD4)
C* TOP2=VERTX3+TOTV1302
C* IF (NOFLD3.LF.0) TOP2=TOP1
C* WRITE OUT SAVED TRAINING FIELDS AND TEST FIELDS
C* IF ( STUP .NE. 0 ) GOTO .200

C-----
C* PRINT OUT SUPERVISOR INFORMATION
C-----
C* IF (THRSKY .EQ. 1) GO TO 80
C* IF (EMPTRS .EQ. 2) GO TO 80
C* IF (THRSVA .EQ. 3) GO TO 80

C* TEST FOR FISHER
C* IF (THRSKY.EQ.4) GO TO 80

C*
C* NOTHRS = 4
C* CONTINUE
C* WRITE(6,819)

C*
C* WRITE(6,800)
C* A00 FORMAT('5. YOU HAVE SELECTED THE FOLLOWING OPTIONS: / ')
C*
C* CALL TAPLAR(MUNIT , MTAPE)
C* MUNIT=MAPTAP
C* MTAPE = MUNIT

C*
C* WRITE(6,801) MTAPE , MUNIT , NFILE

C*
C* A01 FORMAT( '10. PROCESS THE CLASSIFICATION RESULTS FROM MAPTAP ( ' ,
C* ' A6. ' ), UNIT' , 15. ' , FILE' , 15 )
C* IF (NOTHRS .EQ. 4) WRITE(6,803)
C* IF (THRSKY .EQ. 1) WRITE(6,804)
C* IF (EMPTRS .EQ. 2) WRITE(6,805)
C* IF (THRSVA .EQ. 3) WRITE(6,818)

C*
C* TEST FOR FISHER
C* IF (THRSKY.EQ.4) WRITE(6,817)

C*
C*
C* CONTINUE
C*
C* IF (TRNKEY.EQ.1.AND.DOTKEY.LE.0) WRITE (6,806)
C* IF ( TSTKEY .EQ. 1 ) WRITE(6,808)
C* IF ( STATKY .EQ. 1 ) WRITE(6,810)
C* IF ( PCFOKY .EQ. 1 ) WRITE(6,812)
C* IF (NOMAP.EQ.0)WRITE(6,820)
C* IF (PLTKFY.WE.0)WRITE(6,821)
C* IF (FILTER.EQ.1)WRITE(6,822)
C* IF (DESKEY.EQ.1)WRITE(6,823)
C* IF (CRPKFY.EQ.1)WRITE(6,824)CROP

C*
C* CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING
C*
C* IF (DOTKEY.EQ.0) GO TO A2
C* WRITE(6,826)
C* WRITE(6,829) GTUNIT,GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE
C* IF (NAMECT.EQ.HLANK)GO TO 82
C* WRITE(6,827)
C* WRITE(6,830) NAMECT
C* WRITE(6,831) ALP(1),ALP(2)
C* A27 FORMAT('0 LIST PROCESSING OPTION SELECTED ')
C* A28 FORMAT('///.5X. # NUMBERS OF GROUND TRUTH . AI. AND
C* . DISCRIMINATOR UNITS AND FILES ARE AS FOLLOWS ')
C* A29 FORMAT('///.5X,6(2X,15)')

SFT01530
SFT01540
SFT01550
SFT01560
SFT01570
SFT01580
SFT01590
SFT01600
SFT01610
SFT01620
SFT01630
SFT01640
SFT01650
SFT01660
SFT01670
SFT01680
SFT01690
SFT01700
SFT01710
SFT01720
SFT01730
SFT01740
SFT01750
SFT01760
SFT01770
SFT01780
SFT01790
SFT01800
SFT01810
SFT01820
SFT01830
SFT01840
SFT01850
SFT01860
SFT01870
SFT01880
SFT01890
SFT01900
SFT01910
SFT01920
SFT01930
SFT01940
SFT01950
SFT01960
SFT01970
SFT01980
SFT01990
SFT02000
SFT02010
SFT02020
SFT02030
SFT02040
SFT02050
SFT02060
SFT02070
SFT02080
SFT02090
SFT02100
SFT02110
SFT02120
SFT02130
SFT02140
SFT02150
SFT02160
SFT02170
SFT02180
SFT02190
SFT02200
SFT02210
SFT02220
SFT02230
SFT02240
SFT02250
SFT02260
SFT02270
SFT02280
```

12-53  
332



FILE SETUP3

```

830      FORMAT(///,5X,' SELECTED CATEGORY NAME FOR LIST IS ',A4)      SET02290
831      FORMAT(///,5X,' BIAS CORRECTION ALPHAS ARE ',2F10.6)      SET02295
A03  FORMAT(T10,'APPLY NO THRESHOLDING')      SET02310
A04  FORMAT(T10,'APPLY CHI SQUARE THRESHOLDS')      SET02320
805  FORMAT(T10,'APPLY EMPIRICAL THRESHOLDS')      SET02330
806  FORMAT(T10,'OUTLINE THE TRAINING FIELDS')      SET02340
80A  FORMAT(T10,'OUTLINE THE TEST FIELDS')      SET02350
810  FORMAT(T10,'PRINT OUT THE STATISTICS')      SET02360
812  FORMAT(T10,'PRINT THE GROUND TRUTH PERFORMANCE SUMMARIES BY FIELD'
      * )      SET02370
C-
A17  FORMAT(T10,'APPLY FISHER F-DISTRIBUTION THRESHOLDS')      SET02380
C-
A18  FORMAT(T10,'APPLY USER INPUT THRESHOLD VALUES')      SET02390
A19  FORMAT(///)      SET02400
A20  FORMAT(T10,'DO NOT DISPLAY A CLASSIFICATION MAP')      SET02410
A21  FORMAT(T10,'DISPLAY THE HISTOGRAMS OF THE QUADRATIC FORM FOR ALL S
      *URCLASSES')      SET02420
A22  FORMAT(T10,'PERFORM SPATIAL FILTERING')      SET02430
A23  FORMAT(T10,'EXCLUDE PIXELS IN THE DESIGNATED AREAS FROM CLASSIFICA
      *TION SUMMARIES')      SET02440
A24  FORMAT(T10,'PRINT THE INTENSIVE TEST SITE SUMMARY REPORT FOR ',A6)
      SET02450
A25  FORMAT(T10,'PRINT DOT DATA PERFORMANCE SUMMARIES FOR DOT DATA FRO
      M FORTRAN UNIT ',I3,' FILE NO. ',I3,' TAPE (OR FILE) ',A4)
      SET02460
C*
A26  FORMAT(T10,'OUTLINE THE DOTS ON THE CLASSIFICATION MAP')      SET02470
C
      82      CONTINUE      SET02480
C
      CALL WRTFLD(ARRAY(FLDSV2),ARRAY(VERTX2),NOFLD2,1,CLSNAM,SUBNAM)
      IF(NOFLD3.LE.0.AND.NOFLD4.LE.0)GO TO 85      SET02490
      IF(TSTKEY.EQ.1)IK=2      SET02500
      IF(TSTKEY.EQ.1)CALL WRTFLD(ARRAY(FLDSV3),ARRAY(VERTX3),NOFLD3,IK,
      *CLSNAM,SUBNAM)      SET02510
      IF(DESKEY.EQ.1)IK=3      SET02520
      IF(DESKEY.EQ.1)CALL WRTFLD(DESSAV,DESVER,NOFLD4,IK,CLSNAM,SUBNAM)
      SET02530
      85  CONTINUE      SET02540
C*
C*** CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING      SET02550
C
      IF (DOTKEY.EQ.0) GO TO 86      SET02560
C*
C*
      MAKE SPACE AVAILABLE IN ARRAY FOR DOT DATA INFORMATION,
      INCLUDING 1000 SCRATCH LOCATIONS FOR TEMPORARY STORAGE OF DOT
      DATA RETURNED FROM SUBR. RODOTS.      SET02570
C*
C*
      MOVE THE TEST STORAGE ( TSTSAV,TSTFLD,TSTVER ) TO OVERLAY
      THE INPUT ( MAPTAP ) TRAINING FIELD STORAGE .      SET02580
C*
C*
      IF ( NOCAT .LE. 0 ) GO TO 108      SET02590
C*
      87      CONTINUE      SET02600
      NTSAV = FIELD3 - FLDSV3      SET02610
      NTFLD = VERTX3 - FIELD3      SET02620
      NTVER = TOTVT3 * 2      SET02630
      NMOVE = NTSAV + NTFLD + NTVER      SET02640
      FROMAD = FLDSV3      SET02650
      DO 90 I=1,NMOVE      SET02660
      II = I      SET02670
      ARRAY(II) = ARRAY(FROMAD)      SET02680
      FROMAD = FLDSV3 + II      SET02690
C*
      90      RESET THE TEST/DESIG FIELDS STORAGE BASE ADDRESSES IN ARRAY
      SET02700
C*
C*
      FLDSV3 = 1      SET02710
      FIELD3 = FLDSV3 + NTSAV      SET02720
      VERTX3 = FIELD3 + NTFLD      SET02730
      TOP1 = VERTX3 + NTVER      SET02740
C
      FLDSV2 = TOP1      SET02750
      FIELD2 = TOP1 + 4      SET02760
      SET02770
      SET02780
      SET02790
      SET02800
      SET02810
      SET02820
      SET02830
      SET02840
      SET02850
      SET02860
      SET02870
      SET02880
      SET02890
      SET02900
      SET02910
      SET02920
      SET02930
      SET02940
      SET02950
      SET02960
      SET02970
      SET02980
      SET02990
      SET03000
      SET03010
      SET03020
      SET03030
      SET03040

```

FILE SETUP3

```

      VERTX2 = TOP1 * 5
      TOP2 = VERTX2 * 500
      NOFLD2 = 209
      NOTPFD = 209
      PCT103 = TOP2
      RETURN
C*
C*
86  NOTRFD=NOFLD2
    IF (TSTKEY.EQ.1) NOTRFD=NOFLD3
    PCTSZ=NOTRFD*NOSUR2
    IF (PCTSZ.LE.(TOP-TOP2)) GO TO 106
C*  MOVE TEXT FIELD INFO SO NO GAPS IN ARRAY IF STORAGE IS NEEDED
    IF (TSTKEY.EQ.0) GO TO 105
C*  MUST TRAINING FIELD INFO BE KEPT IN CORE
    IAD=FLDSV2-1
    IF (TRNKEY.EQ.1) IAD=TOP1-1
    IRD=FLDSV3-1
    NF=4*NOFLD3
    TIME=0
    99 DO 101 I=1,NF
    101 ARRAY(IAD+I)=ARRAY(IRD+I)
        IF (TIME.GT.0) GO TO 102
        IAD=IAD+NF
        IRD=IRD+NF
        NF=NF-1
        TIME=TIME+1
        GO TO 99
    102 IF (TIME.EQ.2) GO TO 103
        IAD=IAD+NF
        IRD=IRD+NF
        NF=NF-1
        TIME=TIME+1
        GO TO 99
    103 CONTINUE
        FLDSV3=TOP1
        IF (TRNKEY.EQ.0) FLDSV3=1
        FIELD3=FLDSV3 * 4*NOFLD3
        VERTX3=FIELD3 * 5*NOFLD3
        TOP2 = VERTX3 * 2*TOTVT3
C*
C* 105 IF ( PCTSZ .GT. ( TOP-TOP2 ) ) GO TO 508
C*
C* 106 PCT103 = TOP2
C*  SET UP FIELD APPRAY FOR TRAINING FIELDS
C*
C*  IF( DOTERR .GT. 0 ) RETURN
C*
C* 107 IF (TSTKEY.EQ.0.OR.TRNKEY.EQ.1) GO TO 400
    401 CONTINUE
C*
C*  SET FLOKEY
C*
C*
C*  IF (TSTKEY.NE.1) GO TO 340
    FLOKEY=1
    DO 310 I=1,NOFLD3
    NS = FLOSV3-1+3*(I-1)*4
    IF (ARRAY(NS).EQ.0) GO TO 320
    310 CONTINUE
        GO TO 200
    340 FLOKEY=1
        DO 350 I=1,NOFLD2
        NS = FLOSV2-1+3*(I-1)*4
        IF (ARRAY(NS).EQ.0) GO TO 320
    350 CONTINUE
        RETURN
    320 FLOKEY=0
    200 RETURN
    201 DOTKEY = NDCAT
        FLOKEY = 0
        RETURN
C
C  ERROR ROUTINFS
C
-----

```

SET03050  
 SET03060  
 SET03070  
 SET03080  
 SET03090  
 SET03100  
 SET03110  
 SET03120  
 SET03130  
 SET03140  
 SET03150  
 SET03160  
 SET03170  
 SET03180  
 SET03190  
 SET03200  
 SET03210  
 SET03220  
 SET03230  
 SET03240  
 SET03250  
 SET03260  
 SET03270  
 SET03280  
 SET03290  
 SET03300  
 SET03310  
 SET03320  
 SET03330  
 SET03340  
 SET03350  
 SET03360  
 SET03370  
 SET03380  
 SET03390  
 SET03400  
 SET03410  
 SET03420  
 SET03430  
 SET03440  
 SET03450  
 SET03460  
 SET03470  
 SET03480  
 SET03490  
 SET03500  
 SET03510  
 SET03520  
 SET03530  
 SET03540  
 SET03550  
 SET03560  
 SET03570  
 SET03580  
 SET03590  
 SET03600  
 SET03610  
 SET03620  
 SET03630  
 SET03640  
 SET03650  
 SET03660  
 SET03670  
 SET03680  
 SET03690  
 SET03700  
 SET03710  
 SET03720  
 SET03730  
 SET03740  
 SET03750  
 SET03760  
 SET03770  
 SET03780  
 SET03790  
 SET03800

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE SETUP3

```

C
508 WRITE (6,5044) DIFF
5044 FORMAT (/////5X,***** DISPLAY/SETUP3 - CORE OVERFLOW (TOP-TOP2)
1 BY:16,!-- EXECUTION TERMINATED *****/IH1)
CALL CMERR
108 WRITE (6,109)
109 FORMAT (/////5X,*** CLASSIFICATION BY CATEGORY (ON MAPTAP ) IS
1 REQUIRED IN ORDER TO PROCESS THE DOT DATA **** / 5X, *** DOT PE
2RFORMANCE SUMMARIES WILL NOT BE OUTPUT **** //)
C*
GO TO 112
C*
ERROR RETURN = MAX. NO. OF DOTS EXCEEDED
C*
ERROR IN DOTFILE - RESET DOTKEY AND TRNKEY --
C*
112 DOTKEY = 0
TRNKEY=0
RETURN
C*
-----
C*
INTERNAL ROUTINE TO FIND RECTANGULAR COORDINTES FOR TRAINING FIELD
C*
400 CONTINUE
IR=1
IPT=1
DO 20 I=1,NOFLD2
SAMSTR = 100000
SAMEND = 0
LINSTR = 100000
LINEND = 0
NS = FLDSV2-1+4*(I-1)*4
NV = ARRAY(NS)
NS = FIELD2-1+5*(I-1)*5
ARRAY(NS) = IPT
IPT=IPT+NV*2
IE=IR+NV-1
DO 410 J=IR,IE
NS = VERTX2-1+1+(J-1)*2
SAMSTR = MIN0(SAMSTR,ARRAY(NS))
SAMEND = MAX0(SAMEND,ARRAY(NS))
NS = VERTX2-1+2+(J-1)*2
LINSTR=MIN0(LINSTR,ARRAY(NS))
LINEND=MAX0(LINEND,ARRAY(NS))
410 CONTINUE
NS=FIELD2-1+1*(I-1)*5
ARRAY(NS)=LINSTR
NS=FIELD2-1+2*(I-1)*5
ARRAY(NS)=LINEND
NS=FIELD2-1+3*(I-1)*5
ARRAY(NS)=SAMSTR
NS=FIELD2-1+4*(I-1)*5
ARRAY(NS)=SAMEND
IR=IE+1
20 CONTINUE
GO TO 401
C*
END

```

SET03810  
SET03820  
SET03830  
SET03840  
SET03850  
SET03860  
SET03870  
SET03880  
SET03890  
SET03900  
SET03910  
SET03920  
SET03930  
SET03940  
SET03950  
SET03960  
SET03970  
SET03980  
SET03990  
SET04000  
SET04010  
SET04020  
SET04030  
SET04040  
SET04050  
SET04060  
SET04070  
SET04080  
SET04090  
SET04100  
SET04110  
SET04120  
SET04130  
SET04140  
SET04150  
SET04160  
SET04170  
SET04180  
SET04190  
SET04200  
SET04210  
SET04220  
SET04230  
SET04240  
SET04250  
SET04260  
SET04270  
SET04280  
SET04290  
SET04300  
SET04310  
SET04320  
SET04330  
SET04340  
SET04350  
SET04360  
SET04370  
SET04380  
SET04390  
SET04400  
SET04410  
SET04420  
SET04430

FILE: TINORM

```
FUNCTION TINORM(ALPHA,IFLAG)          TIN00010
DIMENSION A(3),B(3)                  TIN00020
DATA A/.010324,.802853,2.515517/,B/.0010308,
1 .189269,1.432788/                  TIN00030
-----TIN00040
C APPROXIMATION TO INVERSE NORMAL DISTRIBUTION TIN00050
C-----TIN00060
IF(.NOT.(ALPHA.GT.0..AND.ALPHA.LT.1.)) IFLAG=1 TIN00080
X=ALPHA                               TIN00090
IF(X.GT..5) X=1.-X                    TIN00100
Y=SQRT(-2.*ALOG(X))                  TIN00110
TINORM=X-(A(3)+X*(A(2)+X*A(1)))/(1.+X*(B(3)+X*(B(2)+X*B(1)))) TIN00120
CALL OVRFL(I)                         TIN00130
IF(I.EQ.1) GO TO 1                    TIN00140
IF(ALPHA.LT..5) TINORM=-TINORM        TIN00150
RETURN                                 TIN00160
1 IFLAG=1                              TIN00170
RETURN                                 TIN00180
END                                    TIN00190
```

ORIGINAL PAGE IS  
OF POOR QUALITY

~~12-57~~  
236

13. DATA-TR PROCESSOR

FILE: DATATR

```

C      SUBROUTINE DATATR(ARRAY, TOP)                                DAT00010
C      IMPLICIT INTEGER(A-Z)                                       DAT00020
C      REAL    BIAS(16), BMAT(480), MAX(16), MIN(16), CON(16), CONMIN(32)  DAT00030
C      REAL    AMAX(16), AMIN(16), ACON(16)                          DAT00040
C      DIMENSION ARRAY(TOP), MAXPT(30), FILHIS(1616)                DAT00050
C      DIMENSION HDR1(15), HDR2(15), COMNT(15), INDATE(3)          DAT00060
C      INCLUDE COMRK1, LIST                                          DAT00070
C      INCLUDE COMRK9, LIST                                          DAT00080
C      INCLUDE COMRK6, LIST                                          DAT00090
C      COMMON/INFORM/NOCLS2, NOSUB2, NOFET2, VARSZ2, TOTVT2, NOFLD2,  DAT00100
C      *      AVAR2, COVAR2, CLSID2, SUBNO2, SUBDS2, FLOSV2, VERTX2,   DAT00110
C      *      FETVC2(30), SUBVC2(75), SURPTH(75), CLSVC2(60),        DAT00120
C      *      KEPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),             DAT00130
C      *      GRPCHK(61), GROUPS(124)                                DAT00140
C      COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,  DAT00150
C      *      HISFIL, HISKEY, TRFORM, EKIPTR, ERPKEY, MAPUNT, NOFILE,  DAT00160
C      *      DRUMAD, ORMWDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL    DAT00170
C      *      ,NHSTUN, NHSTFI, SCTRUN, MAPFIL                          DAT00180
C      *      ,DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,  DAT00190
C      *      CRDUNT, PRTUNT, RANDIO                                  DAT00200
C DATA TRANSFORMATION COMMON BLOCK                                  DAT00210
CSEND COMMON/TRBLCK/OUTFMT, NOFEAT, FLDFIN(6), FETVEC(30)          DAT00220
C      DIMENSION VERTCS(2,11)                                       DAT00230
C      DATA BLANKS/' '/                                           DAT00240
C
C      RESCALING METHOD IS DETERMINED IN SETUP8 :
C      SCAFLG = 1 , RESCALE BY HISTOGRAM METHOD
C      SCAFLG = 2 , RESCALE BY STATISTICS METHOD
C      SCAFLG = 3 , RESCALE WITH USER-INPUT SCALING PARAMETERS
C
C      IF THE FLAG  RESCAL  IS ZERO, NO RESCALING OCCURS
C
C      DO 10 I=1,15
C      IPL3 = I + 3
C      HDR1(I) = HEAD(IPL3)
C      IPL29 = I + 29
C      HDR2(I) = HEAD(IPL29)
C 10 COMNT(I) = BLANKS
C
C      INDATE(1) = HEAD(22)
C      INDATE(2) = HEAD(23)
C      INDATE(3) = HEAD(24)
C
C      CALL SETUP8(BMAT, LCOMB, BMTRIG, PEROUT, MAXPT, ARRAY, LAM,
C      * SCAFLG, TOP, TRANSF, RESCAL, BIAS, ADDNUM, CONMIN, NPUN, NF )
C      IF (RESCAL.EQ.0) GO TO 50
C
C      IF (SCAFLG.EQ.1) GO TO 30
C      IF (SCAFLG.EQ.2) GO TO 20
C      IF (SCAFLG .EQ. 3 )
C      * CALL SETREM ( CONMIN, CON, MIN, ADDNUM, LCOMB )
C      GO TO 50
C
C      IF RESCALING BY THE STATISTICS METHOD, APPLY TRANSFORMATION TO
C      STATS ( MEANS, COVARIANCES ), OBTAIN TRANSFORMED MAX AND MIN
C      USING TRANSFORMED STATS .

```



FILE: KBTRAN

```
      * SUBROUTINE KBTRAN                                KBT00010
      ( BMAT, LCOMB, ARRAY, LAM, MAX, MIN, EPS, TRANSF) KBT00020
C
      IMPLICIT INTEGER(A-Z)                             KBT00030
      REAL TMIN,TMAX                                    KBT00040
      REAL BMAT(480), MAX(16), MIN(16), EPS(16)        KBT00050
      REAL C(480), CC(480), D(16), DIAG(480), BMEAN(900) KBT00060
C
      INCLUDE COMMK1.LIST                                KBT00070
      INCLUDE COMMK9.LIST                                KBT00080
      INCLUDE COMMK4.LIST                                KBT00090
      INCLUDE COMMK6.LIST                                KBT00100
C
      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VAR522,TOTVT2,NOFLD2, COM00010
      * AVAR2,COVAR2,CLSID2,SURN02,SURDS2,FLDSV2,VERTX2, COM00020
      * FETVC2(30),SUHVC2(75),SURPTR(75),CLSV2(60), COM00030
      * KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61), COM00040
      * GRPCHK(61),GROUPS(124) COM00050
      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15) COM00010
      EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)), COM00020
      (HED2(1),HEAD(30)),(COMENT(1),HEAD(48)) COM00030
      COMMON/GLOBAL/HEAD(63),MAPTAP,DATE,SAVTAP,BMFILE,BMKEY, COM00010
      * HISFIL,HISKEY,IFORM,ERIP,ERPKEY,MAPUNT,NOFILE, COM00020
      * DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL COM00030
      * ,NHSTUN,NHSTFI,SCRUN,MAPFIL COM00040
      * ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, COM00050
      * CROUT,PRUNT,HANDIO COM00060
C DATA TRANSFORMATION COMMON BLOCK
      COMMON/TRRLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30)
C$END
C
      DIMENSION COVHD2(15)
      DIMENSION ARRAY(1)
      DIMENSION NSUB(75)
C
      DATA COVHD2/'... ','TRAN','SFOH','MED ','STAT','ISTI','CS ..',
      * '...'/' BLANKS/' '/'
C
      COMPUTE TRANSFORMED MEANS FOR EACH SURCLASS
C
      DO 10 I=1,15
      10 COVHD2(I) = BLANKS
      DO 20 I=1,NOSUR2
      IPP=SUHUS2+I-1
      20 NSUB(I)=ARRAY(IPP)
      LZ=AVAR2
      K=1
      DO 30 I=1,NOCLS2
      CALL MATVEC(BMAT,ARRAY(LZ),BMEAN(K),LCOMB,NOFET2)
      LZ=LZ+NOFET2
      30 K=K+LCOMB
C
      COMPUTE TRANSFORMED COVARIANCE MATRIX FOR EACH SUBCLASS
C
      KK=0
      K=COVAR2
      KKK=1
      DO 50 I=1,NOCLS2
C MULTIPLY BMAT BY COVARIANCE MATRIX
      CALL MTMLSE(BMAT,ARRAY(K),C,LCOMB,NOFET2)
C MULTIPLY RESULTING MATRIX BY TRANSPOSE OF BMAT
      CALL MTMDAT(C,BMAT,CC,LCOMB,NOFET2,LCOMB,(I),ARRAY(KKK))
      DO 40 II=1,LCOMB
      40 DIAG(KK+II)=D(II)
      KKK=KKK+(LCOMB*(LCOMB+1))/2
      KK=KK+LCOMB
      K=K+VAR522
      50 CONTINUE
C
      PRINT TRANSFORMED COVARIANCE MATRIX
C
      CV1=(LCOMB*(LCOMB+1))/2
C
      IF (TRANSF.EQ.0) GO TO 80
      DO 60 I=1,15
      TEMP = COMENT(I)
      COMENT(I) = COVHD2(I)
      60 COVHD2(I) = TEMP
C
      CALL PRTCOV(ARRAY(1),BMEAN(1),CV1,LCOMB,NSUB(1))
```

FILE: KSTRAN

```
DO 70 I=1,15
70 COMENT(I) = COVHD2(I)
80 CONTINUE
C C CALCULATE MINIMUM AND MAXIMUM FOR EACH SUBCLASS
DO 120 I=1,LCOMB
DO 110 J=1,NOCLS2
NEL = (J - 1) * LCOMB + I
MAX(I) = RMEAN(NEL) * LAM * DIAG(NEL)
IF (J.NE.1) GO TO 90
TMAX = MAX(I)
90 CONTINUE
IF (MAX(I) .GE. TMAX) TMAX = MAX(I)
MIN(I) = RMEAN(NEL) - LAM * DIAG(NEL)
IF (J.NE.1) GO TO 100
TMIN = MIN(I)
100 CONTINUE
IF (MIN(I) .LE. TMIN) TMIN = MIN(I)
110 CONTINUE
MIN(I) = TMIN
MAX(I) = TMAX
FPS(I) = 255. / (MAX(I) - MIN(I))
120 CONTINUE
RETURN
END
```

KBT00650  
KBT00660  
KBT00670  
KBT00680  
KBT00690  
KBT00700  
KBT00710  
KBT00720  
KBT00730  
KBT00740  
KBT00750  
KBT00760  
KBT00770  
KBT00780  
KBT00790  
KBT00800  
KBT00810  
KBT00820  
KBT00830  
KBT00840  
KBT00850  
KBT00860  
KBT00870  
KBT00880  
KBT00890  
KBT00900



FILE LNTRAN

```

SUBROUTINE LNTRAN(IDATA,MAX,MIN,CON,BMAT,LCOMB,BMTRIG,SCAFLG,
* PEROUT,FILMIS,TOP,LAR,FLDNAM,NC,VERTCS, RESCAL, HIAS,
* NF, NPUN )
LNT00010
LNT00020
LNT00030
LNT00040
LNT00050
LNT00060
LNT00070
LNT00080
LNT00090
LNT00100
LNT00110
LNT00120
LNT00130
LNT00140
LNT00150
LNT00160
LNT00170
LNT00180
LNT00190
LNT00200
LNT00210
LNT00220
LNT00230
LNT00240
LNT00250
LNT00260
LNT00270
LNT00280
LNT00290
LNT00300
LNT00310
LNT00320
LNT00330
LNT00340
LNT00350
LNT00360
LNT00370
LNT00380
LNT00390
LNT00400
LNT00410
LNT00420
LNT00430
LNT00440
LNT00450
LNT00460
LNT00470
LNT00480
LNT00490
LNT00500
LNT00510
LNT00520
LNT00530
LNT00540
LNT00550
LNT00560
LNT00570
LNT00580
LNT00590
LNT00600
LNT00610
LNT00620
LNT00630
LNT00640
LNT00650
LNT00660
LNT00670
LNT00680
LNT00690
LNT00700
LNT00710
LNT00720
LNT00730
LNT00740
LNT00750
LNT00760

IF SCAFLG = 1 , RESCALE BY HISTOGRAM METHOD
IF SCAFLG = 2 , RESCALE BY THE STATISTICS METHOD
IF SCAFLG = 3 , RESCALE WITH USER-INPUT SCALING PARAMETERS

NOTE: IF THE FLAG, RESCAL , IS ZERO, NO RESCALING IS
PERFORMED, HOWEVER, PEROUT IS APPLIED TO THE TRANSFORMED
DATA DISTRIBUTION PRIOR TO FINAL OUTPUT OF TRANSFORMED
DATA VALUES .

IMPLICIT INTEGER(A-Z)
REAL TMIN(16), TMAX(16), MATOT , MITOT
REAL NEWMAX(16) , NEWMIN(16) , SUM , CUT
REAL NXCON,PMIN,PMAX,CMIN
REAL HIAS(16), XCON(16), XT(16), YREAL(16), NPER1, NPER2
REAL MAX(16), MIN(16), CON(16), RMAT(4*16), XXCON(16)
REAL MINSAV(16), MAXSAV(16), CONSAV(16)

DIMENSION TOTPTS(16) , PMIN(16) , PMAX(16)
DIMENSION HISRUF(101),VERTCS(2,11),FL(8)
DIMENSION IDATA(TOP) , Y(8000) , FILHIS(LCOMB,101)
DIMENSION RADMIN(16) , BAUMAX(16) , MINCUT(16) , MAXCUT(16)

DATA OP/'(%.CP/)'//,COMMA/','//
DATA TTL/'TOTL'//

INCLUDE COMMK1.LIST
INCLUDE COMMK9.LIST
INCLUDE COMMK4.LIST
INCLUDE COMMK6.LIST
COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VAWSZ2,TOTVT2,NOFLD2,
* AVAR2,COVAR2,CLS1D2,SUBNO2,SUMDS2,FLDSV2,VERTX2,
* FETVC2(30),SUBVC2(75),SURPTR(75),CLSV2(60),
* KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
DIMENSION MED1(15),MED2(15),DATE(3),COMENT(15)
EQUIVALENCE (MED1(1),HEAD(4)),(DATE(1),HEAD(22)),
(MED2(1),HEAD(30)),(COMENT(1),HEAD(48))
2 COMMON/GLOBAL/HEAD(63),MATTAP,DATAPE,SAVTAP,HMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPY,EPKEY,MAPUNT,NOFILE,
* DPUMAD,DPMDIS,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVFL
* NHSTUN,NHSTFI,SCTRIN,MAPFIL
* DOTUNT,DATAFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PWTUNT,WANDIO
C DATA TRANSFORMATION COMMON BLOCK
COMMON/TRBLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30)
C$END
C*** CODE ADDED JAN. 15,1979 TO ALLOW MULTI-FILE OUTPUT
C
RE*IND TRFORM
SKIP = NF - 1
CALL FSEFEL(TRFORM,SKIP,ISTAT)
IF (RESCAL.EQ.0) GO TO 50

CHECK FOR RESCALE FACTORS INPUT BY USER ( SCAFLG = 3 )
IF (SCAFLG.NE.3) GO TO 20

COMPUTE THE TRANSFORMED DATA MAX . USING INPUT
SCALING PARAMETERS , CON AND MIN .

DO 10 KF=1,LCOMB
MAX(KF) = 255./ CON(KF) * MIN(KF)
10 CONTINUE
20 CONTINUE

COMPUTE THE OUTPUT HISTOGRAM SCALE FACTOR. XCON

```

FILE LNTRAN

```

C
DO 30 KK=1,LCOMB
XCON(KK)=(MAX(KK)-MIN(KK))/80
30 FETVC2(KK)=KK
IF (SCAFLG.EQ.1) GO TO 80
FOR STATISTICAL OR INPUT SCALE PARAMETERS, SAVE THE INITIAL
SCALING PARAMETERS ( MIN, MAX, CON ) FOR RE-INITIALIZATION
OF THESE PARAMETERS ON THE SECOND AND SUCCEEDING FIELDS TO BE
INPUT, TRANSFORMED, AND RESCALED ( IF RESCAL GT 0 )
DO 40 I=1,LCOMB
MAXSAV(I) = MAX(I)
MINSAV(I) = MIN(I)
40 CONSAV(I) = CON(I)
POSITION THE INPUT DATA FILE, AND READ IN THE HEADER RECORD
50 CONTINUE
CALL TAPHDR(DATAPE,DATFIL)
GO TO 62
60 NF = NF + 1
READ THE COORDINATES ( VERTICES ) OF THE FIELD FOR THE DATA
TO BE TRANSFORMED .
62 LAM=LAREAD(FLDNAM,VERTCS,FLDINF,NC)
IF (LAM.EQ.0) GO TO 920
IF (LAM.LT.0) GO TO 900
FOR STATISTICAL OR INPUT SCALING PARAMETERS, INITIALIZE THE
SCALING PARAMETERS MAX, MIN, CON, XCON FOR THIS FIELD
IF (RESCAL.EQ.0) GO TO 80
DO 70 I=1,LCOMB
MAX(I) = MAXSAV(I)
MIN(I) = MINSAV(I)
CON(I) = CONSAV(I)
70 XCON(I) = ( MAX(I) - MIN(I) )/80.
80 CONTINUE
DO 90 I=1,LCOMB
MAXCUT(I) = 0
MINCUT(I) = 0
NEWMAX(I) = 255.0
XXCON(I) = XCON(I)
90 NEWMIN(I) = 0.0
MTHAN = 0
NSAMP=(FLDINF(5)-FLDINF(4))/FLDINF(6)+1
IDIM=NOFEAT*NSAMP
IF (IDIM.GT.TOP) GO TO 130
IN=NC-1
WRITE (6,100)
WRITE (6,110)FLDNAM,IN,FLDINF(6),FLDINF(3),((OP,VERTCS(1,K),COMMA,
*VERTCS(2,K),CP),K=1,IN)
100 FORMAT(14,14,'NO. OF SAMPLE LINE',/
*12,'FIELDNAME VERTICES INC INC VERTICES(SAMPLE,LINE)')
110 FORMAT(13X,44,4X,12,7X,14,2X,14,2X,5(A1,14,A1,14,A1,2X)/
* 5(A1,14,A1,14,A1,2X))
XDIM=LCOMB*NSAMP
IF (XDIM.LE.8000) GO TO 150
WRITE (6,120)

```

LNT00770  
LNT00780  
LNT00790  
LNT00800  
LNT00810  
LNT00820  
LNT00830  
LNT00840  
LNT00850  
LNT00860  
LNT00870  
LNT00880  
LNT00890  
LNT00900  
LNT00910  
LNT00920  
LNT00930  
LNT00940  
LNT00950  
LNT00960  
LNT00970  
LNT00980  
LNT00990  
LNT01000  
LNT01010  
LNT01020  
LNT01030  
LNT01040  
LNT01050  
LNT01060  
LNT01070  
LNT01080  
LNT01090  
LNT01100  
LNT01110  
LNT01120  
LNT01130  
LNT01140  
LNT01150  
LNT01160  
LNT01170  
LNT01180  
LNT01190  
LNT01200  
LNT01210  
LNT01220  
LNT01230  
LNT01240  
LNT01250  
LNT01260  
LNT01270  
LNT01280  
LNT01290  
LNT01300  
LNT01310  
LNT01320  
LNT01330  
LNT01340  
LNT01350  
LNT01360  
LNT01370  
LNT01380  
LNT01390  
LNT01400  
LNT01410  
LNT01420  
LNT01430  
LNT01440  
LNT01450  
LNT01460  
LNT01470  
LNT01480  
LNT01490  
LNT01500  
LNT01510  
LNT01520



FILE LNTRAN

```

C          CALL TRANSF TO DO A DATA TRANSFORMATION
C          CALL TRANSF
C          (XT, BMAT, IDATA, TOP, I, K, LCOMB, NSAMP, BIAS)
220 C CONTINUE
DO 300 I=1,LCOMB
IF ( XT(I) .LT. TMIN(I) ) TMIN(I) = XT(I)
IF ( XT(I) .GT. TMAX(I) ) TMAX(I) = XT(I)

IF RESCAL = 0 NO RESCALING IS APPLIED. OTHER WISE RESCALE
USING SCALING PARAMETERS DERIVED FROM EITHER HISTOGRAM,
STATISTICS, OR USER-INPUT ( SCAFLG= 1, 2, OR 3 )

IF (RESCAL.GT.0) GO TO 260

IF TRANSFORMED DATA IS NOT RESCALED,
TEST FOR OUT- OF - RANGE TRANSFORMED VALUES

SET = 0 ANY VALUE LESS THAN 0, OR LESS THAN THE NEW MIN
AFTER APPLICATION OF PEROUT
SET = 255 ANY VALUE GREATER THAN 255, OR GREATER THAN
THE NEW MAX AFTER APPLICATION OF PEROUT

IF (XT(I).LT.NEWMIN(I)) GO TO 230
IF (XT(I).GT.NEWMAX(I)) GO TO 240
GO TO 250
230 IF ( MTRAN .EQ. 0 ) BADMIN(I) = BADMIN(I) + 1
XT(I) = 0.0
GO TO 250
240 IF ( MTRAN .EQ. 0 ) BADMIN(I) = BADMIN(I) + 1
XT(I) = 255.
250 CONTINUE

FOR THE CURRENT SCAN LINE, HISTOGRAM THE TRANSFORMED DATA,
AND STORE THE TRANSFORMED DATA INTO THE OUTPUT ARRAY, Y .

DPT = XT(I)/XCON + 1.1
TOTPTS(I) = TOTPTS(I) + 1
IF ( DPT .GT. 101 ) DPT = 101
IF ( DPT .LE. 0 ) DPT = 1
FILHIS(I,DPT) = FILHIS(I,DPT) + 1
ZS = (I-1) * NSAMP + K
Y(ZS) = XT(I) + 0.5
GO TO 300
260 CONTINUE

FOR THE CURRENT SCAN LINE, HISTOGRAM THE TRANSFORMED DATA, AND
STORE THE TRANSFORMED DATA INTO THE OUTPUT ARRAY, Y .

IF (XT(I).LT.MIN(I)) GO TO 270
IF (XT(I).GT.MAX(I)) GO TO 280
YREAL(I)=CON(I)+(XT(I)-MIN(I))

DPT = ( XT(I) - MIN(I) ) / XCON(I) + 11
IF ( DPT .LE. 0 ) DPT = 1
IF ( DPT .GT. 101 ) DPT = 101

GO TO 290
270 DPT = ABS( MIN(I) - XT(I) )/XCON(I)
DPT = 10 - DPT

PMIN(I) = PMIN(I) + 1
IF (DPT.LE.0) DPT=1
YREAL(I)=0
GO TO 290
280 DPT = ABS( XT(I) - MAX(I) )/XCON(I)
DPT = DPT + 91

```

LNT02290  
LNT02300  
LNT02310  
LNT02320  
LNT02330  
LNT02340  
LNT02350  
LNT02360  
LNT02370  
LNT02380  
LNT02390  
LNT02400  
LNT02410  
LNT02420  
LNT02430  
LNT02440  
LNT02450  
LNT02460  
LNT02470  
LNT02480  
LNT02490  
LNT02500  
LNT02510  
LNT02520  
LNT02530  
LNT02540  
LNT02550  
LNT02560  
LNT02570  
LNT02580  
LNT02590  
LNT02600  
LNT02610  
LNT02620  
LNT02630  
LNT02640  
LNT02650  
LNT02660  
LNT02670  
LNT02680  
LNT02690  
LNT02700  
LNT02710  
LNT02720  
LNT02730  
LNT02740  
LNT02750  
LNT02760  
LNT02770  
LNT02780  
LNT02790  
LNT02800  
LNT02810  
LNT02820  
LNT02830  
LNT02840  
LNT02850  
LNT02860  
LNT02870  
LNT02880  
LNT02890  
LNT02900  
LNT02910  
LNT02920  
LNT02930  
LNT02940  
LNT02950  
LNT02960  
LNT02970  
LNT02980  
LNT02990  
LNT03000  
LNT03010  
LNT03020  
LNT03030  
LNT03040

FILE LNTRAN

```

PMAK(I) = PMAK(I) * 1
IF (OPT.GT.101) OPT=101
YREAL(I)=255
290 TOTPTS(I)=TOTPTS(I)+1
FILHIS(I,OPT)=FILHIS(I,OPT)+1
7SAMP=(I-1)*NSAMP*K
Y(2SAMP) = YREAL(I) * 0.5
300 CONTINUE
GO TO 330
310 IF (.J.KP).GE.JJ) GO TO 340
320 CONTINUE
330 CONTINUE
340 CONTINUE
IF (M.EU.LINES) LSTLIN=-1

OUTPUT ONE LINE OF TRANSFORMED DATA ON THE OUTPUT FILE , TRFORM

CALL WPTLN(Y.LSTLIN)
GO TO 180

IF (RESCALING THE TRANSFORMED DATA BY EITHER THE STATISTICAL
OR USER-INPUT SCALING PARAMETERS.
APPLY PEROUT ( OF POINTS TO BE REJECTED) TO THE TRANSFORMED
DATA DISTRIBUTION - OBTAIN THE MAX. AND SCALING PARAMETERS
MIN AND CON . AFTER APPLICATION OF PEROUT ( ALSO, THE HISTOGRAM
SCALE FACTOR . XCON ) .

IF NOT RESCALING. APPLY PEROUT TO THE TRANSFORMED DATA.
GET NEW MAX AND MIN. RE-HISTOGRAM , AND OUTPUT THE REVISED DISTR.

350 CONTINUE
IF (PEROUT.LE.0) GO TO 600
IF (SCAFLG.EQ.1) GO TO 600
IF (MTRAN.EQ.1) GO TO 600
IF (RESCAL.GT.0) GO TO 430
NPER1 = FLOAT(PEROUT)/200.0
DO 470 I=1,LCOMB
CUT = NPER1 * FLOAT( TOTPTS(I) )
SUM = 0.0
DO 370 J=1,101.1
IF (SUM.GE.CUT) GO TO 360
GO TO 370
360 MINCUT(I) = SUM
NEWMIN(I) = (J-1) * NXCON * 0.5
GO TO 380
C
370 SUM = SUM * FILHIS(I,J)
380 SUM = 0.0
J=101
385 J=J-1
IF (SUM.GE.CUT) GO TO 390
GO TO 400
390 MAXCUT(I) = SUM
NEWMAX(I) = ( J - 1 ) * NXCON * 0.5
GO TO 410
400 SUM = SUM * FILHIS(I,J)
IF (J.GT.1) GO TO 385
410 CONTINUE
420 CONTINUE
GO TO 580
C
430 NPER1 = PEROUT * .01 * .001
NPER2 = PEROUT * .01 * .001
C
RSET=0
IG = 0
IH = 0

```

ORIGINAL PAGE IS  
OF POOR QUALITY

LNT03050  
LNT03060  
LNT03070  
LNT03080  
LNT03090  
LNT03100  
LNT03110  
LNT03120  
LNT03130  
LNT03140  
LNT03150  
LNT03160  
LNT03170  
LNT03180  
LNT03190  
LNT03200  
LNT03210  
LNT03220  
LNT03230  
LNT03240  
LNT03250  
LNT03260  
LNT03270  
LNT03280  
LNT03290  
LNT03300  
LNT03310  
LNT03320  
LNT03330  
LNT03340  
LNT03350  
LNT03360  
LNT03370  
LNT03380  
LNT03390  
LNT03400  
LNT03410  
LNT03420  
LNT03430  
LNT03440  
LNT03450  
LNT03460  
LNT03470  
LNT03480  
LNT03490  
LNT03500  
LNT03510  
LNT03520  
LNT03530  
LNT03540  
LNT03550  
LNT03560  
LNT03570  
LNT03580  
LNT03590  
LNT03600  
LNT03610  
LNT03620  
LNT03630  
LNT03640  
LNT03650  
LNT03660  
LNT03670  
LNT03680  
LNT03690  
LNT03700  
LNT03710  
LNT03720  
LNT03730  
LNT03740  
LNT03750  
LNT03760  
LNT03770  
LNT03780  
LNT03790  
LNT03800

FILE LNTRAN

```

      IE = 0
      IA = 0
C
      DO 560 I=1,LCOM9
      MATOT=TOTPTS(I)*NPER1
      MITOT=TOTPTS(I)*NPER2
      IF (PMIN(I).GT.MATOT) GO TO 440
      GO TO 460
C RESET MIN SMALLER
440 CMIN=PMIN(I)
      J=10
445 J=J-1
      IG=IG+1
      CMIN=CMIN-FILHIS(I,J)
      IF (CMIN.GT.MATOT) GO TO 450
      IF (CMIN.LT.MITOT) IG=IG-1
      RSET=1
      MIN(I)=MIN(I)-IG*XCON(I)
      IG=0
      GO TO 490
450 IF (J.GT.1) GO TO 445
460 IF (PMIN(I).LT.MITOT) GO TO 470
      GO TO 490
C RESET MIN LARGER
470 CMIN=PMIN(I)
      DO 480 J=11,91
      IH=IR+1
      CMIN=CMIN+FILHIS(I,J)
      IF (CMIN.LT.MITOT) GO TO 480
      IF (CMIN.GT.MATOT) IH=IH-1
      MIN(I)=MIN(I)+IH*XCON(I)
      IR=0
      RSET=1
      GO TO 490
480 CONTINUE
C CHECK MAX
490 CONTINUE
      IF (PMAX(I).GT.MATOT) GO TO 500
      GO TO 520
C RESET MAX LARGER
500 CMIN=PMAX(I)
      DO 510 J=92,101
      IE=IF+1
      CMIN=CMIN-FILHIS(I,J)
      IF (CMIN.GT.MATOT) GO TO 510
      IF (CMIN.LT.MITOT) IE=IE-1
      RSET=1
      MAX(I)=MAX(I)+IE*XCON(I)
      IE=0
      GO TO 550
510 CONTINUE
C RESET MAX SMALLER
520 IF (PMAX(I).LT.MITOT) GO TO 530
      GO TO 550
530 CMIN=PMAX(I)
      J=91
535 J=J-1
      IA=IA+1
      CMIN=CMIN+FILHIS(I,J)
      IF (CMIN.LT.MITOT) GO TO 540
      IF (CMIN.GT.MATOT) IA=IA-1
      RSET=1
      MAX(I)=MAX(I)-IA*XCON(I)
      IA=0
      GO TO 550
540 IF (J.GT.1) GO TO 535
550 CONTINUE
560 CONTINUE
      IF (RSET.EQ.0) GO TO 600
      DO 570 I=1,LCOM9
      XCON(I)=(MAX(I)-MIN(I))/40
      XXCON(I) = XCON(I)
570 CON(I) = 255. / (MAX(I) - MIN(I))
C
580 IF (NF.EQ.1) GO TO 590
C

```

LNT03810  
 LNT03820  
 LNT03830  
 LNT03840  
 LNT03850  
 LNT03860  
 LNT03870  
 LNT03880  
 LNT03890  
 LNT03900  
 LNT03910  
 LNT03920  
 LNT03930  
 LNT03940  
 LNT03950  
 LNT03960  
 LNT03970  
 LNT03980  
 LNT03990  
 LNT04000  
 LNT04010  
 LNT04020  
 LNT04030  
 LNT04040  
 LNT04050  
 LNT04060  
 LNT04070  
 LNT04080  
 LNT04090  
 LNT04100  
 LNT04110  
 LNT04120  
 LNT04130  
 LNT04140  
 LNT04150  
 LNT04160  
 LNT04170  
 LNT04180  
 LNT04190  
 LNT04190  
 LNT04200  
 LNT04210  
 LNT04220  
 LNT04230  
 LNT04240  
 LNT04250  
 LNT04260  
 LNT04270  
 LNT04280  
 LNT04290  
 LNT04300  
 LNT04310  
 LNT04320  
 LNT04330  
 LNT04340  
 LNT04350  
 LNT04360  
 LNT04370  
 LNT04380  
 LNT04390  
 LNT04400  
 LNT04410  
 LNT04420  
 LNT04430  
 LNT04440  
 LNT04450  
 LNT04460  
 LNT04470  
 LNT04480  
 LNT04490  
 LNT04500  
 LNT04510  
 LNT04520  
 LNT04530  
 LNT04540  
 LNT04550  
 LNT04560

FILE LNTRAN

```

REWIND TRFORM
SKIP=NF-1
CALL FFSMFL (TRFORM,SKIP,ISTAT)
MTRAN=1
GO TO 150
590 REWIND TRFORM
MTRAN=1
GO TO 150
600 CONTINUE
MTRAN=0
C
IF (RESCAL.EQ.0) GO TO 710
C
WRITE (6.610)NF
610 FORMAT (///// 22X, '* OUTPUT FILE ', I3, 1X, '**' // '//// 5X,
* '*** TRANSFORMED VALUES RESCALED TO A RANGE 0 - 255 ***' )
C
IF (SCAFLG.EQ.1) *WRITE (6.620)
IF (SCAFLG.EQ.2) WRITE (6.630)
IF (SCAFLG.EQ.3) WRITE (6.640)
C
620 FORMAT ( 24X, '(HISTOGRAM METHOD)' / )
630 FORMAT ( 23X, '(STATISTICS METHOD)' / )
640 FORMAT ( 20X, '(INPUT SCALING PARAMETERS)' / )
C
WRITE (6.650)
C
650 FORMAT (///// 7X, '... ORIGINAL TRANSFORMED DATA RANGE ...' //
* 'T11, 'MIN', T32, 'MAX', T51, '(BIAS)' // )
C
DO 660 M=1,LCOMB
660 WRITE (6.670)TMIN(M),TMAX(M),BIAS(M)
C
670 FORMAT ( 5X, F11.4, 10X, F11.4, 9X, '( ', F11.4, 1X, ') ' / )
C
WRITE (6.680)
680 FORMAT (///// 7X, '... TRANSFORMED DATA RANGE, AFTER APPLICATION OF
* PEROUT = 255 // 7X, 'MIN', 10X, 'MAX', 10X,
* 'CON = 255/(MAX-MIN)' // )
C PRINT OUT NEW MAX,MIN,CON ARRAYS
DO 690 M=1,LCOMB
690 WRITE (6.700)MIN(M),MAX(M),CON(M)
700 FORMAT (5X,3(F8.4,5X))
C
GO TO 850
C
710 IF (PEROUT.GT.0) GO TO 730
C
DO 720 I=1,LCOMB
IF (HADMIN(I).EQ.0) NEWMIN(I) = TMIN(I)
IF (HADMAX(I).EQ.0) NEWMAX(I) = TMAX(I)
720 CONTINUE
C
730 WRITE (6.740)NF
740 FORMAT (///// 19X, '* OUTPUT FILE ', I3, 1X, '**' )
C
WRITE (6.750)
750 FORMAT (///// 5X, '*** TRANSFORMED VALUES NOT RESCALED ***' // )
C
WRITE (6.760)LCOMB,(TMIN(I),I=1,LCOMB)
760 FORMAT (// 5X, 'TRANSFORMED MINIMUMS, COMPONENTS 1-' , I2, 2X,
* '....' // 2(5X, 8F12.2 / ) )
C
WRITE (6.770)LCOMB,(TMAX(I),I=1,LCOMB)
770 FORMAT (// 5X, 'TRANSFORMED MAXIMUMS, COMPONENTS 1-' , I2, 2X,
* '....' // 2( 5X, 8F12.2 / ) )
C
WRITE (6.780)LCOMB,(BIAS(I),I=1,LCOMB)
780 FORMAT (/// 5X, 'TRANSFORMED VALUE BIAS, COMPONENTS 1 - ' , I4, 2X,
* '...' // 2( 5X, 8F12.2 / ) )
C

```

LNT04570  
LNT04580  
LNT04590  
LNT04600  
LNT04610  
LNT04620  
LNT04630  
LNT04640  
LNT04650  
LNT04660  
LNT04670  
LNT04680  
LNT04690  
LNT04700  
LNT04710  
LNT04720  
LNT04730  
LNT04740  
LNT04750  
LNT04760  
LNT04770  
LNT04780  
LNT04790  
LNT04800  
LNT04810  
LNT04820  
LNT04830  
LNT04840  
LNT04850  
LNT04860  
LNT04870  
LNT04880  
LNT04890  
LNT04900  
LNT04910  
LNT04920  
LNT04930  
LNT04940  
LNT04950  
LNT04960  
LNT04970  
LNT04980  
LNT04990  
LNT05000  
LNT05010  
LNT05020  
LNT05030  
LNT05040  
LNT05050  
LNT05060  
LNT05070  
LNT05080  
LNT05090  
LNT05100  
LNT05110  
LNT05120  
LNT05130  
LNT05140  
LNT05150  
LNT05160  
LNT05170  
LNT05180  
LNT05190  
LNT05200  
LNT05210  
LNT05220  
LNT05230  
LNT05240  
LNT05250  
LNT05260  
LNT05270  
LNT05280  
LNT05290  
LNT05300  
LNT05310  
LNT05320

FILE LNTRAN

```

C
C
C      WRITE (6,790) (I,HADMIN(I),I=1,LCOMB)
C      790 FORMAT(/// 5X,'NO. OF TRANSFORMED VALUES LESS THAN 0 ( SET = 0 ) :
C      * // (5X,'COMPONENT', 1X, I2, '...', 16, 2X, 'VALUES' ) )
C
C      WRITE (6,800) (I,HADMAX(I),I=1,LCOMB)
C      800 FORMAT(/// 5X, 'NO. OF TRANSFORMED VALUES GREATER THAN 255 ( SET =
C      * 255 ) : ' // (5X,'COMPONENT', 1X, I2,'...',16, 2X, 'VALUES' ) )
C
C      NPER1 = FLOAT(PEROUT)/2.0
C      WRITE (6,810)NPER1,LCOMB,(MINCUT(I),I=1,LCOMB)
C      810 FORMAT(/// 3X, 'NO. OF LOWER TAIL POINTS REJECTED ( SET = 0 FL
C      *OR OUTPUT ) TO SATISFY', F6.1,2X, '- CUT-OFF, COMPONENTS 1 -',
C      * I4, 2X, '...' // 16(I8) )
C
C      WRITE (6,820)NPER1,LCOMB,(MAXCUT(I),I=1,LCOMB)
C      820 FORMAT(/// 3X, 'NO. OF UPPER TAIL POINTS REJECTED ( SET = 255 FL
C      *OR OUTPUT ) TO SATISFY', F6.1, 2X, '- CUT-OFF, COMPONENTS 1 -',
C      * I4, 2X, '...' // 16(I8) )
C      NPCT = 100 - PEROUT
C      WRITE (6,830)NPCT,LCOMB,(NEWMIN(I),I=1,LCOMB)
C      830 FORMAT(/// 3X,'**** FINAL OUTPUT TRANSFORMED VALUES, CENTRAL',
C      * 15,3X,'-', 2X, 'OF DISTRIBUTION : ' // 3X, 'MINIMUMS, COMPONENTS
C      * 1 - ' , I4, 2X, '...' // 2(5X, 8F12.2 / ) )
C
C      WRITE (6,840)LCOMB,(NEWMAX(I),I=1,LCOMB)
C      840 FORMAT(// 3X,'MAXIMUMS, COMPONENTS 1 - ', I4, 2X,'...' //
C      * 2( 5X, 8F12.2 / ) )
C
C
C      PRINT HISTOGRAMS
C      850 CONTINUE
C
C      XSIZ=101
C      XHGH=255
C      XLOW=0
C      YSIZ=15
C      CALL COMHST(FILHIS,HISBUF,TTL,LCOMB,FETVC2,XSIZ,XHGH,XLOW,YSIZ)
C
C
C      IF (RESCAL.EQ.0) GO TO 60
C
C
C      WRITE (6,860)NF
C      860 FORMAT(1H1 // 5X,'SCALING PARAMETERS USED ON TRANSFORMED VALUES, O
C      *UTPUT FILE', I5// 19X, 'MINIMUM', 7X, 'MAXIMUM', 7X,
C      * 'SCALE FACTOR ( CON )' )
C
C      WRITE (6,870) (FETVC2(IL),MIN(IL),MAX(IL),CON(IL),IL=1,LCOMB)
C      870 FORMAT(1X, 'COMPONENT', 13, 1X, F12.3, 2X, F12.3, 7X, F12.3 )
C
C      IF (NPUN.LE.0) GO TO 890
C
C      PUNCH 880, (CON(MN),MIN(MN),MN=1,LCOMB)
C      880 FORMAT( ('OPTION', 4X, 'SCAFAC=', 2( ' (', F9.3, ',', F9.3,
C      * ' ) , ' ) ) )
C      890 CONTINUE

```

LNT05330  
 LNT05340  
 LNT05350  
 LNT05360  
 LNT05370  
 LNT05380  
 LNT05390  
 LNT05400  
 LNT05410  
 LNT05420  
 LNT05430  
 LNT05440  
 LNT05450  
 LNT05460  
 LNT05470  
 LNT05480  
 LNT05490  
 LNT05500  
 LNT05510  
 LNT05520  
 LNT05530  
 LNT05540  
 LNT05550  
 LNT05560  
 LNT05570  
 LNT05580  
 LNT05590  
 LNT05600  
 LNT05610  
 LNT05620  
 LNT05630  
 LNT05640  
 LNT05650  
 LNT05660  
 LNT05670  
 LNT05680  
 LNT05690  
 LNT05700  
 LNT05710  
 LNT05720  
 LNT05730  
 LNT05740  
 LNT05750  
 LNT05760  
 LNT05770  
 LNT05780  
 LNT05790  
 LNT05800  
 LNT05810  
 LNT05820  
 LNT05830  
 LNT05840  
 LNT05850  
 LNT05860  
 LNT05870  
 LNT05880  
 LNT05890  
 LNT05900  
 LNT05910  
 LNT05920  
 LNT05930  
 LNT05940  
 LNT05950  
 LNT05960  
 LNT05970  
 LNT05980  
 LNT05990  
 LNT06000  
 LNT06010  
 LNT06020  
 LNT06030  
 LNT06040  
 LNT06050  
 LNT06060  
 LNT06070  
 LNT06080



FILE LNTRAN

```
C      IF ( SCAFLG .EQ. 1 ) RETURN
C
C      GO TO 60
900 IF (LAM.EQ.0) GO TO 920
WRITE (6,910)FLDNAM
910 FORMAT(//// 5X,'***** DATATR/LNTRAN ***** ERROR ON INPUT FIELD DE
*FINITION CARD. FOR FIELD NAME ',1H', A4, 1H', 3X, '*****' /
* 10X, 'CONTINUING TO NEXT FIELD DEFINITION CARD(S): //// ')
C      GO TO 60
920 CONTINUE
RETURN
END
```

LNT06090  
LNT06100  
LNT06110  
LNT06120  
LNT06130  
LNT06140  
LNT06150  
LNT06160  
LNT06170  
LNT06180  
LNT06190  
LNT06200  
LNT06210  
LNT06220

ORIGINAL PAGE  
OF POOR QUALITY

FILE: MAXMAT

```

C      SURROUTINE MAXMAT ( MAX, MIN, CON, BMAT, LCOMB, MAXPT )           MXM00010
C      COMPUTE AN APPROXIMATE TRANSFORMED MAX AND MIN FOR EACH COMPONENT MXM00020
C      OF THE TRANSFORMATION                                           MXM00030
C                                                                           MXM00040
C      IMPLICIT INTEGER(A-Z)                                           MXM00050
C      DIMENSION MAXPT(30)                                             MXM00060
C                                                                           MXM00070
C                                                                           MXM00080
C      REAL  BMAT(480), MAX(16), MIN(16), CON(16)                     MXM00090
C                                                                           MXM00100
C                                                                           MXM00110
C      USING INPUT ( OR DEFAULT ) MAXIMUM DATA VALUE FOR EACH      MXM00120
C      CHANNEL, COMPUTE THE TRANSFORMED VALUE RANGE ( MAX AND MIN )  MXM00130
C      AND COMPUTE THE HISTOGRAM SCALING FACTOR, CON .                MXM00140
C                                                                           MXM00150
C                                                                           MXM00160
C                                                                           MXM00170
C                                                                           MXM00180
C      INCLUDE COMRK9.LIST
C DATA TRANSFORMATION COMMON HLOCK
COMMON/TRHLCK/OUTFMT,NOFEAT,FLDINF(6),          FETVEC(30)
CSEND
DO 30 I=1,LCOMB
MAX(I) = 0.0
MIN(I) = 0.0
DO 20 J=1,NOFEAT
K=(J-1)*LCOMB+I
IF (BMAT(K).LE.0.0) GO TO 10
C      MAX(I) = MAX(I) + BMAT(K) * MAXPT(J)
C
C      GO TO 20
10 CONTINUE
C
C      MIN(I) = MIN(I) + BMAT(K) * MAXPT(J)
C
C      20 CONTINUE
CON(I)=(MAX(I)-MIN(I))/100.
30 CONTINUE
RETURN
END
MXM00200
MXM00210
MXM00220
MXM00230
MXM00240
MXM00250
MXM00260
MXM00270
MXM00280
MXM00290
MXM00300
MXM00310
MXM00320
MXM00330
MXM00340
MXM00350
MXM00360
MXM00370
MXM00380
MXM00390
```

FILE: SETREM

```
      SUPROUTINE SETREM
      (CONMIN, CON, MIN, ADDNUM, LCOMB)
C
      IMPLICIT INTEGER (A-Z)
      REAL CON(16), MIN(16), CONMIN(2, 16)
      NUMCM = ADDNUM / 2
      IF (NUMCM.NE.LCOMB) GO TO 20
      DO 10 NM=1,LCOMB
        CON(NM) = CONMIN(1, NM)
        MIN(NM) = CONMIN(2, NM)
10     CONTINUE
      RETURN
20     WRITE (6,30)NUMCM,LCOMB
30     FORMAT (5X, 'SETREM ERROR - THERE WERE ', I5, ' SCALE ',
C        'FACTORS AND MINIMUM VALUES INPUT THROUGH THE SCAFAC ',
C        'OPTION.',
C        /, 5X, I5, ' LINEAR COMBINATIONS WERE REQUESTED.',
C        /, 5X, 'THERE MUST BE A SCALE FACTOR AND A MINIMUM ',
C        'VALUE FOR EACH LINEAR COMBINATION.',
C        /, 5X, 'THE PROGRAM WILL TERMINATE THROUGH CMERR')
      CALL CMERR
      END
```

SRE00010  
SRE00020  
SRE00030  
SRE00040  
SRE00050  
SRE00060  
SRE00070  
SRE00080  
SRE00090  
SRE00100  
SRE00110  
SRE00120  
SRE00130  
SRE00140  
SRE00150  
SRE00160  
SRE00170  
SRE00180  
SRE00190  
SRE00200  
SRE00210  
SRE00220

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: SETUP8

```

SUBROUTINE SETUP8(BMAT,LCOMB,BMTRIG,PEROUT,MAXPT,ARRAY,LAM,SCAFLG,SET00010
* TOP, TRANSF, RESCAL, BIAS, ADDNUM, CONMIN, NPUN, NF) SET00020
C IMPLICIT INTEGER(A-Z) SET00030
C LOGICAL NUDTAP, NUDFIL, NUSTAP, NUSFIL SET00040
C REAL CONMIN(32), BIAS(16), BMAT(480) SET00050
DIMENSION MAXPT(30) SET00060
DIMENSION ARRAY(1) SET00070
DIMENSION EQUVEC(2) SET00080
DIMENSION CINDEX(19), SET00090
* SINVEC(3), FRVEC1(3), FRVEC2(3), CARD2(62), SET00100
* BTEST(3) SET00110
DIMENSION COVHD1(15), OP(2), CP(2) SET00120
DIMENSION MTX(5), ACARD(20) SET00130
C INCLUDE COMAK1,LIST SET00140
C INCLUDE COMAK4,LIST SET00150
C INCLUDE COMAK6,LIST SET00160
C INCLUDE COMAK9,LIST SET00170
COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARZ2,TOTVT2,NOFLD2, SET00180
* AVAR2,COVAR2,CLSD2,SURN02,SUBDS2,FLDSV2,VENTX2, SET00190
* FETVC2(30),SUBVC2(75),SUBPTR(75),CLVC2(60), SET00200
* KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61), SET00210
* GRPCHK(61),GROUPS(124) SET00220
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15) SET00230
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)), SET00240
2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48)), SET00250
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY, SET00260
* HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE, SET00270
* DRUMAD,DRMWD5,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL SET00280
* ,NHSTUN,NHSTFI,SCTRUN,MAPFIL SET00290
* ,DOTUNT,DOTFIL,NCHPAS,TRANSF,BMTRFL,HISTFL,PCHUNT, SET00300
* CRDUNT,PKTUNT,RANDIO SET00310
C DATA TRANSFORMATION COMMON BLOCK SET00320
COMMON/TRBLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30) SET00330
C END SET00340
EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND), SET00350
* (FLDINF(3),LININC),(FLDINF(4),SAMSTR), SET00360
* (FLDINF(5),SAMEND),(FLDINF(6),SAMINC) SET00370
C DATA CINDEX/'R-MA','CHAN','FORM','HED1','HED2', SET00380
* 'DATE','COMM','MAXP','PERO','SUBC', SET00390
* 'MODU','LAM','OPTI','END','DATA','STAT', SET00400
* 'RESC','BIAS','TROUT' SET00410
DATA MTX / 4, '0', 'T', 'S', 'P' / SET00420
DATA OP / 1, '1' / , CP / 1, '1' / , ZERO / 0 / SET00430
DATA EQUVEC / 1, '1' / SET00440
DATA SINVEC / 2, '1', '1' / SET00450
* CINMAX / 19, MAXFET / 30, SET00460
* FRVEC1 / 2, '1', '0' / , FRVEC2 / 2, 'U', 'L' / , BLANK / ' ' / SET00470
* BTEST / 2, 'C', 'F' / SET00480
4, CBCD / 'C' / , FRCD / 'F' / , UBDC / 'U' / SET00490
DATA COVHD1 / '... ' , 'ORIG', 'INAL', 'STA', 'TIST', 'ICS ' , '... ' / SET00500
C INITIALIZE FLAGS AND DEFAULT VALUES SET00510
NOSUR2=0 SET00520
NOGRP=0 SET00530
BMTRIG=0 SET00540
NSF = 1 SET00550
RESCAL = 0 SET00560
SCAFLG = 0 SET00570
MPT = 0 SET00580
ORIG = 0 SET00590
TRANSF = 0 SET00600
NPUN = 0 SET00610
OUTFMT=2 SET00620
TRFORM=14 SET00630
C INITIALIZE THE TRANSFORMATION BIAS VECTOR (BIAS) AND NO. OF SET00640
BIAS VALUES ( NBS ) SET00650
DO 10 I=1,16 SET00660
10 BIAS(I) = 0.0 SET00670
SET00680
SET00690
SET00700
SET00710
SET00720
SET00730
SET00740
SET00750
SET00760
SET00770
SET00780
SET00790

```

FILE: SETUP8

```

NBS = 0
NUDTAP = .FALSE.
NUDFIL = .FALSE.
NUSTAP = .FALSE.
NUSFIL = .FALSE.
BMSWT=1

C INITIALIZE THE MAXIMUM EXPECTED DATA VALUE, FOR EACH CHANNEL
C
  DO 20 I=1,30
  20 MAXPT(I)=255

C INITIALIZE DISTRIBUTION CUT-OFF, PEROUT, AND
C THE STANDARD DEVIATION MULTIPLE, LAM.
C
  LAM=2
  PEROUT=5
  DO 30 I=8,15
  30 COVHD1(I) = BLANK

C
  DO 40 I=1,15
  40 COMENT(I) = BLANK

C NOW SET UP REREAD BUFFER,
C
  CALL REREAD(30,80)
  50 COL=0
C NOW READ A CARD INTO THE BUFFER
  HEAD(21,55)(ACARD(I),I=1,20)
  55 FORMAT(20A4)
  WRITE(30,55)(ACARD(I),I=1,20)
  REWIND 30
C STATFILE CARD READ
  IF (NUSTAP .OR. NUSFIL) SCAFLG = 2
  READ (30,60)CODE,CARD2
  60 FORMAT(A4,6X,62A1)
  REWIND 30
  WRITE (6,70)CODE,CARD2
  70 FORMAT(T5,A4,6X,62A1)
  DO 80 I=1,CINMAX
  IF (CINDEX(I).EQ.CODE)GO TO(110,150,160,180,190,210,200,230,250,
*270,360,280,290,500,380,430,480,490,600),I
  80 CONTINUE

C
  90 WRITE (6,100)CODE,CARD2
C
  100 FORMAT(//// 5X,*** BAD CONTROL CARD - DATATR/SETUP8 *** // 5X,
* A4, 6X, 62A1 /// )
C
  GO TO 50
C B-MATRIX CARD
  110 J=NXTCHR(CARD2,COL)
  IF (J.EQ.BLANK) GO TO 540
  COL=COL-1
  M=FIND12(CARD2,COL,BTEST)
  IF (M.EQ.-1) GO TO 540
  HMTRIG=1
  IF (M.EQ.2) GO TO 120
C B-MATRIX DATA ON TAPE FILE
  KEY=2
C READ B-MATRIX ARRAY FROM TAPE FILE
  CALL BMFIL(BMAT,LCOMB,NOFEAT,FETVEC,KEY)
  GO TO 130
C B-MATRIX DATA HEAD FROM CARD FILE
  120 KEY=1
  CALL BMFIL(BMAT,LCOMB,NOFEAT,FETVEC,KEY)
  130 NOFET2=NOFEAT
  NOFET4=LCOMB
  DO 140 B=1,NOFEAT
  140 FETVC2(B)=FETVEC(B)
  GO TO 50
C FEATURE CARD
  150 CONTINUE
  GO TO 50
C FORMAT CARD
  160 CONTINUE
  170 M=FIND12(CARD2,COL,FRVEC1)
  IF (M.EQ.-1) GO TO 540
  KZ=FIND12(CARD2,COL,SINVEC)

```

SET00800  
SET00810  
SET00820  
SET00830  
SET00840  
SET00850  
SET00860  
SET00870  
SET00880  
SET00890  
SET00900  
SET00910  
SET00920  
SET00930  
SET00940  
SET00950  
SET00960  
SET00970  
SET00980  
SET00990  
SET01000  
SET01010  
SET01020  
SET01030  
SET01040  
SET01050  
SET01060  
SET01070  
SET01080  
SET01090  
SET01100  
SET01110  
SET01120  
SET01130  
SET01140  
SET01150  
SET01160  
SET01170  
SET01180  
SET01190  
SET01200  
SET01210  
SET01220  
SET01230  
SET01240  
SET01250  
SET01260  
SET01270  
SET01280  
SET01290  
SET01300  
SET01310  
SET01320  
SET01330  
SET01340  
SET01350  
SET01360  
SET01370  
SET01380  
SET01390  
SET01400  
SET01410  
SET01420  
SET01430  
SET01440  
SET01450  
SET01460  
SET01470  
SET01480  
SET01490  
SET01500  
SET01510  
SET01520  
SET01530  
SET01540  
SET01550  
SET01560  
SET01570  
SET01580

FILE: SETUP8

```

      KM=FIN12(CARD2,COL,FRVEC2)
      IF ((M.EQ.3).AND.(KM.EQ.2)) OUTFMT=1
      IF ((M.EQ.3).AND.(KM.EQ.3)) OUTFMT=2
      Z=FIN12(CARD2,COL,SINVEC)
      IF (Z.NE.2) GO TO 50
      GO TO 170
C HED1 CARD
180 READ (30,220)HED1
      REWIND 30
      GO TO 50
C HED2 CARD
190 READ (30,220)HED2
      REWIND 30
      GO TO 50
C COMMENT CARD
200 READ (30,220)COMENT
      REWIND 30
      GO TO 50
C DATE CARD
210 M=NXTCHK(CARD2,COL)
      IF (M.EQ.BLANK) GO TO 50
      READ (30,220)DATE
220 FORMAT(10X,15A4)
      REWIND 30
      GO TO 50
C MAXPT CARD
230 J=NXTCHR(CARD2,COL)
      IF (J.EQ.BLANK) GO TO 540
      COL=COL-1
      MPT = NUMBER( CARD2, COL, MAXPT, MPT )
C
      IF (MPT.GT.30) GO TO 90
C
      GO TO 50
C PEROUT CARD
250 J=NXTCHR(CARD2,COL)
      IF (J.EQ.BLANK) GO TO 540
      COL=COL-1
      M = NUMBER ( CARD2, COL, ARRAY, ZERO )
      PEROUT = ARRAY(1)
C
      IF (M.NE.1) GO TO 90
C
      GO TO 50
C SUBCLASS CARD
270 NOSUB2=NUMBER(CARD2,COL,SUBVC2,NOSUB2)
      CALL ORDER(SUBVC2,NOSUB2)
      GO TO 50
C LAM CARD
280 J=NXTCHR(CARD2,COL)
      IF (J.EQ.BLANK) GO TO 540
      COL=COL-1
      M = NUMBER ( CARD2, COL, ARRAY, ZERO )
      LAM = ARRAY(1)
C
      IF (M.NE.1) GO TO 90
C
      GO TO 50
C
      OPTION CARD
C
290 M=FIN12(CARD2,COL,MTX)
      M = IABS( M )
C
      IF (M.EQ.0.OR.M.GT.5) GO TO 540
C
      IF M = 1, END-OF-CARD HAS BEEN REACHED
C
      GO TO (50,300,310,320,350),M
C
      IF M = 2, 'O' , OR 'ORIG'
C
300 ORIG = 1
C
      SET01590
      SET01600
      SET01610
      SET01620
      SET01630
      SET01640
      SET01650
      SET01660
      SET01670
      SET01680
      SET01690
      SET01700
      SET01710
      SET01720
      SET01730
      SET01740
      SET01750
      SET01760
      SET01770
      SET01780
      SET01790
      SET01800
      SET01810
      SET01820
      SET01830
      SET01840
      SET01850
      SET01860
      SET01870
      SET01880
      SET01890
      SET01900
      SET01910
      SET01920
      SET01930
      SET01940
      SET01950
      SET01960
      SET01970
      SET01980
      SET01990
      SET02000
      SET02010
      SET02020
      SET02030
      SET02040
      SET02050
      SET02060
      SET02070
      SET02080
      SET02090
      SET02100
      SET02110
      SET02120
      SET02130
      SET02140
      SET02150
      SET02160
      SET02170
      SET02180
      SET02190
      SET02200
      SET02210
      SET02220
      SET02230
      SET02240
      SET02250
      SET02260
      SET02270
      SET02280
      SET02290
      SET02300
      SET02310
      SET02320
      SET02330
      SET02340
      SET02350
      SET02360
      SET02370

```

FILE: SETUP8

ORIGINAL PAGE IS  
OF POOR QUALITY

```
C M = FIND12( CARD2, COL, SINVEC )
C IF (M.EQ.2) GO TO 290
C GO TO 50
C
C IF M = 3, 'T' OR 'TRANSF'
C
C 310 TRANSF = 1
C M = FIND12( CARD2, COL, SINVEC )
C IF (M.EQ.2) GO TO 290
C GO TO 50
C
C IF M = 4, 'S' --- CHECK FOR 'SCAFAC='
C 320 J = NXTCHR ( CARD2, COL )
C
C IF NEXT CHARACTER IS 'C' , ASSUME 'SCAFAC='
C IF (J .NE. CBCD) GO TO 540
C Z = FIND12( CARD2, COL, SINVEC )
C IF (Z.EQ.3) GO TO 330
C GO TO 540
C
C SCALE FACTOR OPTION : READ SCALING PAIRS, CON AND MIN , INTO
C CONMIN
C
C 330 SCAFLG = 3
C 340 Z = FIND12( CARD2, COL, UP )
C IF (Z.NE.2) GO TO 50
C NMN = FLTNUM ( CARD2, COL, CONMIN(NSF) , 2 )
C IF (NMN.NE.2) GO TO 540
C ADDNUM = NSF + 1
C IF ((NSF+NMN).GT.31) GO TO 50
C NSF = NSF + NMN
C Z = FIND12( CARD2, COL, CP )
C IF (Z.EQ.2) GO TO 340
C GO TO 540
C
C PUNCH OPTION
C 350 NPUN = 1
C
C GO TO 290
C
C MODULE STAT DECK
C 350 MK=NXTCHR(CARD2,COL)
C IF (MK.NE.BTEST(3)) GO TO 370
C
C SCAFLG = 2
C
C GO TO 50
C 370 CALL CRDSTA(ARRAY, TOP)
C
C SCAFLG = 2
```

```
SET02380
SET02390
SET02400
SET02410
SET02420
SET02430
SET02440
SET02450
SET02460
SET02470
SET02480
SET02490
SET02500
SET02510
SET02520
SET02530
SET02540
SET02550
SET02560
SET02570
SET02580
SET02590
SET02600
SET02610
SET02620
SET02630
SET02640
SET02650
SET02660
SET02670
SET02680
SET02690
SET02700
SET02710
SET02720
SET02730
SET02740
SET02750
SET02760
SET02770
SET02780
SET02790
SET02800
SET02810
SET02820
SET02830
SET02840
SET02850
SET02860
SET02870
SET02880
SET02890
SET02900
SET02910
SET02920
SET02930
SET02940
SET02950
SET02960
SET02970
SET02980
SET02990
SET03000
SET03010
SET03020
SET03030
SET03040
SET03050
SET03060
SET03070
SET03080
SET03090
SET03100
SET03110
SET03120
SET03130
SET03140
SET03150
SET03160
```

FILE: SETUP8

C	GO TO 50	SET03170
C		SET03180
C	DATAFILE POSITIONING CARD	SET03190
C		SET03200
C	380 IF (NUDTAP.AND.NUDFIL) GO TO 50	SET03210
C		SET03220
C	M = NXTCHR ( CARD2 , COL )	SET03230
C		SET03240
C	IF (M.EQ.BLANK) GO TO 50	SET03250
C		SET03260
C	IF (M .EQ. UHCD) GO TO 410	SET03270
C	IF (M .EQ. FRCD) GO TO 420	SET03280
C	390 WRITE (6,400)	SET03290
C	400 FORMAT(///// 5X, ***** DATATR/SETUP8 ***** ERROR ON INPUT DATA	SET03300
C		SET03310
C	*FILE CARD --- CONTINUING TO PROCESS INPUT ***** (//// )	SET03320
C		SET03330
C	GO TO 50	SET03340
C	410 J= FIND12(CARD2,COL,EQUVEC)	SET03350
C	IF (J.EQ.-1) GO TO 390	SET03360
C	M=NUMBER(CARD2,COL,DATEP,ZERO)	SET03370
C	COL=COL-1	SET03380
C		SET03390
C	IF (M.NE.1) GO TO 390	SET03400
C		SET03410
C	NUDTAP = .TRUE.	SET03420
C		SET03430
C	GO TO 380	SET03440
C	420 J= FIND12(CARD2,COL,EQUVEC)	SET03450
C	IF (J.EQ.-1) GO TO 390	SET03460
C	FILNO = NUMBER ( CARD2, COL, DATFIL, ZERO )	SET03470
C		SET03480
C	IF (FILNO.NE.1) GO TO 390	SET03490
C		SET03500
C	NUDFIL = .TRUE.	SET03510
C		SET03520
C		SET03530
C	DATFIL=DATFIL-1	SET03540
C	COL=COL-1	SET03550
C	GO TO 380	SET03560
C		SET03570
C	STATFILE POSITIONING CARD	SET03580
C		SET03590
C	430 M=NXTCHR(CARD2,COL)	SET03600
C	IF (M.EQ.BLANK) GO TO 50	SET03610
C		SET03620
C	IF (M .EQ. URCD) GO TO 460	SET03630
C	IF (M .EQ. FRCD) GO TO 470	SET03640
C	440 WRITE (6,450)	SET03650
C	450 FORMAT(///// 5X, ***** DATATR/SETUP8 ***** ERROR ON INPUT OR OUT	SET03660
C		SET03670
C	*PUT CARD --- CONTINUING TO PROCESS INPUT ***** (/// )	SET03680
C		SET03690
C	NUSTAP = .FALSE.	SET03700
C	NUSFIL = .FALSE.	SET03710
C		SET03720
C	GO TO 50	SET03730
C		SET03740
C	460 J= FIND12(CARD2,COL,EQUVEC)	SET03750
C	IF (J.EQ.-1) GO TO 440	SET03760
C	M=NUMBER(CARD2,COL,SAVTAP,ZERO)	SET03770
C	COL=COL-1	SET03780
C		SET03790
C	IF (M.NE.1) GO TO 440	SET03800
C		SET03810
C	*****NUSTAP = .TRUE.	SET03820
C		SET03830
C	GO TO 430	SET03840
C	470 J= FIND12(CARD2,COL,EQUVEC)	SET03850
C	IF (J.EQ.-1) GO TO 440	SET03860
C	FILST = NUMBER( CARD2, COL, STAFIL, ZERO )	SET03870
C		SET03880
C	IF (FILST.NE.1) GO TO 440	SET03890
C		SET03900
C	*****NUSFIL = .TRUE.	SET03910
C		SET03920
C	STAFIL=STAFIL-1	SET03930
C	COL=COL-1	SET03940
C	GO TO 430	SET03950



ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: SETUP8

```
C
C   RESCALE OPTION CARD
C   480 CONTINUE
      RESCAL = 1
      GO TO 50
C
C*** CODE ADDED JAN. 18, 1979 TO OUTPUT MULTI-FILE OUTPUT
C
C   600      M = NXTCHR(CARD2,COL)
      IF (M.EQ.BLANK) GO TO 50
      IF (M.EQ.UBCD) GO TO 660
      IF (M.EQ.FBCD) GO TO 670
C   660      J = FIND12(CARD2,COL,EQUVEC)
      IF (J.EQ.-1) GO TO 440
      M = NUMBER(CARD2,COL,TRFORM,ZERO)
      COL = COL - 1
      IF (M.NE.1) GO TO 440
      GO TO 600
C   670      J = FIND12(CARD2,COL,EQUVEC)
      IF (J.EQ.-1) GO TO 440
      M = NUMBER(CARD2,COL,NF,ZERO)
      COL = COL - 1
      GO TO 50
C   BIAS CONTROL CARD
C   490 CONTINUE
      NR = NXTCHR(CARD2, COL)
      IF (NR.EQ.BLANK) GO TO 540
      COL = COL - 1
      VECMAX = 16 - NBS
      NK = NBS + 1
      NBS = FLTNUM (CARD2, COL, BIAS(NK), VECMAX)
      GO TO 50
C
C *END* CARD
C   500 CONTINUE
C
C      IF ( RESCAL .GT. 0 .AND. SCAFLG .EQ. 0 ) SCAFLG = 1
C
C      IF RESCALING BY THE STATISTICAL METHOD, READ STATISTICS FROM
C      FILE ( SAVTAP ), REDUCE THE STATISTICS TO THE SET OF CHANNELS
C      SPECIFIED IN FETVC2 , AND STORE IN ARRAY .
C
C      IF ( SCAFLG .EQ. 2 ) CALL REDSAV( ARRAY, TOP, BMSWT )
C
C      IF (ORIG.EQ.0) GO TO 530
C
C      DO 510 I=1,15
      TEMP = COMENT(I)
      COMENT(I) = COVHD1(I)
C   510 COVHD1(I) = TEMP
C
C      CALL PRTCOV(ARRAY(COVAR2),ARRAY(AVAR2),VANSZ2,NOFET2,ARRAY(SUBDS2;
      *)
C
C      DO 520 I=1,15
C   520 COMENT(I) = COVHD1(I)
C
C      PRINT OUT THE INPUT TRANSFORMATION MATRIX
C
C   530 CALL WRBMM(RMAT,NOFET4,NOFET2,FETVC2)
C   SET NOCLS2=NOSUR2 FOR REST OF PROGRAM
      NOCLS2=NOSUR2
C
C      IF ( RESCAL .EQ. 0 ) SCAFLG = 0
C
C      RETURN
C
C
```

SET03960  
SET03970  
SET03980  
SET03990  
SET04000  
SET04010  
SET04020  
SET04030  
SET04040  
SET04050  
SET04060  
SET04070  
SET04080  
SET04090  
SET04100  
SET04110  
SET04120  
SET04130  
SET04140  
SET04150  
SET04160  
SET04170  
SET04180  
SET04190  
SET04200  
SET04210  
SET04220  
SET04230  
SET04240  
SET04250  
SET04260  
SET04270  
SET04280  
SET04290  
SET04300  
SET04310  
SET04320  
SET04330  
SET04340  
SET04350  
SET04360  
SET04370  
SET04380  
SET04390  
SET04400  
SET04410  
SET04420  
SET04430  
SET04440  
SET04450  
SET04460  
SET04470  
SET04480  
SET04490  
SET04500  
SET04510  
SET04520  
SET04530  
SET04540  
SET04550  
SET04560  
SET04570  
SET04580  
SET04590  
SET04600  
SET04610  
SET04620  
SET04630  
SET04640  
SET04650  
SET04660  
SET04670  
SET04680  
SET04690  
SET04700  
SET04710  
SET04720  
SET04730  
SET04740

13-21  
257

FILE: SETUP8

```
C 540 WRITE (6,55)CODE,CARDZ
C 550 FORMAT(//// SX,***** INVALID CONTROL CARD REJECTED BY DATATH/SET
C      *UP8 ***** // SX, A*,6X, 62A1 //// )
C
      GO TO 50
      END
```

```
SET04750
SET04760
SET04770
SET04780
SET04790
SET04800
SET04810
SET04820
SET04830
SET04840
```

FILE: TRANSF

```

      SUBROUTINE TRANSF
      (XT, RMAT, IDATA, TOP, IL, K, LCOMB, NSAMP, BIAS)
      IMPLICIT INTEGER (A-Z)
      REAL XT(16), RMAT(480), BIAS(16)
      INCLUDE COMRQ9.LIST
      DATA TRANSFORMATION COMMON BLOCK
      COMMON/TRBLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30)
      CSEND
      DIMENSION IDATA (TOP)
      SUBROUTINE TRANSF DOES A DATA-TRANSFORMATION USING THE
      FORMULA
      XT = IDATA * RMAT * BIAS
      XT = COMPONENT(IL), TRANSFORMED DATA VECTOR
      IDATA = INPUT DATA VECTOR ( NOFEAT X 1 )
      RMAT = TRANSFORMATION MATRIX ( LCOMB X NOFEAT )
      BIAS = ADDITIVE BIAS
      DO 10 IT=1,NOFEAT
      JSAMP = (IT - 1) * NSAMP * K
      ZCOMB = LCOMB * (IT - 1) * IL
      XT (IL) = XT (IL) *
      IDATA(JSAMP) * RMAT(ZCOMB)
      C CONTINUE
      XT(IL) = XT(IL) + BIAS(IL)
      RETURN
      END
      TRA00010
      TRA00020
      TRA00030
      TRA00040
      TRA00050
      TRA00060
      TRA000H0
      TRA00090
      TRA00100
      TRA00110
      TRA00120
      TRA00130
      TRA00140
      TRA00150
      TRA00160
      TRA00170
      TRA00180
      TRA00190
      TRA00200
      TRA00210
      TRA00220
      TRA00230
      TRA00240
      TRA00250
      TRA00260
      TRA00270

```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: TRMIST

```
      SUBROUTINE TRMIST(IDATA,AMAX,AMIN,ACON,FMAT,LCOMB,
      *PENOUT,FILE,HIS,TOP,LAR,FLDNAM,NC,VERTCS,MAX,MIN,CON,
      *BIAS)
      HISTOGRAM THE TRANSFORMED DATA AND CALCULATE THE MIN MAX AND RANGE
      FOR THIS DATA TO ALLOW RESCALING IN THE 0-255 RANGE
      IMPLICIT INTEGER(A-Z)
      REAL   BIAS(16), XT(16), PERCEN(16), MIN(16), MAX(16), CON(16)
      REAL   FMAT(400), AMIN(16), AMAX(16), ACON(16)
      REAL   XMIN, XPER, SUMFIL, DUMMY
      INCLUDE COM-K1.LIST
      INCLUDE COM-K9.LIST
      INCLUDE COM-K4.LIST
      INCLUDE COM-K5.LIST
      COMMON/INFORM/NOCLS2,NOSUM2,NOFET2,VARZ2,TOTVT2,NOFLD2,
      *AVAR2,COVAR2,CLS102,SUBNO2,SUMDS2,FLDSV2,VERTX2,
      *FFTVC2(30),SUFVC2(75),SUBPTR(75),CLSV2(60),
      *KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
      *GRPCPK(6),GRPNMS(124)
      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
      EQUIVALENCE (HED1(1),HED1(4)),(DATE(1),HEAU(22)),
      * (HED2(1),HEAU(30)),(COMENT(1),HEAU(48))
      COMMON/GLOBAL/HEAD(63),MARTAP,DATEP,SAVTAP,HMFILE,HMKEY,
      *HISFIL,HISKEY,INFORM,ERIPTR,EXPKY,MAPUNT,NOFILE,
      *DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVL
      *NMSTUN,NMSTFI,SCROUT,MAPFIL
      *DOTUNT,DOTFIL,NCHPAS,TRNSFL,HMTRFL,HISTFL,PCHUNT,
      *CRDUNT,PRDUNT,WANDIO
      DATA TRANSFORMATION COMMON BLOCK
      COMMON/TRPLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30)
      CSEND
      DIMENSION FILHIS(LCOMB,101), IDATA(TOP), TOTPTS(16),
      * VERTCS(2,11),FL(8),FLDINP(6)
      READ THE COORDINATES ( VERTICES ) OF THE FIELD, FOR THE
      DATA TO BE TRANSFORMED
      10 LAR=LAR*AD(FLDNAM,VERTCS,FLDINF,NC)
      IF (LAR.EQ.0) GO TO 210
      IF (LAR.LF.-1) GO TO 10
      POSITION THE INPUT DATA TAPE AND READ IN THE HEADER RECORD
      CALL TAPHOR( DATEP, DATFIL )
      DO 20 I=1,LCOMB
      MAX(I)=AMAX(I)
      MIN(I)=AMIN(I)
      20 CON(I)=ACON(I)
      NSAMP=(FLDINF(5)-FLDINF(4))/FLDINF(6)+1
      LINES=(FLDINF(2)-FLDINF(1))/FLDINF(3)+1
      DUMMY = (LINES*NSAMP)/2000
      ALP=SORT(DUMMY)
      IF (ALP.LE.1) ALP=1
      FLDINP(1)=FLDINF(1)
      FLDINP(2)=FLDINF(2)
      FLDINP(4)=FLDINF(4)
      FLDINP(5)=FLDINF(5)
      FLDINP(3)=ALP
      FLDINP(6)=ALP
      LINES=(FLDINP(2)-FLDINP(1))/FLDINP(3)+1
      NSAMP=(FLDINP(5)-FLDINP(4))/FLDINP(6)+1
      CALL FLDFMT(FLDINP,FETVEC,NOFEAT)
      DO 30 I=1,LCOMB
      TOTPTS(I) = 0
      DO 30 J=1,101
      30 FILHIS(I,J)=0
      DO 130 I=1,LINES
      CALL LINFMT(IDATA,ENDTAP)
      IF (ENDTAP.NE.0) GO TO 140
```

TRM00010  
TRM00020  
TRM00030  
TRM00040  
TRM00050  
TRM00060  
TRM00070  
TRM00080  
TRM00090  
TRM00100  
TRM00110  
TRM00120  
TRM00130  
TRM00140  
TRM00150  
TRM00160  
TRM00170  
TRM00180  
TRM00190  
TRM00200  
TRM00210  
TRM00220  
TRM00230  
TRM00240  
TRM00250  
TRM00260  
TRM00270  
TRM00280  
TRM00290  
TRM00300  
TRM00310  
TRM00320  
TRM00330  
TRM00340  
TRM00350  
TRM00360  
TRM00370  
TRM00380  
TRM00390  
TRM00400  
TRM00410  
TRM00420  
TRM00430  
TRM00440  
TRM00450  
TRM00460  
TRM00470  
TRM00480  
TRM00490  
TRM00500  
TRM00510  
TRM00520  
TRM00530  
TRM00540  
TRM00550  
TRM00560  
TRM00570  
TRM00580  
TRM00590  
TRM00600  
TRM00610  
TRM00620  
TRM00630  
TRM00640  
TRM00650  
TRM00660  
TRM00670  
TRM00680  
TRM00690  
TRM00700  
TRM00710  
TRM00720  
TRM00730  
TRM00740  
TRM00750  
TRM00760  
TRM00770  
TRM00780  
TRM00790

FILE: TRHIST

```
IF (I.NE.1) GO TO 40
ILIN=FLDINP(1)
GO TO 50
40 ILIN=ILIN+FLDINP(3)
50 CONTINUE
CALL FDLINT(VERTCS,NC,FL,ILIN,NS,JJ)
DO 110 K=1,NSAMP
KP=(K-1)*FLDINP(6)+FLDINP(4)
DO 100 LK=1,JJ,2
LKPI = LK + 1
IF (KP.LT.FL(LK)) GO TO 110
IF (KP.GT.FL(LKPI)) GO TO 90
DO 80 J=1.LCOMB
XT(J)=0.

CALL TRANSF TO DO A DATA TRANSFORMATION

C CALL TRANSF
  (XT, BMAT, IDATA, TOP, J, K, LCOMB, NSAMP, BIAS)

HISTOGRAM THE TRANSFORMED DATA ( USING TRANSFORMED DATA MAX
AND MIN AND SCALE FACTOR, CON, COMPUTED IN SURR. MAXMAT
TO OBTAIN THE HISTOGRAM "BIN LEVEL" FOR EACH TRANSFORMED
DATA POINT )

IF (XT(J).LE.MIN(J)) GO TO 60
IF (XT(J).GE.MAX(J)) GO TO 70
DPT=(XT(J)-MIN(J))/CON(J)+1

IF ( DPT .LE. 0 ) DPT = 1
IF ( DPT .GT. 101 ) DPT = 101

FILHIS(J,DPT)=FILHIS(J,DPT)+1
GO TO 80
60 FILHIS(J,1)=FILHIS(J,1)+1
GO TO 80
70 FILHIS(J,101) = FILHIS(J,101) + 1
80 TOTPTS(J)=TOTPTS(J)+1
GO TO 110
90 IF (LKPI.GE.JJ) GO TO 120
100 CONTINUE
110 CONTINUE
120 CONTINUE
130 CONTINUE
140 CONTINUE

ELIMINATE PEROUT/? OF POINTS FROM UPPER AND LOWER TAILS OF
THE TRANSFORMED DATA DISTRIBUTION --- OBTAIN THE REVISED MAX
AND SCALING PARAMETERS CON AND MIN AFTER APPLICATION OF PEROUT

C XPER=PEROUT/200.
CALCULATE MIN,MAX,AND CON ARRAYS
DO 150 I=1.LCOMB
150 PERCEN(I)=XPER*TOTPTS(I)
DO 200 I=1.LCOMB
XMIN=MIN(I)
SUMFIL=0.
DO 160 J=1.101
SUMFIL=SUMFIL+FILHIS(I,J)
IF (SUMFIL.LT.PERCEN(I)) GO TO 160
MIN(I)=(J-1)*CON(I)+XMIN
GO TO 170
160 CONTINUE
170 SUMFIL=0.
J=102
175 J=J-1
SUMFIL=SUMFIL+FILHIS(I,J)
IF (SUMFIL.LT.PERCEN(I)) GO TO 180
JMI = J - 1
MAX(I) = JMI * CON(I) + XMIN
GO TO 190
180 IF (J.GT.1) GO TO 175
190 CONTINUE
CON(I) = 255. / (MAX(I) - MIN(I))
200 CONTINUE
```

ORIGINAL PAGE IS  
OF POOR QUALITY

TRH00800  
TRH00810  
TRH00820  
TRH00830  
TRH00840  
TRH00850  
TRH00860  
TRH00870  
TRH00880  
TRH00890  
TRH00900  
TRH00910  
TRH00920  
TRH00930  
TRH00940  
TRH00950  
TRH00960  
TRH00970  
TRH00980  
TRH00990  
TRH01000  
TRH01010  
TRH01020  
TRH01030  
TRH01040  
TRH01050  
TRH01060  
TRH01070  
TRH01080  
TRH01090  
TRH01100  
TRH01110  
TRH01120  
TRH01130  
TRH01140  
TRH01150  
TRH01160  
TRH01170  
TRH01180  
TRH01190  
TRH01200  
TRH01210  
TRH01220  
TRH01230  
TRH01240  
TRH01250  
TRH01260  
TRH01270  
TRH01280  
TRH01290  
TRH01300  
TRH01310  
TRH01320  
TRH01330  
TRH01340  
TRH01350  
TRH01360  
TRH01370  
TRH01380  
TRH01390  
TRH01400  
TRH01410  
TRH01420  
TRH01430  
TRH01440  
TRH01450  
TRH01460  
TRH01470  
TRH01480  
TRH01490  
TRH01500  
TRH01510  
TRH01520  
TRH01530  
TRH01540  
TRH01550  
TRH01560  
TRH01570  
TRH01580

FILE: TRMIST

C  
C  
C  
C  
C  
C

RETURN THE SCALING PARAMETERS. CON AND MIN , REQUIRED TO  
RESCALE THE TRANSFORMED DATA TO THE RANGE , 0 - 255 .

210 RETURN  
END

TRH01590  
TRH01600  
TRH01610  
TRH01620  
TRH01630  
TRH01640  
TRH01650  
TRH01660

~~13-26~~

262

14. TRSTAT PROCESSOR

FILE: TRSTAT

	SUBROUTINE TRSTAT(ARRAY, TOP)	TRS00010
	IMPLICIT INTEGER(A-Z)	TRS00020
	DIMENSION ARRAY(TOP)	TPS00030
C	INCLUDE COMMK1.LIST	TPS00040
C	INCLUDE COMMK6.LIST	TRS00050
	COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,	COM00010
	AVAR2, COVAR2, CLS102, SUBNO2, SURDS2, FLDSV2, VERTX2,	COM00020
	FETVC2(30), SUBVC2(75), SURPTH(75), CLSVC2(60),	COM00030
	FEPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),	COM00040
	GRPCHK(61), GROUPS(124)	COM00050
	COMMON/GLOBAL/HEAD(63), MAPTAP, DATAE, SAVTAP, BMFILE, BMKEY,	COM00010
	HISFIL, HISKEY, TRFORM, EPIPTP, ERPKEY, MAPUNT, NOFILF,	COM00020
	DRIMAD, DRMWD5, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL	COM00030
	.NHSTUN, NHSTFI, SCTRUN, MAPFIL	COM00040
	.DOTUNT, DOTFIL, NCHPAS, TRNSFL, RMTFL, HISTFL, PCHUNT,	COM00050
	CRDUNT, PRUNT, RANDIO	COM00060
CSEND	REAL AMAT(1800), R(30)	TRS00070
	CALL SFUP9(ARRAY, TOP, AMAT, ROW, IP, TRAN, B)	TPS00080
	CALL TRAMTX(ARRAY, TOP, AMAT, ROW, IP, TRAN, B)	TPS00090
	RETURN	TRS00100
	END	TPS00110
		TPS00120

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: AMFIL

```
SUBROUTINE AMFIL(ROW,COLUMN,AMAT,VEC,B)
IMPLICIT INTEGER(A-Z)
C INCLUDE COMMON6
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILF,
* DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* .NHSTUN,NHSTFI,SCTRUN,MAPFIL
* .DOTUNT,DOTFIL,NCHPAS,TRANSFL,BMTRFL,HISTFL,PCHUNT,
* CROUNT,PRUNT,WANDIO
CSEND
REAL AMAT(1),B(30)
DIMENSION VEC(1)
1 READ(2,2) ROW,COLUMN,(VEC(I),I=1,COLUMN)
IK=ROW*COLUMN
READ(2,3) (AMAT(I),I=1,IK)
READ(2,3) (B(I),I=1,ROW)
RETURN
2 FORMAT(5X,I2,5X,I2,2X,30I2)
3 FORMAT(5X,5E15.8)
END
```

AMF00010  
AMF00020  
AMF00030  
AMF00040  
AMF00050  
AMF00060  
AMF00070  
AMF00080  
AMF00090  
AMF00100  
AMF00110  
AMF00120  
AMF00130  
AMF00140  
AMF00150  
AMF00160  
AMF00170  
AMF00180  
AMF00190  
AMF00200

~~10-2~~  
264



FILE: AMFILE

```
C* SURROUTINE AMFILE (ROW,NOCHAN,CHNVEC,AMAT,BVEC) AMF00010
C* AMFILE WILL READ INTO CORE THE A-MATRIX AND B VECTOR FROM UNIT AMF00020
C* 7 AMF00030
C* AMF00040
C* AMF00050
C* IMPLICIT INTEGER (A-Z) AMF00060
C* REAL AMAT(1),BVEC(1) AMF00070
C* INCLUDE COMR6 AMF00080
C* COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,RMKEY, AMF00090
C* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE, AMF00100
C* DRUMAD,DRM=DS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL AMF00110
C* .NHSTUN,NHSTFI,SCTHUN,MAPFIL AMF00120
C* .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, AMF00130
C* CRDUNT,PRUNT,HANDIG AMF00140
CSEND AMF00150
C* DIMENSION CHNVEC(1) AMF00160
C* READ(7) ROW,NOCHAN,(CHNVEC(I),I=1,NOCHAN) AMF00170
C* TOTAL = NOCHAN**2 AMF00180
C* READ(7) ( AMAT(I),I=1,TOTAL),(BVEC(I),I=1,NOCHAN) AMF00190
C* RETURN AMF00200
C* END AMF00210
AMF00220
```



FILE: SETUP9

```
6 NOSUB2=NUMBER(CARD,COL,SURVC2,NOSUB2)
CALL ORDER(SUBVC2,NOSUB2)
GO TO 13
C READ IN A-MATRIX
  R CONTINUE
  AMTXXSW = -1
  CALL AMFIL(POW,NOCHAN,AMAT,CHNVEC,8)
  GO TO 13
C OPTIONS
  9 J=NXTCHR(CARD,COL)
  IF(J.EQ.BLANK) GO TO 16
  IF(J.EQ.P) IP=1
  IF(J.EQ.O) ORIG=1
  IF(J.EQ.T) TRAN=1
  M=FIND12(CARD,COL,SINVEC)
  IF(M.NE.2) GO TO 13
  GO TO 9
C MODULE CARD
  11 J=NXTCHR(CARD,COL)
  IF(J.EQ.F) GO TO 13
  CALL CRDSTA(ARRAY,TOP)
  GO TO 13
C*
C* STAT FILE CARD
C*
  20 M = NXTCHR(CARD,COL)
  21 IF (M .EQ. BLANK) GO TO 13
  IF (M .EQ. IRCD) GO TO 25
  IF (M .EQ. ORCD) GO TO 30
  22 WRITE(4,222)
  222 FORMAT(' EMPOR ON STAT FILE CARD')
  GO TO 13
  25 J = FIND12(CARD,COL,SLASH)
  IF (J .EQ. -1) GO TO 22
  23 M = NXTCHR(CARD,COL)
  IF (M .EQ. URCD) GO TO 26
  IF (M .EQ. FRCD) GO TO 28
  GO TO 21
  26 J = FIND12(CARD,COL,SINVEC)
  IF (J .NE. 3) GO TO 22
  M = NUMBER(CARD,COL,SAVTAP,ZERO)
  COL = COL - 1
  GO TO 23
  28 J = FIND12(CARD,COL,SINVEC)
  IF (J .NE. 3) GO TO 22
  M = NUMBER(CARD,COL,STAFIL,ZERO)
  STAFIL = STAFIL - 1
  COL = COL - 1
  GO TO 23
  30 J = FIND12(CARD,COL,SLASH)
  IF (J .EQ. -1) GO TO 22
  32 M = NXTCHR(CARD,COL)
  IF (M .EQ. URCD) GO TO 34
  IF (M .EQ. FRCD) GO TO 36
  GO TO 21
  34 J = FIND12(CARD,COL,SINVEC)
  IF (J .NE. 3) GO TO 22
  M = NUMBER(CARD,COL,ASAV,ZERO)
  COL = COL - 1
  GO TO 32
  36 J = FIND12(CARD,COL,SINVEC)
  IF (J .NE. 3) GO TO 22
  M = NUMBER(CARD,COL,ASAVFL,ZERO)
  ASAVFL = ASAVFL - 1
  COL = COL - 1
  GO TO 32
C*
C* CHANNEL CARD
C*
  40 M = NXTCHR(CARD,COL)
  IF (M .EQ. BLANK) GO TO 16
  COL = COL - 1
  NOFET2 = NUMBER(CARD,COL,FETVC2,NOFET2)
  CALL ORDER(FETVC2,NOFET2)
  GO TO 13
C *FND* CARD
  60 CALL RFDOSAV(ARRAY,TOP,RMSWT)
  VARS72=NOFET2*(NOFET2+1)/2
  IF (AMTXXSW .EQ. -1) GO TO 65
```

ORIGINAL PAGE IS  
OF POOR QUALITY

T00800  
T00810  
T00820  
T00830  
T00840  
T00850  
T00860  
T00870  
T00880  
T00890  
T00900  
T00910  
T00920  
T00930  
T00940  
T00950  
T00960  
T00970  
T00980  
T00990  
T01000  
T01010  
T01020  
T01030  
T01040  
T01050  
T01060  
T01070  
T01080  
T01090  
T01100  
T01110  
T01120  
T01130  
T01140  
T01150  
T01160  
T01170  
T01180  
T01190  
T01200  
T01210  
T01220  
T01230  
T01240  
T01250  
T01260  
T01270  
T01280  
T01290  
T01300  
T01310  
T01320  
T01330  
T01340  
T01350  
T01360  
T01370  
T01380  
T01390  
T01400  
T01410  
T01420  
T01430  
T01440  
T01450  
T01460  
T01470  
T01480  
T01490  
T01500  
T01510  
T01520  
T01530  
T01540  
T01550  
T01560  
T01570  
T01580

FILE: SETUP9

```
CALL AMFILE(ROW,NOCHAN,CHNVEC,AMAT,B)
65 CONTINUE
C WRITE A-MATRIX
  CALL WRTAMT(AMAT,ROW,NOFET2,FETVC2,B)
  IF(ORIG.EQ.1) CALL PRTCOV(ARRAY(COVAR2),ARRAY(AVAR2),VARSZ2,
    * NOFET2,ARRAY(SUBDS2))
  IF (NOFET2 .EQ. NOCHAN) GO TO 13
  WRITE(6,100)NOFET2,NOCHAN
100 FORMAT(' NO. OF CHANNELS FROM STAT FILE DOES NOT EQUAL THE /
  * NO. OF CHANNELS ON A-MATRIX FILE.*/ CHANNELS ON STAT FILE = ',
  * I2/' CHANNELS ON A-MATRIX = ', I2)
  CALL CMERR
C SEND* CARD
  RETURN
16 WRITE(6,19) CARD
19 FORMAT(' INVALID CONTROL CARD REJECTED ***SETUP9***')
  GO TO 13
END
```

ST01590  
ST01600  
ST01610  
ST01620  
ST01630  
ST01640  
ST01650  
ST01660  
ST01670  
ST01680  
ST01690  
ST01700  
ST01710  
ST01720  
ST01730  
ST01740  
ST01750  
ST01760



FILE: TRAMTX

```
WRITE (PUNCH,40) (ARRAY(L),L=SUBDS2,LL)
40 FORMAT((SUBDES ',10(A4,3X)))
23 CONTINUE
C FOR EACH SIMCLASS COMPUTE TRANS STATS AND OUTPUT THEM
J=AVAR2
K=COVAR2
L=1
KKK=1
DO 20 I=1,NOSUB2
  RR=ROW+(ROW+1)/2
  P=SUBVC2(I)
  KCNT=KKK+RR-1
C MULTIPLY A-MATRIX BY MEAN VECTOR
  CALL MATVEC (AMAT,ARRAY(J),AMEAN(L),ROW,NOFET2)
C ADD P VECTOR TO GET TRANSFORMED MEANS
  KM=0
  LL=L+ROW-1
  DO 21 I21=L,LL
    KM=KM+1
    21 AMEAN(I21)=AMEAN(I21)+8(KM)
C COMPUTE TRANSFORMED COVAR MATRIX
C A-MATR * COVAR * A-MATR TRANSPOSE
C MULTIPLY A-MATR * COVAR AND STORE IN C
  CALL MTMLS6 (AMAT,ARRAY(K),C,ROW,NOFET2)
C MULTIPLY C BY A TRANSPOSE AND STORE IN ARRAY
  CALL MTMDAT (C,AMAT,CC,ROW,NOFET2,ROW+DD,ARRAY(KKK))
  WRITE (ASAV) KEPPTS (P), (ARRAY (II), II=KKK, KCNT),
    * (AMEAN (II), II=L, LL)
  IF (IP.FO.0) GO TO 22
  WRITE (PUNCH,95) KEPPTS (P)
95 FORMAT ('NOPTS ',6X,1H)
  WRITE (PUNCH,96) (AMEAN (II), II=L, LL)
96 FORMAT ('MEANS ',5E15.8)
  WRITE (PUNCH,97) (ARRAY (II), II=KKK, KCNT)
97 FORMAT ('COVAR ',5E15.8)
22 CONTINUE
  J=J+ROW
  L=LL+1
  KKK=KKK+RR
  K=K+VAR522
20 CONTINUE
IF (TRAN.NE.1) RETURN
CV1=ROW+(ROW+1)/2
CALL PRTCOV (ARRAY (1), AMEAN (1), CV1, ROW, NSUB (1))
ENDFILE ASAV
PEWIND ASAV
END
```

TRA00800  
TRA00810  
TRA00820  
TRA00830  
TRA00840  
TRA00850  
TRA00860  
TRA00870  
TRA00880  
TRA00890  
TRA00900  
TRA00910  
TRA00920  
TRA00930  
TRA00940  
TRA00950  
TRA00960  
TRA00970  
TRA00980  
TRA00990  
TRA01000  
TRA01010  
TRA01020  
TRA01030  
TRA01040  
TRA01050  
TRA01060  
TRA01070  
TRA01080  
TRA01090  
TRA01100  
TRA01110  
TRA01120  
TRA01130  
TRA01140  
TRA01150  
TRA01160  
TRA01170  
TRA01180  
TRA01190  
TRA01200  
TRA01210  
TRA01220  
TRA01230  
TRA01240  
TRA01250  
TRA01260

FILE: WRTAMT

```

SUBROUTINE WRTAMT(AMAT,ROW,COLUMN,FETVC2*8)
IMPLICIT INTEGER(A-Z)
C INCLUDE COM-K4.LIST
C INCLUDE COM-K4.LIST
DIMENSION HEAD(15),HEAD(15),DATE(3),COMMENT(15)
EQUIVALENCE (HEAD(1),HEAD(4)),(DATE(1),HEAD(22)),
              (HEAD(1),HEAD(30)),(COMMENT(1),HEAD(48))
2 COMMON/GLOBAL/HEAD(63),WRTAMT,DATE,SAVTAP,MAPFILE,MMKEY,
  * HISFIL,HISKFY,TRFORM,EN1,TP,FMPKEY,MAPJUNT,NOFILE,
  * DRIMAN,DRMWDOS,PAGS17,DATEFIL,STAFIL,ASAV,ASAVFL
  * NHSTUN,INSTFI,SCTRUN,MAPFIL
  * DOTUNT,DATEFIL,CHPAS,TRNSFL,HMTXFL,HISTFL,PCHUNT,
  * CPDUNT,PNTUNT,RANDIO
CSEND
REAL AMAT(ROW,COLUMN),R(1)
DIMENSION FETVC2(1),CH(2)
DATA CH/CH(' ',' ')
WRITE(6,HEAD)
WRITE(6,1) (I=1,ROW)
11 FORMAT('E-VECTOR =',//10X,12F10.4//10X,12E10.4//10X,6E10.4)
WRITE(6,1) FOR COLUMN
IR=1
IK=1
5 IF(IK.GT.COLUMN) IK=COLUMN
WRITE(6,2) (CH,I=1,IK)
WRITE(6,3) (FETVC2(I),I=IR,IK)
WRITE(6,6)
DO 10 J=1,ROW
10 WRITE(6,4) J,(AMAT(I,I),I=1,IK)
IF(IK.EQ.COLUMN) RETURN
IR=IK+1
IK=IK+1
GO TO 5
1 FORMAT(///57X,'A-MATRIX'//
  * 50X,'NO. OF COMBINATIONS =',I3/
  * 50X,'NO. OF CHANNELS =',I3)
2 FORMAT(/12X,12(2A4,2X))
3 FORMAT(14X,14X,11(12,4X),12)
6 FORMAT(' COMBINATIONS')
4 FORMAT(1X,15,4X,12(1A,4X,3))
END
WRT00010
WRT00020
WRT00030
WRT00040
WRT00050
WRT00060
WRT00070
WRT00080
WRT00090
WRT00100
WRT00110
WRT00120
WRT00130
WRT00140
WRT00150
WRT00160
WRT00170
WRT00180
WRT00190
WRT00200
WRT00210
WRT00220
WRT00230
WRT00240
WRT00250
WRT00260
WRT00270
WRT00280
WRT00290
WRT00300
WRT00310
WRT00320
WRT00330
WRT00340
WRT00350
WRT00360
WRT00370
WRT00380
WRT00390
WRT00400
WRT00410
```

ORIGINAL PAGE IS  
OF POOR QUALITY

## 15. NDHIST PROCESSOR

FILE: NDHIST

```
C SUBROUTINE NDHIST(ARRAY, TOP)
C NDHIST IS THE DRIVER ROUTINE FOR THE N-DIMENSIONAL PROCESSOR
C TWO LARGE ARRAYS ARE USED - PLANK COMMON ARRAY CALLED ARRAY(TOP)
C AND AN ARRAY CALLED HIST(LIMIT)
C
C DATA LIMIT/12000/
C
C DIMENSION HIST(12000)
C DIMENSION ARRAY(1)
C
C READ IN CONTROL CARDS
C CALL SETIO(LIMIT)
C
C NDHIST IS THE ORGANIZER
C CALL NDHST1(HIST, ARRAY(1), ARRAY(201), ARRAY(801),
C LIMIT, ARRAY(1), TOP)
C
C RETURN
C
C END
```

CM

ORIGINAL PAGE IS  
OF POOR QUALITY



FILE: ADDRES

```
SUBROUTINE ADDRES(TOP,NSAMP,NOFEAT,REGIN,REGIND)
IMPLICIT INTEGER (A-Z)
C
C
INCLUDE CMHR11.LIST
INCLUDE CNT11.LIST
COMMON /NDIM/NCLRCH,CLRVEC(30),MAXVEC,MAPKEY,
* CLASS,SURCLS,FIELD,MEANSW,NOVEC,FLDINF(6),SIZE,TOTMNS
* ,CNT1,CNT2,ID1,ID2,COLOR1,COLOR2,RUFLEN,ID3,COLOR3,NODUMP
* ,IDATA,TOTVEC
COMMON BLOCK NDIM,IS USED ONLY BY THE N-DIMENSIONAL HISTOGRAM
PROCESSOR
DEFINITIONS
NCLRCH - NO. OF COLOR CHANNELS
CLRVEC - ARRAY CONTAINING COLOR CHANNELS
MAXVEC - MAXIMUM NO. OF VECTORS ARRAY HIST CAN STORE
CLASS - KEY INDICATING FIELDS WILL BE GROUPED ON CLASS BASES
SURCLS - KEY INDICATING FIELDS WILL BE GROUPED ON SURCLASS BASES
FIELD - KEY INDICATING FIELDS WILL BE GROUPED ON FIELD BASES
MEANSW - KEY INDICATING MEANS FOR INPUT FIELDS WILL BE COMPUTED
NOVEC - NO. OF UNIQUE VECTOR HISTOGRAMMED
FLDINF - ARRAY CONTAINING RECTANGULAR FIELD COORDINATES AROUND THE
INPUT FIELDS
SIZE - NUFFET2/4 - NO. OF COMPUTER WORDS TO STORE A PACKED
HISTOGRAMMED VECTOR
TOTMNS - TOTAL NO. OF ELEMENTS IN ARRAY CONTAINING MEANS
CNT2 - BEGINNING DRUM ADDRESS FOR STORING FREQUENCY
ID1 - ADDRESS FOR STORING ID CODES IN ARRAY
ID2 - BEGINNING DRUM ADDRESS FOR STORING ID CODES
COLOR1 - ADDRESS FOR STORING COLOR CODES IN ARRAY
COLOR2 - BEGINNING DRUM ADDRESS FOR STORING COLOR CODES
RUFLEN - AMOUNT OF STORAGE AVAILABLE FOR STORING ID/COLOR CODES
ID3 - ACCUMULATIVE ID CODE DRUM ADDRESS
COLOR3 - ACCUMULATIVE COLOR CODE DRUM ADDRESS
NODUMP - NO OF TIMES ID/COLOR CODES WERE DUMPED ON DRUM
IDATA - ADDRESS FOR STORING IMAGERY DATA IN ARRAY
TOTVEC - TOTAL NO. OF VECTORS IN THE AREA HISTOGRAMMED
CONTINUE
C GRAYMAP AND HIST COMMON BLOCK
CSEND
BASE ADDRESSES FOR ARRAY
ID1 = 350
IDATA = TOP - (NSAMP*NOFEAT + 1)
IF (IDATA .GT. 6400) GO TO 110
WRITE(6,100)
100 FORMAT(/ ' TOO MUCH DATA REQUESTED. REDUCE NO. OF SAMPLES PER SCAN
* LINE// AND/OR NO. OF CHANNELS')
CALL CMHR2
110 CONTINUE
REMO = 3000
IF (NCLRCH .NE. 0) RUFLEN = REMO / 2
IF (NCLRCH .EQ. 0) RUFLEN = REMO
IF (MEANSW .EQ. 1) RUFLEN = REMO
COLOR1 = ID1 + RUFLEN
DRUM ADDRESSES
CNT2 = REGIN
ID2 = CNT2 + MAXVEC
COLOR2 = ID2 + MAXVEC
ORIGINAL DRUM START ADDRESSES
REGIN = COLOR2 + MAXVEC
ID3 = ID2
COLOR3 = COLOR2
RETURN
END
```

ORIGINAL PAGE IS  
OF FOUR QUALITY

FILE: FLOCLS

```

SUBROUTINE FLOCLS(FIELDS,STAMNT,*,*,*,IPT,VERTEX)
FLOCLS GROUPS THE FIELDS CARDS ON A CLASS BASES FOR PROCESSING
IMPLICIT INTEGER (A-Z)
INCLUDE CMXK1.LIST
INCLUDE CMXK11.LIST
COMMON/INFORM/NOCLS2, NOSUR2, NOFFT2, VAHSZ2, TOTVT2, NOFLD2,
      AVA2, COVAR2, CLS102, SURNO2, SURNS2, FLOS2, VERTX2,
      FEVC2(30), SUHVC2(75), SUHPT2(75), CLSVC2(60),
      KFPPT5(40), NOGRP, GRPNAM(60), GRPDEX(61),
      GRPCHK(41), GROUPS(124)
COMMON /ND14/NCLRCH, CLAVEC(30), MAXVEC, MAPKEY,
      CLASS, SUHCLS, FIELD, MEANSW, NOVEC, FLDINF(5), SIZE, TOTMNS,
      CNTR1, CNTR2, ID1, ID2, COLOR1, COLOR2, HUFLEN, ID3, COLOR3, NODUMP,
      IDATA1, TOTVEC
CSEND
DIMENSION FIELDS(4,1), VERTEX(1)
LOGICAL SWITCH
DATA SWITCH/.TRUE./
IF (NOFLD2 .EQ. 0) IPT = 1
IF (NOFLD2 .EQ. 0) GO TO 75
IPT = IPT + FIELDS(4,NOFLD2)*2
75 CONTINUE
GO TO (40,100), STAMNT
80 I = LREAD(FIELDS(1,NOFLD2+1), VERTEX(IPT), FLDINF(1), FIELDS(4,NOFLD2+1))
C
      WAS CLASS, SUHCLASS, FIELD, OR SEND* ENCOUNTERED
      IF (I .EQ. -1) GO TO 90
      IF (I .EQ. -2) GO TO 120
      IF (I .EQ. 1) GO TO 130
      IF (I .EQ. 0) GO TO 140
C
      CLASS CARD
80 IF (SWITCH) GO TO 100
STAMNT = 2
RETURN 2
100 READ(30,110) CLSVC2(1)
      DEVID = 30
      IPT = 1
      NOCLS2 = 1
      NOFLD2 = 0
      NOSUR2 = 0
      SWITCH = .FALSE.
      GO TO 80
C
      SURCLASS CARD
110 FORMAT(10X,A4)
120 NOSUR2 = NOSUR2 + 1
      READ(30,110) SURVC2(NOSUR2)
      DEVID = 30
      GO TO 80
C
      FIELD CARD
130 NOFLD2 = NOFLD2 + 1
      STAMNT = 1
      RETURN 1
C
      SEND*
140 RETURN 3
C
      FND

```

FLO00010  
FLO00020  
FLO00030  
FLO00040  
FLO00050  
FLO00060  
FLO00070  
FLO00080  
FLO00090  
FLO00100  
FLO00110  
FLO00120  
FLO00130  
FLO00140  
FLO00150  
FLO00160  
FLO00170  
FLO00180  
FLO00190  
FLO00200  
FLO00210  
FLO00220  
FLO00230  
FLO00240  
FLO00250  
FLO00260  
FLO00270  
FLO00280  
FLO00290  
FLO00300  
FLO00310  
FLO00320  
FLO00330  
FLO00340  
FLO00350  
FLO00360  
FLO00370  
FLO00380  
FLO00390  
FLO00400  
FLO00410  
FLO00420  
FLO00430  
FLO00440  
FLO00450  
FLO00460  
FLO00470  
FLO00480  
FLO00490  
FLO00500  
FLO00510  
FLO00520  
FLO00530  
FLO00540  
FLO00550  
FLO00560  
FLO00570  
FLO00580  
FLO00590  
FLO00600  
FLO00610  
FLO00620  
FLO00630  
FLO00640  
FLO00650  
FLO00660  
FLO00670  
FLO00680



FILE: FLDMEN

```
C* SURROUTINE FLOWEN(IDATA,J,NSAMP,NOFEAT,MEANS,RGCHAN,N) FLD00010
C* FLOWEN COMPUTES THE FIELD MEANS FLD00020
C* FLD00030
C* FLD00040
C* FLD00050
C* FLD00060
C* FLD00070
C* FLD00080
C* FLD00090
C* FLD00100
C* FLD00110
C* FLD00120
C* FLD00130
C* FLD00140
C* FLD00150
C* FLD00160
C* FLD00170
C* FLD00180
C* FLD00190
C* FLD00200
C* FLD00210
C* FLD00220
C* FLD00230
C* FLD00240
C* FLD00250
C* FLD00260
C* FLD00270
C* FLD00280
C* FLD00290
C* FLD00300
C* FLD00310
C* FLD00320
C* FLD00330

SUBROUTINE FLOWEN(IDATA,J,NSAMP,NOFEAT,MEANS,RGCHAN,N)
FLOWEN COMPUTES THE FIELD MEANS
IMPLICIT INTEGER (A-Z)
REAL MEANS,RND
INCLUDE COMP1.LIST
INCLUDE CMRK1.LIST
COMMON/INF/CLM/NOCLS?,NOSUR?,NOFET?,VARSL?,TOTVT?,NOFLD?,
* AVAR?,COVAR?,CLSID?,SURN?,SURDS?,FLDSV?,VERTX?,
* FETVC?(30),SIBVC?(75),SURPTR(75),CLSVC?(60),
* KEPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
COMMON /NDIM/NCLRCH,CLRVEC(30),MAXVEC,MAPKEY,
* CLASS,SURCLS,FIELD,MEANSW,NOVEC,FLDINF(6),SIZE,TOTMNS
* ,CNTR1,CNTR2,ID1,ID2,COLOR1,COLOR2,RUFLEN,ID3,COLOR3,NODUMP
* ,IDATA1,TOTVEC
C$END
DIMENSION IDATA(NSAMP,NOFEAT),MEANS(NCLRCH,1)
MEANS = ( (N-1)*OLD MEAN )/N + DATA PT./N
RND = (FLOAT(N)-1.0)/FLOAT(N)
ICHAN = 0
DO 100 K = RGCHAN,NOFEAT
ICHAN = ICHAN + 1
100 MEANS(ICHAN,NOFLD?) = RND*MEANS(ICHAN,NOFLD?) +
* FLOAT(IDATA(J,K))/FLOAT(N)
RETURN
END
```



FILE: NDHST1

```

SUBROUTINE NDHST1(HIST,FIELDS,      MEANS,VERTEX,LIMIT,ARRAY,TOP) NDH00010
C* NDHST1 SETS UP THE LOGIC FOR HISTOGRAMMING THE DATA AND WRITING NDH00020
C* THE NDM FILE NDH00030
C* NDH00040
C* NDH00050
C* NDH00060
C* NDH00070
C* NDH00080
C* NDH00090
C* NDH00100
C* NDH00110
C* NDH00120
C* NDH00130
C* NDH00140
C* NDH00150
C* NDH00160
C* NDH00170
C* NDH00180
C* NDH00190
C* NDH00200
C* NDH00210
C* NDH00220
C* NDH00230
C* NDH00240
C* NDH00250
C* NDH00260
C* NDH00270
C* NDH00280
C* NDH00290
C* NDH00300
C* NDH00310
C* NDH00320
C* NDH00330
C* NDH00340
C* NDH00350
C* NDH00360
C* NDH00370
C* NDH00380
C* NDH00390
C* NDH00400
C* NDH00410
C* NDH00420
C* NDH00430
C* NDH00440
C* NDH00450
C* NDH00460
C* NDH00470
C* NDH00480
C* NDH00490
C* NDH00500
C* NDH00510
C* NDH00520
C* NDH00530
C* NDH00540
C* NDH00550
C* NDH00560
C* NDH00570
C* NDH00580
C* NDH00590
C* NDH00600
C* NDH00610
C* NDH00620
C* NDH00630
C* NDH00640
C* NDH00650
C* NDH00660
C* NDH00670
C* NDH00680
C* NDH00690
C* NDH00700
C* NDH00710
C* NDH00720
C* NDH00730
C* NDH00740
C* NDH00750
C* NDH00760
C* NDH00770
C* NDH00780
C* NDH00790

SURROUTINE NDHST1(HIST,FIELDS,      MEANS,VERTEX,LIMIT,ARRAY,TOP) NDH00010
NDHST1 SETS UP THE LOGIC FOR HISTOGRAMMING THE DATA AND WRITING
THE NDM FILE
IMPLICIT INTEGER (A-Z)
REAL MEANS
DIMENSION HIST(1),ARRAY(1),MEANS(1)
DIMENSION FIELDS(4,1),FETVEC(30),FL(12),VERTEX(1)
INCLUDE COMMK1.LIST
INCLUDE COMMK6.LIST
INCLUDE COMH11.LIST
COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
AVAP2,COV42,CLSID2,SURN02,SURDS2,FLDSV2,VERTX2,
FETVC2(30),SURVC2(75),SUMPTH(75),CL SVC2(60),
KRPPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),
GRPCHA(61),GROUPS(124)
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,HMKEY,
HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
,NHSTUN,NHSTFI,SCTRUN,MAPFIL
,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
CRDUNT,PRTUNT,RANDIO
COMMON /NDM/NCLRCH,CLRVEC(30),MAXVEC,MAPKEY,
CLASS,SUBCLS,FIELD,MEANSW,NOVEC,FLDINF(6),SIZE,TOTMNS
,CNTR1,CNTR2,ID1,ID2,COLOR1,COLOR2,RUFLEN,ID3,COLOR3,NODUMP
,IDATA1,TOTVEC
CSEND
C INITIALIZE PARAMETERS
IENTER = 0
STAMNT = 1
90 LENTH = NOVEC*SIZE
DO 80 JK=1,LENTH
80 HIST(JK) = 0
REGIN = DRUMAD
NOVFLD = 0
VECCNT = 0
NODUMP = 0
NOVEC = 0
TOTVEC = 0
NOFLD2 = 0
TOTVT2 = 0
IF (MEANSW.NE.1) GO TO 85
DO 82 J=1,TOTMNS
82 MEANS(J) = 0.0
TOTMNS = 0
85 IF (CLASS.EQ.1) CALL FLDCLS(FIELDS,STAMNT,&100,&510,&520,IPT,
,VERTEX)
IF (SUBCLS.EQ.1) CALL FLDSUB(FIELDS,STAMNT,&100,&510,&520,IPT,
,VERTEX)
IF (FIELD.EQ.1) CALL FLDFLD(FIELDS,STAMNT,&100,&530,IPT,VERTEX)
C
100 LINSTR = FLDINF(1)
LINEND = FLDINF(2)
LININC = FLDINF(3)
SAMSTR = FLDINF(4)
SAMEND = FLDINF(5)
SAMINC = FLDINF(6)
FIELDS(2,NOFLD2) = NOCLS2
FIELDS(3,NOFLD2) = NOSUR2
TOTVT2 = FIELDS(4,NOFLD2) + TOTVT2
IF (MEANSW.EQ.1) TOTMNS = NCLRCH + TOTMNS
C
ILINE = (LINEND-LINSTR)/LININC + 1
NSAMP = (SAMEND-SAMSTR)/SAMINC + 1
C
C COMBINE PLOTTING AND COLOR CHANNELS
C
DO 110 I=1,NOFET2
110 FETVEC(I) = FETVC2(I)
IF (NCLRCH.EQ.0) GO TO 130
C
C ARE COLOR CHANNELS AND PLOTTING CHANNELS THE SAME CHANNELS
C
IF (NOFET2.NE.NCLRCH) GO TO 102
DO 101 I=1,NOFET2
101 CONTINUE
IF (FETVEC(I).NE.CLRVEC(I)) GO TO 102
NOFEAT = NOFET2

```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: NDHST1

```

      RGCHAN = 1
      GO TO 135
102 CONTINUE
      DO 120 I=1,NCLRCH
120 FETVEC(NOFET2+I) = CLRVEC(I)
130 NOFEAT = NOFET2 + NCLRCH
C
      RGCHAN = NOFET2 + 1
135 CONTINUE
      COMPUTE ADDRESSES
      CALL ADDR3(TOP,NSAMP,NOFEAT,BEGIN,BEGIN)
      IF REMAINING STORAGE IN HIST ARRAY IS LESS THAN 2000, EMPTY
      HIST ARRAY (DATA VECTORS) ONTO DRUM, READ IN MAP TAPE,STORE ONTO
      DRUM, THEN READ DATA VECTORS BACK INTO HIST
      IF (MAPKEY .NE. 1) GO TO 105
      STORGE = LIMIT - NOVEC*SIZE
      IF (STORGE .LT. 2000) GO TO 103
      CALL STODAT(ILINE,NSAMP,HIST(NOVEC+1),STORGE,BEGIN)
      GO TO 105
103 VECTR1 = COLOR2
      WRDS = NOVEC*SIZE
      CALL RWRITE(VECTR1,HIST,WRDS,ISTAT)
104 IF (ISTAT .EQ. 1) GO TO 104
      REGIN1 = VECTR1 + WRDS
      CALL STODAT(ILINE,NSAMP,HIST,LIMIT,REGIN1)
      CALL RREAD(VECTR1,HIST,WRDS,ISTAT2)
C
      INITIALIZE IMAGE DATA TAPE
105 CALL TAPHDR(DATAPF,DATAFIL)
106 IF (ISTAT2 .EQ. 1) GO TO 106
      POSITION IMAGE TAPE FOR THIS FIELD
      CALL FLDINT(FLDINF(1),FETVEC,NOFEAT)
      NLINE = 0
      NPTS = 0
      READ A SCAN LINE OF DATA AND PROCESS IT
      DO 500 LINE=I.INSTR,LINEND,LININC
      NLINE = NLINE + 1
      CALL LINEDD(ARRAY(IDATA1),FNDTAP)
      IF (FNDTAP .EQ. -1) GO TO 600
      READ IN A SCAN LINE FROM CLASSIFICATION/CLUSTER MAP TAPE
      IF (MAPKEY .EQ. 1) CALL RFSTO(INLINE,NSAMP,BEGIN1)
      FIND INTERSECTIONS FOR N-R FIELDS
      CALL FDLINT(VERTEX(IPT),FIELDS(4,NOFLD2),FL,LINE, SAMP,NI)
      DO 400 J=1,NI,2
      IR = (FL(J)-SAMSTR)/SAMINC + 1
      IE = (FL(J+1)-SAMSTR)/SAMINC + 1
      IF (MOD(SAMSTR,SAMINC) .NE. MOD(FL(J),SAMINC)) IB = IB + 1
      IF (IR .GT. IE) GO TO 400
      DO 350 K=IB,IE
      TOTVEC = TOTVEC + 1
      NPTS = NPTS + 1
      HISTOGRAM VECTOR
      CALL NDHST2(K,ARRAY(IDATA1),HIST,NOFET2,VECSWT,NSAMP,
      * ARRAY(1),VECCNT,OVRFLO,NOFEAT,RGCHAN)
      IF A NEW VECTOR WAS FOUND,VECSWT = 1
220 IF (MEANSW .EQ. 0) GO TO 330
      COMPUTE MEANS FOR TRAINING/TEST FIELDS
      NDH00800
      NDH00810
      NDH00820
      NDH00830
      NDH00840
      NDH00850
      NDH00860
      NDH00870
      NDH00880
      NDH00890
      NDH00900
      NDH00910
      NDH00920
      NDH00930
      NDH00940
      NDH00950
      NDH00960
      NDH00970
      NDH00980
      NDH00990
      NDH01000
      NDH01010
      NDH01020
      NDH01030
      NDH01040
      NDH01050
      NDH01060
      NDH01070
      NDH01080
      NDH01090
      NDH01100
      NDH01110
      NDH01120
      NDH01130
      NDH01140
      NDH01150
      NDH01160
      NDH01170
      NDH01180
      NDH01190
      NDH01200
      NDH01210
      NDH01220
      NDH01230
      NDH01240
      NDH01250
      NDH01260
      NDH01270
      NDH01280
      NDH01290
      NDH01300
      NDH01310
      NDH01320
      NDH01330
      NDH01340
      NDH01350
      NDH01360
      NDH01370
      NDH01380
      NDH01390
      NDH01400
      NDH01410
      NDH01420
      NDH01430
      NDH01440
      NDH01450
      NDH01460
      NDH01470
      NDH01480
      NDH01490
      NDH01500
      NDH01510
      NDH01520
      NDH01530
      NDH01540
      NDH01550
      NDH01560
      NDH01570
      NDH01580
```

FILE: NDHST1

```
C      CALL FLDOPEN(ARRAY(IDATA),K,NSAMP,NOFEAT,MFANS,RCCHAN,NPTS)      NDH01590
C      330 IF (VECSWT .NE. 1) GO TO 350      NDH01600
C      ARRAY(ID1 + VECNT) = NOFLD2      NDH01610
C      IF (MAPKEY .EQ. 0) GO TO 340      NDH01620
C      RETRIEVE CLUSTERED/CLASSIFIED DATA FROM DRUM      NDH01630
C      CALL RSTOR(K,ARRAY(ID1+VECCNT))      NDH01640
C      340 VECCNT = VECCNT + 1      NDH01650
C      IF (VECCNT .LE. (RUFLEN-1)) GO TO 350      NDH01660
C      DUMP ONTO DRUM      NDH01670
C      NODUMP = NODUMP + 1      NDH01680
C      CALL RWRITE(ID2,ARRAY(ID1),RUFLEN,ISTAT1)      NDH01690
C      ID2 = ID2 + RUFLEN      NDH01700
C      VECCNT = 0      NDH01710
C      IF (MEANSW .EQ. 1) GO TO 350      NDH01720
C      IF (INLRCH .EQ. 0) GO TO 350      NDH01730
C      CALL RWRITE(COLOR2,ARRAY(COLOR1),RUFLEN,ISTAT2)      NDH01740
C      COLOR2 = COLOR2 + RUFLEN      NDH01750
C      350 CONTINUE      NDH01760
C      400 CONTINUE      NDH01770
C      500 CONTINUE      NDH01780
C      IF OVRFLD IS GREATER THAN 0, HISTOGRAMMED DATA VECTOR TABLE IS      NDH01790
C      FULL      NDH01800
C      IF (OVRFLD .EQ. 0) GO TO 505      NDH01810
C      WRITE(6,221)      NDH01820
C      221 FORMAT(////)      NDH01830
C      WRITE(6,225)OVRFLD      NDH01840
C      225 FORMAT(1X,16,' VECTORS WERE NOT HISTOGRAMMED, BUT USED IN COMPUTING      NDH01850
C      *G FIELD MEANS, IF APPLICABLE*')      NDH01860
C      505 CONTINUE      NDH01870
C      IF (FIELD .NE. 1) GO TO 65      NDH01880
C      CALL WRTFIL(HIST,MEANS,ARRAY(ID1),ARRAY(COLOR1),FIELDS,VERTEX,      NDH01890
C      * TENTE*)      NDH01900
C      GO TO 90      NDH01910
C      WRITE HISTOGRAM FILE      NDH01920
C      510 CALL WRTFIL(HIST,MEANS,ARRAY(ID1),ARRAY(COLOR1),FIELDS,VERTEX,      NDH01930
C      * TENTE*)      NDH01940
C      GO TO 90      NDH01950
C      %END CARD FOUND      NDH01960
C      520 CALL WRTFIL(HIST,MEANS,ARRAY(ID1),ARRAY(COLOR1),FIELDS,VERTEX,      NDH01970
C      * TENTE*)      NDH01980
C      530 ENDFILE NHSTUN      NDH01990
C      DEWIND NHSTUN      NDH02000
C      RETURN      NDH02010
C      600 WRITE(6,610)      NDH02020
C      610 FORMAT(' ERROR IN FIELD CARD. ABORTING*')      NDH02030
C      CALL CWERH      NDH02040
C      END      NDH02050
C      END      NDH02060
C      END      NDH02070
C      END      NDH02080
C      END      NDH02090
C      END      NDH02100
C      END      NDH02110
C      END      NDH02120
C      END      NDH02130
C      END      NDH02140
C      END      NDH02150
C      END      NDH02160
C      END      NDH02170
C      END      NDH02180
C      END      NDH02190
C      END      NDH02200
C      END      NDH02210
```

ALL PAGE IS  
OF POOR QUALITY



FILE: NDHST2

```
SUBROUTINE NDHST2(J, IDATA, HIST, NOFET2, VEC SWT, NSAMP, ARRAY, VEC CNT, NDH00010
* OVPFLO, NOFEAT, RGCHAN) NDH00020
C NDH00030
C NDH00040
C NDH00050
C NDH00060
C NDH00070
C NDH00080
C NDH00090
C NDH00100
C NDH00110
C NDH00120
C NDH00130
C NDH00140
C NDH00150
C NDH00160
C NDH00170
C NDH00180
C NDH00190
C NDH00200
C NDH00210
C NDH00220
C NDH00230
C NDH00240
C NDH00250
C NDH00260
C NDH00270
C NDH00280
C NDH00290
C NDH00300
C NDH00310
C NDH00320
C NDH00330
C NDH00340
C NDH00350
C NDH00360
C NDH00370
C NDH00380
C NDH00390
C NDH00400
C NDH00410
C NDH00420
C NDH00430
C NDH00440
C NDH00450
C NDH00460
C NDH00470
C NDH00480
C NDH00490
C NDH00500
C NDH00510
C NDH00520
C NDH00530
C NDH00540
C NDH00550
C NDH00560
C NDH00570
C NDH00580
C NDH00590
C NDH00600
C NDH00610
C NDH00620
C NDH00630
C NDH00640
C NDH00650
C NDH00660
C NDH00670
C NDH00680
C NDH00690
C NDH00700
C NDH00710
C NDH00720
C NDH00730
C NDH00740
C NDH00750
C NDH00760
C NDH00770
C NDH00780
C NDH00790
```

NDHST2 PERFORMS THE 1 TO 16 CHANNEL HISTOGRAM  
THE HISTOGRAM IS COMPUTED FOR EITHER ONE OR TWO SETS OF CHANNELS

IMPLICIT INTEGER (A-Z)  
DIMENSION HIST(SIZE, MAXVEC), IDATA(1), COMWRD(4)  
DIMENSION ARRAY(1)  
INCLUDE COMMK6.LIST

INCLUDE COMMK11.LIST  
COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,  
\* HISFIL, HISKEY, TRFORM, EKIP, ERPKEY, MAPUNT, NOFILF,  
\* DRUMAD, DRUMWDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL  
\* NHSTUN, NHSTFI, SCTRUN, MAPFIL  
\* DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,  
\* CRDUNT, PRUNT, RANDIO  
COMMON /NDIM/NCLCH, CLRVEC(30), MAXVEC, MAPKEY,  
\* CLASS, SURCLS, FIELD, MEANS, NOVEC, FLDINF(6), SIZE, TOTMNS  
\* CNTF1, CNTF2, ID1, ID2, COLOR1, COLOR2, BUFLN, ID3, COLOR3, NODUMP  
\* IDATA1, TOTVEC

VECSWT = 0

LOGICAL \*1 LDUM(4), LLDUM(4)  
EQUIVALENCE (IDUM, LDUM(1)), (IIDUM, LLDUM(1))  
DO 50 I=1,4  
50 COMWRD(I)=0  
II=0  
DO 60 I=1, NOFET2  
II=II+1  
III=1+(II-1)/4  
IIII=J+(II-1)\*NSAMP  
IYTE=II-(II-1)/4  
IF (IYTE .EQ. 1) IIDUM=0  
IDUM=IDATA(IIII)  
LLDUM(IYTE)=LDUM(4)  
COMWRD(III)=IIDUM  
50 CONTINUE

STORE LAST 8 BITS OF IDATA(IIII) INTO COMWRD(III) IN  
RYE POSITION IYTE, LEFT TO RIGHT  
IF (NCLCH .EQ. 0) GO TO 100  
IF (MEANS .EQ. 1) GO TO 100  
CALL PICOLR(IDATA, J, NOFEAT, COLWRD, NSAMP, NOFET2, RGCHAN)  
100 CONTINUE  
IF (NOVEC .EQ. 0) GO TO 135

COMPARING VECTORS ALREADY FOUND WITH IN COMING VECTOR

110 DO 130 K=1, NOVEC  
DO 120 L=1, SIZE  
IF (COMWRD(L) .NE. HIST(L, K)) GO TO 130  
120 CONTINUE

FOUND A MATCHING VECTOR

DO ONE OF THE FOLLOWING :  
1) HISTOGRAM THE VECTOR ONLY IF COLOR CODES FOR BOTH VECTORS ARE  
THE SAME.  
2. DO NOT CHECK COLOR CODES IF MAP TAPE IS BEING INPUT OR MEANS  
FOR FIELDS ARE BEING COMPUTED

IF (NCLCH .EQ. 0) GO TO 126  
IF (MEANS .EQ. 1) GO TO 126

IS COLOR CODE IN CORE OR ON DRUM

IF (NODUMP .EQ. 0) GO TO 122  
IF (K .GT. (NODUMP\*BUFLN)) GO TO 122  
ADDRS = COLOR3 + (NODUMP-1)\*BUFLN + K - 1  
CALL RPFAD(ADDRS, COFS, 1, ISTAT2)  
121 IF (ISTAT2 .EQ. 1) GO TO 121  
GO TO 124  
122 KK = K - (NODUMP\*BUFLN)  
COFS = ARRAY(COLOR1 + KK - 1)  
124 IF (COFS .NE. COLWRD) GO TO 150

FILE: NDHST2

```
C 126 CONTINUE
      CNTN1 = CNTN2 + K - 1
      CALL PREAD(CNTN1,COUNTR,1,ISTAT)
125 IF (ISTAT .EQ. 1) GO TO 125
      COUNTN = COUNTN + 1
      CALL PWRITE(CNTN1,COUNTR,1,ISTAT1)
127 IF (ISTAT1 .EQ. 1) GO TO 127
      GO TO 150
130 CONTINUE
C C C
      INSERT NEW VECTOR
135 IF (NOVEC .LT. MAXVEC) GO TO 137
      OVPFLO = OVPFLO + 1
      IF (OVPFLO .GT. 1) GO TO 136
133 FORMAT(//)
      WRITE(6,133)
123 FORMAT(// 'CORE LIMITS EXCEEDED. MAXIMUM NO. OF VECTORS// ACCEPTED
      * IS ',I6)
C C C
      VECTOR TABLE IS FULL--CONTINUE TO HISTOGRAM DATA VECTORS THAT
      ALREADY EXIST
136 RETURN
137 NOVEC = NOVEC + 1
      DO 140 L=1,SIZE
140 HIST(L,NOVEC) = COMWRD(L)
      CNTN1 = CNTN2 + NOVEC - 1
      COUNTN = 1
      CALL PWRITE(CNTN1,COUNTR,1,ISTAT3)
145 IF (ISTAT3 .EQ. 1) GO TO 145
      VECST = 1
      IF (MFANSW .NE. 0 .OR. MAPKEY .NE. 0) GO TO 150
      ARRAY(COLOR1 + VECST) = COLWRD
150 CONTINUE
200 CONTINUE
C
      RETURN
      END
```

NDH00800  
NDH00810  
NDH00820  
NDH00830  
NDH00840  
NDH00850  
NDH00860  
NDH00870  
NDH00880  
NDH00890  
NDH00900  
NDH00910  
NDH00920  
NDH00930  
NDH00940  
NDH00950  
NDH00960  
NDH00970  
NDH00980  
NDH00990  
NDH01000  
NDH01010  
NDH01020  
NDH01030  
NDH01040  
NDH01050  
NDH01060  
NDH01070  
NDH01080  
NDH01090  
NDH01100  
NDH01110  
NDH01120  
NDH01130  
NDH01140  
NDH01150  
NDH01160  
NDH01170  
NDH01180  
NDH01190  
NDH01200

FILE: PICOLR

```
C* SUBROUTINE PICOLR(IDATA,K,NOFEAT,COLWRD,NSAMP,NOFET2,RGCHAN) PIC00010
C* PICOLR EXTRACT THE COLOR CHANNELS FROM IDATA AND PIC00020
C* PACK THE 8 BIT PIXELS INTO A COMPUTER WORD PIC00030
C* IMPLICIT INTEGER (A-Z) PIC00040
C INCLUDE CMH011.LIST PIC00050
COMMON /NDIM/NCLRCH,CLRVEC(30),MAXVEC,MAPKEY, PIC00060
* CLASS,SUBCLS,FIELD,MFANSW,NOVEC,FLDINF(6),SIZE,TOTMNS PIC00070
* ,CNT01,CNTR2,ID1,ID2,COLOR1,COLOR2,HUFLEN,ID3,COLOR3,NODIMP PIC00080
* ,IDATA1,TOTVEC . PIC00090
CSEND PIC00100
DIMENSION IDATA(1) PIC00110
LOGICAL*1 LDUM(4),LLDUM(4) PIC00120
EQUIVALENCE (IDUM,LDUM(1)),(IIDUM,LLDUM(1)) PIC00130
IIDUM=0 PIC00140
DO 100 I=1,NCLRCH PIC00150
II=K+(RGCHAN+I-2)*NSAMP PIC00160
IDUM=IDATA(II) PIC00170
100 LLDUM(I)=LDUM(4) PIC00180
COLWRD=IIDUM PIC00190
RETURN PIC00200
C END PIC00210
PIC00220
PIC00230
```





FILE: SET10

C CHANNEL CARD

```
150 M = NKTCHR(CARD, COL)
    IF (M.EQ. BLANK) GO TO 105
    IF (M.EQ. PRCN) GO TO 160
    IF (M.EQ. C4CD) GO TO 170
153 WRITE(A, 155)
155 FORMAT(' ERROR ON CHANNELS CARD')
    GO TO 105
160 J = FIND12(CARD, COL, EQUOM)
    IF (J.NE. 2) GO TO 153
    NOFET2 = NUMBER(CARD, COL, FETVC2, NOFET2)
    COL = COL - 1
    CALL ORDER(FETVC2, NOFET2)
    GO TO 150
170 J = FIND12(CARD, COL, EQUOM)
    IF (J.NE. 2) GO TO 153
    NCLRCH = NUMBER(CARD, COL, CLRVEC, NCLRCH)
    COL = COL - 1
    CALL ORDER(CLRVEC, NCLRCH)
    GO TO 150
```

C DATA FILE CARD

```
180 M = NKTCHR(CARD, COL)
    IF (M.EQ. BLANK) GO TO 105
    IF (M.EQ. UNCD) GO TO 190
    IF (M.EQ. FPCD) GO TO 200
185 WRITE(A, 187)
187 FORMAT(' ERROR ON DATA FILE CARD')
    GO TO 105
190 J = FIND12(CARD, COL, EQUOM)
    IF (J.NE. 2) GO TO 185
    M = NUMBER(CARD, COL, DATAPE, ZERO)
    COL = COL - 1
    GO TO 180
200 J = FIND12(CARD, COL, EQUOM)
    IF (J.NE. 2) GO TO 185
    M = NUMBER(CARD, COL, DATFIL, ZERO)
    DATFIL = DATFIL - 1
    COL = COL - 1
    GO TO 180
```

C CLUSTER/CLASSIFICATION MAP TAPE

```
210 M = NKTCHR(CARD, COL)
    IF (M.EQ. BLANK) GO TO 105
    IF (M.EQ. UNCD) GO TO 230
    IF (M.EQ. FPCD) GO TO 240
215 WRITE(A, 220)
220 FORMAT(' ERROR ON UAS FILE CARD')
    GO TO 105
230 J = FIND12(CARD, COL, EQUOM)
    IF (J.NE. 2) GO TO 215
    M = NUMBER(CARD, COL, MAPUNT, ZERO)
    MAPKEY = 1
    COL = COL - 1
    GO TO 210
240 J = FIND12(CARD, COL, EQUOM)
    IF (J.NE. 2) GO TO 215
    M = NUMBER(CARD, COL, MAPFIL, ZERO)
    MAPFIL = MAPFIL - 1
    COL = COL - 1
    GO TO 210
```

C N-DIM HISTOGRAM FILE

```
250 M = NKTCHR(CARD, COL)
    IF (M.EQ. BLANK) GO TO 105
    IF (M.EQ. UNCD) GO TO 270
    IF (M.EQ. FPCD) GO TO 280
260 WRITE(A, 265)
265 FORMAT(' ERROR ON N-DIM HISTOGRAM FILE CARD')
    GO TO 105
270 J = FIND12(CARD, COL, EQUOM)
    IF (J.NE. 2) GO TO 260
    M = NUMBER(CARD, COL, NHSTUN, ZERO)
    COL = COL - 1
    GO TO 250
```

T00800  
T00810  
T00820  
T00830  
T00840  
T00850  
T00860  
T00870  
T00880  
T00890  
T00900  
T00910  
T00920  
T00930  
T00940  
T00950  
T00960  
T00970  
T00980  
T00990  
T01000  
T01010  
T01020  
T01030  
T01040  
T01050  
T01060  
T01070  
T01080  
T01090  
T01100  
T01110  
T01120  
T01130  
T01140  
T01150  
T01160  
T01170  
T01180  
T01190  
T01200  
T01210  
T01220  
T01230  
T01240  
T01250  
T01260  
T01270  
T01280  
T01290  
T01300  
T01310  
T01320  
T01330  
T01340  
T01350  
T01360  
T01370  
T01380  
T01390  
T01400  
T01410  
T01420  
T01430  
T01440  
T01450  
T01460  
T01470  
T01480  
T01490  
T01500  
T01510  
T01520  
T01530  
T01540  
T01550  
T01560  
T01570  
T01580

ORIGINAL PAGE IS  
OF POOR QUALITY



FILE: SET10

```
1010 FORMAT(' HISTOGRAM DATA VECTORS FROM CHANNELS ',16(I2.1X))
1020 FORMAT(' COLOR CODES ARE FROM CHANNELS ',4(I2.1X))
1030 FORMAT(' CLUSTER/CLASSIFICATION TAPE IS BEING INPUT')
1040 FORMAT(' HISTOGRAM FIELDS BY CLASS')
1050 FORMAT(' HISTOGRAM FIELDS BY SUBCLASS')
1060 FORMAT(' HISTOGRAM FIELDS ON PEX FIELDS BASES')
1070 FORMAT(' COMPUTE MEANS OF INPUT FIELDS')
      RETURN
```

C  
END

```
SF T02380
SUM T02390
SUM T02400
SUM T02410
SUM T02420
SUM T02430
SUM T02440
SUM T02450
SUM T02460
SUM T02470
```

ORIGINAL PAGE 1  
OF POOR QUALITY

~~15-17~~

288



FILE: STODAT

```

SURROUTINE STODAT(ILINE,NSAMP,HIST,LIMIT,BEGIN)
C*
C* STODAT READS AND STORES THE CLASSIFICATION/CLUSTER MAP ON DRUM
C*
C IMPLICIT INTEGER (A-Z)
C INCLUDE COMRAK.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILF,
* DRUMAD,DRMWD,DRMWD,DRMWD,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* .NHSTUN,NHSTFI,SCTRUN,MAPFIL
* .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
CSEND
DIMENSION HIST(LIMIT),FETVEC(1),FLD(6),NLINE(4)
TOTWRD = ILINE*NSAMP
IF (TOTWRD .LE. (DRMWD-(DRUMAD-BEGIN))) GO TO 120
WRITE(6,110)
110 FORMAT(' NOT ENOUGH DRUM SPACE TO STORE DAS TAPE DATA',
CALL CMERR
C
120 CALL TAPHDR(MAPUNT,MAPFIL)
FETVEC(1) = 1
NOFEAT = 1
FLD(1) = 1
FLD(2) = ILINE
FLD(3) = 1
FLD(4) = 1
FLD(5) = NSAMP
FLD(6) = 1
REGIN = BEGIN
CALL FLDINT(FLD ,FETVEC,NOFEAT)
DUMPS = TOTWRD / LIMIT
IF (MOD(TOTWRD,LIMIT) .NE. 0) DUMPS = DUMPS + 1
TOTLNS = LIMIT / NSAMP
IF (TOTLNS .GE. ILINE) GO TO 140
DMP = DUMPS - 1
DO 130 I=1,DMP
130 NLINE(I) = TOTLNS
NLINE(DUMPS) = ILINE - TOTLNS*DMP
GO TO 150
140 NLINE(1) = ILINE
C
150 DO 200 J=1,DUMPS
NUMLIN = NLINE(J)
DO 160 K=1,NUMLIN
WORDS = NSAMP*(K-1)
160 CALL LINERD(HIST(WORDS+1),ENDTAP)
C
C STORE ON HIGH SPEED DRUM
C
NWORDS = WORDS + NSAMP
CALL RWRITE(REGIN,HIST(1),NWORDS,ISTAT)
200 REGIN = BEGIN + NLINE(J) * NSAMP
C
MAPFIL = MAPFIL + 1
C
RETURN
END

```

ST000010  
ST000020  
ST000030  
ST000040  
ST000050  
ST000060  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00060  
ST000080  
ST000090  
ST000100  
ST000110  
ST000120  
ST000130  
ST000140  
ST000150  
ST000160  
ST000170  
ST000180  
ST000190  
ST000200  
ST000210  
ST000220  
ST000230  
ST000240  
ST000250  
ST000260  
ST000270  
ST000280  
ST000290  
ST000300  
ST000310  
ST000320  
ST000330  
ST000340  
ST000350  
ST000360  
ST000370  
ST000380  
ST000390  
ST000400  
ST000410  
ST000420  
ST000430  
ST000440  
ST000450  
ST000460  
ST000470  
ST000480  
ST000490  
ST000500  
ST000510  
ST000520  
ST000530

FILE: WRTFIL

```

SUBROUTINE WRTFIL(HIST,MEANS,ID,COLOR,FIELDS,VERTEX,I)
WRT00010
WRT00020
WRT00030
WRT00040
WRT00050
WRT00060
WRT00070
WRT00080
WRT00090
WRT00100
WRT00110
WRT00120
WRT00130
WRT00140
WRT00150
WRT00160
WRT00170
WRT00180
WRT00190
WRT00200
WRT00210
WRT00220
WRT00230
WRT00240
WRT00250
WRT00260
WRT00270
WRT00280
WRT00290
WRT00300
WRT00310
WRT00320
WRT00330
WRT00340
WRT00350
WRT00360
WRT00370
WRT00380
WRT00390
WRT00400
WRT00410
WRT00420
WRT00430
WRT00440
WRT00450
WRT00460
WRT00470
WRT00480
WRT00490
WRT00500
WRT00510
WRT00520
WRT00530
WRT00540
WRT00550
WRT00560
WRT00570
WRT00580
WRT00590
WRT00600
WRT00610
WRT00620
WRT00630
WRT00640
WRT00650
WRT00660
WRT00670
WRT00680
WRT00690
WRT00700
WRT00710
WRT00720
WRT00730
WRT00740
WRT00750
WRT00760
WRT00770
WRT00780
WRT00790

SUBROUTINE WRTFIL(HIST,MEANS,ID,COLOR,FIELDS,VERTEX,I)
WRTFIL WRITES THE NDIM FILE
IMPLICIT INTEGER (A-Z)
INCLUDE COMAK1.LIST
INCLUDE COMAK6.LIST
INCLUDE COMAK11.LIST
COMMON/INFORM/NOCS2,NOSUR2,NOFET2,VAR5Z2,TOTVT2,NOFLD2,
* AVAP2,CVAR2,CLSID2,SURNO2,SURDS2,FLDSV2,VERTX2,
* FETVC2(30),SUBVC2(75),SURPTR(75),CL SVC2(60),
* KEPPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,RMKEY,
* HISFIL,HISKEY,THFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRM4DS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* NHSTUN,NHSTFI,SCTRUN,MAPFIL
* DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTWFL,HISTFL,PCHUNT,
* CRDUNT,PRTUNT,RANDIO
COMMON/NDIM/NCLRCH,CLRVEC(30),MAXVEC,MAPKEY,
* CLASS,SUBCLS,FIELD,MEANS,NOVEC,FLDINF(6),SIZE,TOTMNS
* CNTR1,CNTR2,ID1,ID2,COLOR1,COLOR2,RUFLEN,ID3,COLOR3,NODUMP
* IDATA1,TOTVEC
CSEND
DIMENSION HIST(1),MEANS(1),ISTAT(4),ID(1),COLOR(1),VERTEX(2,1)
DIMENSION FIELDS(4,1)
REAL MFANS
DATA BLANK/' '/
I = 1
IF (I .GT. 1) GO TO 100
WRITE HEADER RECORD
WRITE (NHSTUN) TOTMNS,SIZE,NOFET2,(FETVC2(I),I=1,NOFET2),
* NCLRCH,(CLRVEC(I),I=1,NCLRCH)
ENDFILE NHSTUN
WRITE RECORD 1
100 WRITE (NHSTUN) NOFLD2,NOSUR2,TOTVT2,NOVEC
WRITE RECORD 2
WRITE (NHSTUN) CLSVC2(1),(SUBVC2(I),I=1,NOSUR2),((FIELDS(I,J),
* I=1,4),J=1,NOFLD2),((VERTEX(I,J),I=1,2),J=1,TOTVT2)
WRITE RECORD 3
IF (TOTMNS .GT. 0) WRITE (NHSTUN) (MEANS(I),I=1,TOTMNS)
WRITE RECORD 4
II = SIZE*NOVEC
WRITE (NHSTUN) (HIST(I),I=1,II)
IF (NODUMP .EQ. 0) GO TO 160
READ ID INFORMATION INTO CORE
WRITE RECORD 5
VECDRM = NODUMP*RUFLEN
VECARY = NOVEC - VECDRM
CALL RREAD(ID3,HIST(1),VECDRM,ISTAT(1))
IF (VECARY .EQ. 0) GO TO 115
DO 110 I=1,VECARY
110 HIST(VECDRM+I) = ID(I)
115 CONTINUE
120 IF (ISTAT(1) .EQ. 1) GO TO 120
WRITE (NHSTUN) (HIST(I),I=1,NOVEC)
WRITE REC 6
CALL RREAD(CNTR2,HIST(1),NOVEC,ISTAT(3))
125 IF (ISTAT(3) .EQ. 1) GO TO 125
WRITE (NHSTUN) (HIST(I),I=1,NOVEC)
WRITE REC 7
```

FILE: WRTFIL

```
IF (TOTMNS .NE. 0) GO TO 180
IF (NCLRCH .EQ. 0) GO TO 180
CALL BREAD(COLOR3,HIST(1),VECDRM,ISTAT(2))
IF (VECARV .EQ. 0) GO TO 135
DO 130 I=1,VECARV
130 HIST(VECDRM+I) = COLOR(I)
135 IF (ISTAT(2) .EQ. 1) GO TO 135
140 WRITE(NHSTUN)(HIST(I),I=1,NOVEC)
GO TO 180

INFORMATION DID NOT NEED TO BE STORED ON DRUM

RECORD 5
160 WRITE(NHSTUN)(ID(I),I=1,NOVEC)
RECORD 6

HISTOGRAM INFORMATION IS ALWAYS STORED ON HIGH SPEED FRUM
165 CALL BREAD(CNTR2,HIST(1),NOVEC,ISTAT(3))
170 IF (ISTAT(3) .EQ. 1) GO TO 170
WRITE(NHSTUN)(HIST(I),I=1,NOVEC)

WRITE REC 7

IF (TOTMNS .NE. 0) GO TO 180
IF (NCLRCH .GT. 0) WRITE(NHSTUN)(COLOR(I),I=1,NOVEC)
180 ENDFILE NHSTUN

WRITE FILE INFO ON LINE PRINTER
CALL WRTFLD(FIELDS,VERTEX,NOFLD2,2,CLSV2,SURVC2)
WRITE(6,190)TOTVEC,NOVEC
190 FORMAT(///55X,'TOTAL NO. OF VECTORS =' ,I6/48X,'TOTAL NO. OF UNIONS' ,I6)
WRITE(6,200) (BLANK,CLRVEC(I),I=1,NCLRCH)
200 FORMAT(///T60,'FIELD MEANS'//T44,4(A1,'CH(',I2,')',5X))

WRITE(6,210)(MFANS(I),I=1,TOTMNS)
210 FORMAT(T44,4(F7.2,5X))

RETURN
END
```

WRT00810  
WRT00810  
WRT00820  
WRT00830  
WRT00840  
WRT00850  
WRT00860  
WRT00870  
WRT00880  
WRT00890  
WRT00900  
WRT00910  
WRT00920  
WRT00930  
WRT00940  
WRT00950  
WRT00960  
WRT00970  
WRT00980  
WRT00990  
WRT01000  
WRT01010  
WRT01020  
WRT01030  
WRT01040  
WRT01050  
WRT01060  
WRT01070  
WRT01080  
WRT01090  
WRT01100  
WRT01110  
WRT01120  
WRT01130  
WRT01140  
WRT01150  
WRT01160  
WRT01170  
WRT01180  
WRT01190  
WRT01200  
WRT01210  
WRT01220  
WRT01230  
WRT01240  
WRT01250  
WRT01260  
WRT01270  
WRT01280  
WRT01290  
WRT01300

## 16. SCTRPL PROCESSOR

FILE: SCTRPL

```

SUBROUTINE SCTRPL (ARRAY, TOP)
IMPLICIT INTEGER (A-Z)
SCTRPL IS THE DRIVER FOR THE SCATTER PLOT PROCESSOR
C*
C*
C*
DATA LIMIT/12000/
C INCLUDE COMMK1.LIST
C INCLUDE COMMK12.LIST
COMMON/INFORM/NOCLS2, NOSUB2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
* AVAR2, COVAR2, CLSID2, SUPNO2, SURDS2, FLOSV2, VERTX2,
* FETVC2(30), SUMVC2(75), SURPTH(75), CLSVC2(60),
* KEFPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),
* GRPCHK(61), GROUPS(124)
COMMON/SCATTER/RSCALE, XYSCLF, CLRVEC(30), NCLPCH, CLRKEY, LOG,
* FREQ, XMAX, YMAX, XMIN, YMIN, HCKGND, XHI, XLO, YLO, XSIZ,
* YHI, YSIZ, NPINS, SYMTX(32), RMATX(60), BVEC(30), NRVCHN, NOFEAT
* SCALKY, MENADR, FLDADR, PNTADR, IDADR, NC, RMFEAT, BMCOMB
* NOVEC, TOTMNS, SIZE, DRMID, DRMID1, DRMCLK, DRMCRI, DRMTNS, DRMTN1,
* DRMCNT, DRMCT1, DRMVEC, DRMVC1, VECTR1, DATA1, NVEC, NOREAD, LREAD
* DRMPTR, DRMPT1, FETVEC(16), DRMPLT, CSCALE
* NOSUB
CSEND DIMENSION ARRAY(1), BUFF(12000)
C CALL SFT11 (ARRAY(1), ARRAY(1), BUFF(1))
C
C COMPUTE ADDRESSES
C
10 CALL SETADR (&20, &30, TOP, BUFF, LIMIT)
C*
C* SCATR IS THE MAIN DRIVER FOR CREATING THE SPECTRAL PLOTS
C*
20 CALL SCATR (ARRAY (FLDADR), ARRAY (VERTX2), ARRAY (VECTR1), ARRAY (MENADR
*) ,
* BUFF (1), BUFF (1), ARRAY (DATA1), TOP, LIMIT, BUFF (1))
C
C PROCESS ANOTHER FILE
C
C GO TO 10
C
C SEND* CARD
C
30 READ (21, 100) CARD
100 FORMAT (A4)
C
RETURN
END

```

SCT00010  
SCT00020  
SCT00030  
SCT00040  
SCT00050  
SCT00060  
SCT00070  
SCT00080  
SCT00090  
SCT00100  
SCT00110  
SCT00120  
SCT00130  
SCT00140  
SCT00150  
SCT00160  
SCT00170  
SCT00180  
SCT00190  
SCT00200  
SCT00210  
SCT00220  
SCT00230  
SCT00240  
SCT00250  
SCT00260  
SCT00270  
SCT00280  
SCT00290  
SCT00300  
SCT00310  
SCT00320  
SCT00330  
SCT00340  
SCT00350  
SCT00360  
SCT00370  
SCT00380  
SCT00390  
SCT00400  
SCT00410  
SCT00420  
SCT00430  
SCT00440  
SCT00450  
SCT00460  
SCT00470

FILE: CLRCOD

```
C      SUBROUTINE CLRCOD(IB,MEANS,IDATA,IPOSTN,II)
C      IMPLICIT INTEGER (A-Z)
C      REAL MEANS(1)
C
C      INCLUDE COMRK1.LIST
C      INCLUDE COMRK2.LIST
C      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
*      AVAR2,COVAR2,CLSID2,SUBNO2,SURDS2,FLDSV2,VERTX2,
*      FETVC2(30),SUBVC2(75),SUPTR(75),CLVC2(60),
*      KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
*      GRPCHK(61),GROUPS(124)
C      COMMON/SCTTFR/RSCALF,XYSCLF,CLRVEC(30),NCLRCH,CLRKEY,LOG,
*      FREQ,XMAX,YMAX,XMIN,YMIN,RCKGND,XHI,XLO,YLO,XSIZ,
*      YHI,YSIZ,NHINS,SYMTX(32),BMATRX(60),BVEC(30),NBVCHN,NOFEAT
*      ,SCALKY,MENADR,FLDADR,PNTADR,IDADR,NC,BMFEAT,BMCOMB
*      ,NOVEC,TOTMNS,SIZE,DRMID,DRMID1,DRMCLR,DRMCR1,DRMTNS,DRMTN1,
*      DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD
*      ,DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE
*      ,NOSUH
CSEND
C      DIMENSION IDATA(1)
C      LOGICAL*1 LDUM(4),LLDUM(4)
C      EQUIVALENCE (IDUM,LDUM(1)),(IIDUM,LLDUM(1))
C
C      IF (CLRKEY .NE. 3) GO TO 50
C      COLOR CODES (RADIANCE VALUES) ARE COMING FROM N-DIM HIST FILE
C      COLADR = DRMCLR + NVEC*(II-1) + IB - 1
C      CALL RREAD(COLADR,CODE,1,ISTAT1)
C 105 IF (ISTAT1 .EQ. 1) GO TO 105
C
C      IIDUM=0
C      IIDUM=CODE
C      DO 10 I=1,NC
C      III=XSIZ*(I-1)+IPOSTN
C      LLDUM(4)=LDUM(I)
C 10 IDATA(III)=IIDUM
C
C      RETURN
C
C      COLOR CODES (STAT MEANS OR USER INPUT) ARE STORED IN CORE UNPACKED
C
C 50 IDADR = DRMID + NVEC*(II-1) + IB - 1
C      CALL RREAD(IDADR,IDNUM,1,ISTAT1)
C 55 IF (ISTAT1 .EQ. 1) GO TO 55
C      DO 80 I=1,NC
C      III = IPOSTN + XSIZ*(I-1)
C      JJ = (IDNUM-1)*NC + I
C 80 IDATA(III) = MEANS(JJ) + 0.5
C      RETURN
C      END
```

CLR00010  
CLR00020  
CLR00030  
CLR00040  
CLR00050  
CLR00060  
CLR00070  
CLR00080  
CLR00090  
CLR00100  
CLR00110  
CLR00120  
CLR00130  
CLR00140  
CLR00150  
CLR00160  
CLR00170  
CLR00180  
CLR00190  
CLR00200  
CLR00210  
CLR00220  
CLR00230  
CLR00240  
CLR00250  
CLR00260  
CLR00270  
CLR00280  
CLR00290  
CLR00300  
CLR00310  
CLR00320  
CLR00330  
CLR00340  
CLR00350  
CLR00360  
CLR00370  
CLR00380  
CLR00390  
CLR00400  
CLR00410  
CLR00420  
CLR00430  
CLR00440  
CLR00450  
CLR00460  
CLR00470  
CLR00480  
CLR00490  
CLR00500  
CLR00510  
CLR00520  
CLR00530  
CLR00540

ORIGINAL PAGE IS  
OF POOR QUALITY

16-2  
293



FILE: CNTER

```
C SUBROUTINE CNTER(IB, IDATA, IPOSTN, II, COUNTR) CNT00010
C CNT00020
C CNT00030
C CNT00040
C CNT00050
C CNT00060
C CNT00070
C CNT00080
  INCLUDE CMH12.LIST
  COMMON/SCITFF/RSCALE, XYSCL, CLRVEC(30), NCLRCH, CLRKEY, LOG,
  * FREQ, XMAX, YMAX, XMIN, YMIN, HCKGND, XHI, XLO, YLO, XSIZ,
  * YHI, YSIZ, NRINS, SYMTX(32), RMATRX(60), BVEC(30), NBVCHN, NOFEAT
  * SCALKY, MENADR, FLOADR, PNTADR, IDADR, NC, BMFEAT, HMCOMB
  * NOVEC, TOTMNS, SIZE, DKMID, DRMID1, DRMCLR, DRMCR1, DRMTNS, DRMTN1,
  * DRMCNT, DRMCT1, DRMVEC, DRMVC1, VECTR1, DATA1, NVEC, NOREAD, LREAD
  * DRMPTR, DRMPT1, FETVEC(16), DRMPLT, CSCALE
  * NOSUB
CSEND CNT00100
C DIMENSION IDATA(1) CNT00110
C COMPUTE DRUM ADDRESSES CNT00120
C CNT00130
C CTRADR = NVEC * (II-1) + IR + DRMCNT - 1 CNT00140
C CALL PREAD(CTRADR, COUNTR, 1, ISTAT1) CNT00150
105 IF (ISTAT1 .EQ. 1) GO TO 105 CNT00160
C THE VECTOR COUNTER IS THE LAST CHANNEL CNT00170
C CNT00180
C I = XSIZ * NC + IPOSTN CNT00190
C IF (COUNTR .GT. 255) COUNTR = 255 CNT00200
C IDATA (I) = COUNTR CNT00210
C RETURN CNT00220
C END CNT00230
CNT00240
CNT00250
CNT00260
CNT00270
```

FILE: LINPLT

```

SURROUTINE LINPLT
LINPLT CREATES THE PIXEL FREQUENCY PLOT ON DRUM AND LATER PRINTS
THE IMAGE ON THE LINE PRINTER

THERE ARE 3 ENTRIES :
1. SURROUTINE LINPLT - COMPUTES THE SCALES
2. ENTRY STOPTS      - COMPUTES THE POSITION OF THE PIXEL ON THE
                      PLOT (DRUM ADDRESS)
                      *** MUST BE CALLED FOR EVERY PIXEL ***
3. ENTRY PRTPLOT    - PRINTS THE PLOT ON THE LINE PRINTER
                      *** PIXEL MUST BE POSITIVE ***

IMPLICIT INTEGER (A-Z)
REAL XSCALE,XSHFT,YSCALE,YSHFT,SCALEY,SCALEX,SHFTY,SHFTX
REAL SUM,COUNT,LOG2

DIMENSION YAXIS(11),XAXIS(11)

INCLUDE COMRKA.LIST
INCLUDE COMRKA.LIST
INCLUDE COMRKA.LIST
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
             (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
COMMON/GLORAL/HEAD(63),MAPTAP,DATEP,SAVTAP,BMFILE,BMKEY,
           HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
           DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
           NHSTUN,NHSTFI,SCTHUN,MAPFIL
           DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
           CRDUNT,PRTUNT,RANDIO
COMMON/SCTTR/ASCALE,XYSCLF,CLHVEC(30),NCLRCH,CLRKEY,LOG,
           FREQ,XMAX,YMAX,XMIN,YMIN,BCKGNU,XHI,XLO,YLO,XSIZ,
           YHI,YSIZ,NRINS,SYMTEX(32),HMATRX(60),BVEC(30),NBVCHN,NOFEAT
           SCALKY,MENADH,FLUADH,PNTADH,IDADR,NC,HMFEAT,RMCOMR
           NVEC,TOT,NS,SIZF,DRMID,DRMID1,DRMCLR,DRMCR1,DRMTNS,DRMTN1,
           DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOHEAD,LREAD
           DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE
           NOSUB

CSEND
DATA BLANK/0 0/
LOG2 = ALOG10(2.0)
MAXSUM = 1
IF (XYSCLF .EQ. 0) GO TO 70

DATA IS RESCALED TO 101 BINS

XSIZE = XSIZ
YSIZE = YSIZ
IF (XSIZEF .GT. 101) XSIZE = 101
IF (YSIZEF .GT. 101) YSIZE = 101

RANGES FOR THE X-AXIS
NSIZ = XSIZE / 10 + 1
XSCALE = FLOAT(XLO-XHI)/(XSIZE - 1)
XSHFT = FLOAT(XSIZE*XHI-XLO)/(XSIZE-1)
DO 50 I=1,NSIZ
  XAXIS(NSIZ-I+1) = (10*I-9)*XSCALE*XSHFT + .501
CONTINUE
SCALEY = FLOAT(1-YSIZE)/YHI-YLO
SHFTY = -YHI*SCALEY + 1.0
SCALEX = FLOAT(1-XSIZE)/XHI-XLO
SHFTX = -XHI*SCALEX + 1.0

RANGES FOR THE Y-AXIS
ISIZ = YSIZF / 10 + 1
YSCALE = FLOAT(YLO-YHI)/(YSIZE-1)
YSHFT = FLOAT(YSIZE*YHI-YLO)/(YSIZE-1)
DO 60 I=1,ISIZ
  YAXIS(ISIZ-I+1) = (10*I-9)*YSCALE + YSHFT + .501
CONTINUE
RETURN

DATA IS NOT RESCALED
```



FILE: LINPLT

```

70  DSIZ = 11
    ISIZ = 11
    XSIZ = 101
    YSIZ = 101
    NO 75 = 11 * DSIZ
    YAXIS(1) = YLO * (1-1) * 10
75  XAXIS(1) = XLO * (1-1) * 10
    RETURN
    . . . . .
    ENTRY STOPSTICOUNTN(LINE,SAMPLE)
    REAL SAMPLE,LINE
    COMPUTE POSITION ON GRAPH FOR (SAMPLE,LINE)
    IF (XYSCLC .EQ. 1) GO TO 110
    YPOINT = YSIZ - (LINE - YLO)
    XPOINT = SAMPLE - XLO
    IF (YPOINT .GT. 0) GO TO 80
    YPT = YPT + 1
    RETURN
80  IF (YPOINT .LE. 101) GO TO 90
    YPT = YPT + 1
    RETURN
90  IF (XPOINT .GE. 0) GO TO 100
    XPT = XPT + 1
    RETURN
100 IF (XPOINT .LE. 100) GO TO 115
    XPT = XPT + 1
    RETURN
110 CONTINUE
    YPOINT = LINE * SCALEY + SHFTY + .501
    XPOINT = XSIZ - (SAMPLE * SCALEX + SHFTX + .501) + 1
    IF (XPOINT .LT. 0) XPOINT = 0
    IF (XPOINT .GE. XSIZ) XPOINT = XSIZ - 1
    IF (YPOINT .LT. 1) YPOINT = 1
    IF (YPOINT .GT. YSIZ) YPOINT = YSIZ
115 XYADR = (YPOINT - 1) * XSIZ + XPOINT + DRMPLT
    LOG X BASE 2 OF THE FREQUENCY
    IF (LOG .NE. 1) GO TO 117
    CALL RREAD(XYADR,SUM,1,1,ISTAT1)
122 IF (ISTAT1 .EQ. 1) GO TO 122
    COUNT = COUNT
    SUM1 = SUM1 + ALOG10(COUNT) / LOG2
    CALL RWRITE(XYADR,SUM,1,1,ISTAT2)
121 IF (ISTAT2 .EQ. 1) GO TO 121
    SUM = SUM1 + 0.5
    GO TO 125
117 CONTINUE
    CALL RREAD(XYADR,SUM,1,1,ISTAT1)
120 IF (ISTAT1 .EQ. 1) GO TO 120
    TOTAL NO. OF OCCURRENCES FOR ALL THE DATA VECTORS ASSIGNED
    TO THIS (SAMPLE,LINE) POSITION
    SUM = SUM + COUNT
    CALL RWRITE(XYADR,SUM,1,1,ISTAT2)
123 IF (ISTAT2 .EQ. 1) GO TO 123
    SAVE THE LARGEST NO. OF OCCURRENCES
125 CONTINUE
    IF (SUM .GT. MAXSUM) MAXSUM = SUM
    RETURN
    . . . . .
    ENTRY PRTPLT(PNTR,PNTRS)
    REAL PNTRS(1)
    DIMENSION HINS(16),PNTR(1),PNTR1(101)
    IF (LOG .NE. 1) GO TO 129
    CALL RREAD(DRMPLT,PNTRS,1,201,ISTAT3)
126 IF (ISTAT3 .EQ. 1) GO TO 126

```

```

LIN00800
NO0810
NO0820
NO0830
NO0840
NO0850
NO0860
NO0870
NO0880
NO0890
NO0900
NO0910
NO0920
NO0930
NO0940
NO0950
NO0960
NO0970
NO0980
NO0990
NO1000
NO1010
NO1020
NO1030
NO1040
NO1050
NO1060
NO1070
NO1080
NO1090
NO1100
NO1110
NO1120
NO1130
NO1140
NO1150
NO1160
NO1170
NO1180
NO1190
NO1200
NO1210
NO1220
NO1230
NO1240
NO1250
NO1260
NO1270
NO1280
NO1290
NO1300
NO1310
NO1320
NO1330
NO1340
NO1350
NO1360
NO1370
NO1380
NO1390
NO1400
NO1410
NO1420
NO1430
NO1440
NO1450
NO1460
NO1470
NO1480
NO1490
NO1500
NO1510
NO1520
NO1530
NO1540
NO1550
NO1560
NO1570
NO1580

```

FILE: LINPLT

```
C
C
C      IF FREQ. LESS THAN ZERO. SET THE FREQ. TO 1
DO 124 I=1,10201
IF (PNTRS(I)) .EQ. 0.0) GO TO 127
IF (PNTRS(I)) .GE. 1.0) GO TO 127
PNTR(I) = 1
GO TO 128
127 PNTR(I) = PNTRS(I) * 0.5
128 CONTINUE
GO TO 131
129 CONTINUE
CALL READ(DAMPLT,PNTR,10201,ISTAT3)
130 IF (ISTAT3 .EQ. 1) GO TO 130
131 CONTINUE

C
C      SET BIN LEVELS
CALL SETMRG(66,0.66)
WRITE(6,HEAD)
WRITE(6,135)
135 FORMAT(//153,' PIXEL FREQUENCY SCATTER PLOT')
IF (NBINS .GT. MAXSUM) NBINS = MAXSUM
RANGE = MAXSUM / NBINS
IF (MOD(MAXSUM,NBINS) .NE. 0) RANGE = RANGE + 1
DO 140 I=1,NBINS
140 BINS(I) = RANGE*I
WRITE(6,150) (BINS(I),I=1,NBINS)
150 FORMAT(//17X,1H1,2X,13,9(3X,13))
NB = NBINS + 1
NBS = 2*NBINS
DO 155 J=1,4
WRITE(6,160) ((SYMTX(J),I=1,6),J=1,NBINS)
WRITE(6,163) ((SYMTX(J),I=1,6),J=NB,NBS)
163 FORMAT(1H*.16X,96A1)
155 CONTINUE
160 FORMAT(17X,96A1)
WRITE(6,165)
165 FORMAT(//)
DO 200 J=1,YSIZE
KK = (J-1)*XSIZE + 1
KKK = KK + YSIZE - 1
DO 180 I=1,XSIZE
II = (J-1)*XSIZE + I
IF (PNTR(II) .NE. 0) GO TO 170
PNTR(II) = BLANK
PNTR(KK) = BLANK
GO TO 180
170 BINLEV = PNTR(II) / RANGE
IF (MOD(PNTR(II),RANGE) .NE. 0) BINLEV = BINLEV + 1
PNTR(II) = SYMTX(BINLEV)
PNTR(KK) = SYMTX(BINLEV+NBINS)
180 CONTINUE

C
C      PRINT A LINE
L = 11 - J/10
IF (MOD(J,10) .EQ. 1) GO TO 190
WRITE(6,183)
183 FORMAT(16X,1H-)
WRITE(6,185) (PNTR(K),K=KK,KKK)
WRITE(6,185) (PNTR(K),K=1,101)
185 FORMAT(1H* .T17,101A1)
GO TO 200
190 WRITE(6,195) YAXIS(L)
195 FORMAT(10X,15,1X,1H*)
WRITE(6,197) (PNTR(K),K=KK,KKK)
WRITE(6,197) (PNTR(K),K=1,101)
197 FORMAT(1H* .T17,101A1)
200 CONTINUE

C
C      PRINT X-AXIS SCALES
WRITE(6,220) (XAXIS(I),I=1,DSIZ)
220 FORMAT(1H*.16X, 10(1H*.9(1H-),1H*/14X,11(13,7X))

C
TOTPTS = XPT + YPT
IF (TOTPTS .EQ. 0) RETURN
WRITE(6,225) TOTPTS
```

```
L NO 590
L NO 600
L NO 610
L NO 620
L NO 630
L NO 640
L NO 650
L NO 660
L NO 670
L NO 680
L NO 690
L NO 700
L NO 710
L NO 720
L NO 730
L NO 740
L NO 750
L NO 760
L NO 770
L NO 780
L NO 790
L NO 800
L NO 810
L NO 820
L NO 830
L NO 840
L NO 850
L NO 860
L NO 870
L NO 880
L NO 890
L NO 900
L NO 910
L NO 920
L NO 930
L NO 940
L NO 950
L NO 960
L NO 970
L NO 980
L NO 990
L NO2010
L NO2010
L NO2020
L NO2030
L NO2040
L NO2050
L NO2060
L NO2070
L NO2080
L NO2090
L NO2100
L NO2110
L NO2120
L NO2130
L NO2140
L NO2150
L NO2160
L NO2170
L NO2180
L NO2190
L NO2200
L NO2210
L NO2220
L NO2230
L NO2240
L NO2250
L NO2260
L NO2270
L NO2280
L NO2290
L NO2300
L NO2310
L NO2320
L NO2330
L NO2340
L NO2350
L NO2360
L NO2370
```

FILE: LINPLT

```
225 FORMAT(//) A TOTAL OF 100 POINTS WERE NOT DISPLAYED ON THE LINFLIN02380
* PRINTER GRAPH.// THE POINTS WERE OUT OF RANGE IN EITHER THE X DIRECTION LIN02390
* RECTION OR Y DIRECTION) LIN02400
CALL SETMRG(66.4.62) LIN02410
RETURN LIN02420
END LIN02430
```

16-8  
299

FILE: MATTNS

```
C      SUBROUTINE MATTNS(A,B,C,D,L,M)
C      MULTIPLY A BY B AND ADD D. STORE IN C
C      INTEGER B
C      DIMENSION A(L,M),B(M),C(L),D(L)
C      DO 20 I=1,L
C      SUM = 0.0
C      DO 10 K=1,M
10 SUM = SUM + A(I,K) * B(K)
20 C(I) = SUM + D(I)
C      RETURN
C      END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: OFFSET

```
C SURROUTINE OFFSET(YSCALE,XSCALE) OFF00010
C INTEGER BMKEY,RSCALE OFF00020
C INTEGER YSIZ,XSIZ,XHI,YHI,XLO OFF00030
C INTEGER CSCALE OFF00040
C INCLUDE COMRK6.LIST OFF00050
C INCLUDE CMK12.LIST OFF00060
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,ISFORM,ER,PTP,ERPKEY,MAPUNT,NUFILE,
* DRIMAD,DRMVD5,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* NMSTUN,NMSTFI,SCTRUN,MAPFIL
* DOTUNT,DOTFIL,NCHPAS,TRNSFL,RMTRFL,HISTFL,PCHUNT,
* CRDUNT,PHTIUNT,HANDIO
COMMON/SCATTER/SCALE,XSCALE,CLRVEC(30),NCLPCH,CLRKEY,LOG,
* FRFC,XMAX,YMAX,XMIN,YMIN,HCRGND,XHI,XLO,YLO,XSIZ,
* YHI,YSIZ,NRINS,SYMMTX(32),RMATRIX(60),HVEC(30),NHVCHN,NOFEAT
* SCALY,MFNADH,FLNADH,PNTADR,LDADR,NC,RMFEAT,RMCOMB
* NAVEC,TOTMNS,SIZE,DRMID,DRMID1,DRMCLN,DRMCR1,DRMTNS,DRMTN1,
* DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DAT1,NVEC,NOREAD,LREAD
* DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE
* NOSUM
CSEND OFF00100
C DIMENSION YSCALE(YSIZ),XSCALE(XSIZ) OFF00110
C SCALFS ARE COMPUTED IN ONE OF 3 WAYS : OFF00120
C 1) DATA HAS NOT BEEN TRANSFORMED OFF00130
C 2) DATA HAS BEEN TRANSFORMED AND RESCALED OFF00140
C 3) DATA HAS BEEN TRANSFORMED, BUT NOT RESCALED OFF00150
C 1) DATA HAS NOT BEEN TRANSFORMED OFF00160
C BMKEY = 0 OFF00170
C 2) DATA HAS BEEN TRANSFORMED AND RESCALED OFF00180
C HFSCALE = 1 OFF00190
C 3) DATA HAS BEEN TRANSFORMED, BUT NOT RESCALED OFF00200
C RFSCALE = 0 OFF00210
C CSCALE = 1 -- XSIZ AND/OR YSIZ HAS BEEN INPUT - MIN AND MAX
C WILL BE USED FOR HI AND LO PARAMETERS OFF00220
C IF (RMKEY .EQ. 0) GO TO A0 OFF00230
C IF (RSCALE .EQ. 1) GO TO A0 OFF00240
C IF (CSCALE .EQ. 1) GO TO A0 OFF00250
C XHIGH = XMAX OFF00260
C XLOW = XMIN OFF00270
C YHIGH = YMAX OFF00280
C YLOW = YMIN OFF00290
C GO TO 90 OFF00300
C XHIGH = XHI OFF00310
C XLOW = XLO OFF00320
C YHIGH = YHI OFF00330
C YLOW = YLO OFF00340
C 90 CONTINUE OFF00350
C XINC = (XHIGH - XLOW) / FLOAT(XSIZ-1) OFF00360
C YINC = (YHIGH - YLOW) / FLOAT(YSIZ-1) OFF00370
C XSCALE(1) = XLOW OFF00380
C YSCALE(1) = YLOW OFF00390
C DO 100 I=2,XSIZ OFF00400
C 100 XSCALE(I) = XSCALE(I-1) + XINC OFF00410
C DO 110 I=2,YSIZ OFF00420
C 110 YSCALE(I) = YSCALE(I-1) + YINC OFF00430
C RETURN OFF00440
C END OFF00450
OFF00460
OFF00470
OFF00480
OFF00490
OFF00500
OFF00510
```









FILE: SCATTR

```
130 I=I-1
    IF (I .EQ. 1 .AND. CLRKEY .EQ. 3) LSTLIN = -1
    TLIN = I
    YUPPER = YSCALE(I)
    YLOWER = YSCALE(I-1)
C
    DO 140 K=1,CH
    DO 140 J=1,XSIZ
140 IDATA(J,K) = BCKGND
C
143 IF (ISTAT3 .EQ. 1) GO TO 143
    COLLECT ALL POINTS THAT BELONG TO THIS LINE(I)
C
    THE DATA VECTORS WERE READ IN NVEC AT A TIME. EACH BLOCK OF DATA
    VECTORS HAS ITS OWN POINTER ARRAY FOR SORTING THE DATA VECTORS IN
    DESCENDING ORDER. EACH POINTER ARRAY PNTR(1...NOREAD) MUST BE
    SEARCHED FOR POINTS BELONGING TO LINE(I)
C
    DO 180 II=1,NOREAD
    K = (II-1)*NVEC
    IR = IRG(II)
145 IF (SWITCH(II)) LINADR(II) = (2*NVEC)*(II-1) + 2*IB + DRMTNS - 1
    IF (SWITCH(II)) CALL RREAD(LINADR(II),LINE(II),1,ISTA4)
146 IF (ISTAT4 .EQ. 1) GO TO 146
    IF (ILINE .NE. 1) GO TO 1465
    IF (LINE(II) .LE. YUPPER) GO TO 147
1465 CONTINUE
    IF (LINE(II) .LE. YUPPER .AND. LINE(II) .GT. YLOWER) GO TO 147
    SWITCH(II) = .FALSE.
    IF (ILINE .NE. YSIZ) GO TO 180
    IF (LINE(II) .GT. YUPPER) GO TO 147
    GO TO 180
C
    POSITION POINT IN X CO-ORDINATES
147 SAMADR = (2*NVEC)*(II-1) + 2*IB + DRMTNS - 2
    CALL RREAD(SAMADR,SAMPLE,1,ISTAT5)
    SWITCH(II) = .TRUE.
148 IF (ISTAT5 .EQ. 1) GO TO 148
    DO 150 J=1,XSIZ
    IPOSTN = J
    XLOWER = XSCALE(J)
    XUPPER = XSCALE(J+1)
C
    IF (J .NE. 1) GO TO 149
    IF (SAMPLE .LE. XLOWER) GO TO 160
149 CONTINUE
    IF (SAMPLE .GE. XLOWER .AND. SAMPLE .LT. XUPPER) GO TO 160
150 CONTINUE
C
    GET COLOR CODES
160 CALL CLPCOD(IR,MEANS,IDATA,IPOSTN,II)
    CALL CNTR(IR,IDATA,IPOSTN,II,COUNTR)
    IF (LOG .EQ. 1 .OR. FREQ .EQ. 1) CALL STOPTS(COUNTR,LINE(II),
        SAMPLE)
C
    CHECK NEXT VECTOR
    IF (IR .EQ. IFN(II)) GO TO 180
    IF (SWITCH(II)) IR = PNTR(II*K)
    IF (SWITCH(II)) IRG(II) = IR
    IF (SWITCH(II)) GO TO 145
C
180 CONTINUE
C
    WRITE A LINE
    CALL WRTLW(IDATA,LSTLIN)
200 IF (I.GT.1) GO TO 130
C
    IF (CLRKEY .EQ. 1) CALL CLRKYS(XSIZ,IDATA,NOSUB2,CH,MEANS,NC)
    IF (CLRKEY .EQ. 2) CALL CLRKYS(XSIZ,IDATA,NOSUB7,CH,MEANS,NC)
    IF (CLRKEY .EQ. 4) CALL CLRKYS(XSIZ,IDATA,NOFLD2,CH,MEANS,NC)
C
    IF (LOG .EQ. 1 .OR. FREQ .EQ. 1) CALL PRTPLT(RUFF,BUFF)
C
```

SCA01590  
SCA01600  
SCA01610  
SCA01620  
SCA01630  
SCA01640  
SCA01650  
SCA01660  
SCA01670  
SCA01680  
SCA01690  
SCA01700  
SCA01710  
SCA01720  
SCA01730  
SCA01740  
SCA01750  
SCA01760  
SCA01770  
SCA01780  
SCA01790  
SCA01800  
SCA01810  
SCA01820  
SCA01830  
SCA01840  
SCA01850  
SCA01860  
SCA01870  
SCA01880  
SCA01890  
SCA01900  
SCA01910  
SCA01920  
SCA01930  
SCA01940  
SCA01950  
SCA01960  
SCA01970  
SCA01980  
SCA01990  
SCA02000  
SCA02010  
SCA02020  
SCA02030  
SCA02040  
SCA02050  
SCA02060  
SCA02070  
SCA02080  
SCA02090  
SCA02100  
SCA02110  
SCA02120  
SCA02130  
SCA02140  
SCA02150  
SCA02160  
SCA02170  
SCA02180  
SCA02190  
SCA02200  
SCA02210  
SCA02220  
SCA02230  
SCA02240  
SCA02250  
SCA02260  
SCA02270  
SCA02280  
SCA02290  
SCA02300  
SCA02310  
SCA02320  
SCA02330  
SCA02340  
SCA02350  
SCA02360  
SCA02370

FILE: SCATTR

ORIGINAL PAGE IS  
OF POOR QUALITY

400 RETURN  
END

SCA02380  
SCA02390

~~16-15~~  
306

FILE: SETADR

SUBROUTINE SETADR(\*,\*,TOP,BUFF,LIMIT)

SETADR COMPUTES THE ADDRESS FOR STORING THE NDIM FILE ON DRUM  
AND ADDRESS FOR THE TWO ARRAYS - BUFF(LIMIT) AND ARRAY(TOP)

IMPLICIT INTEGER (A-Z)

INCLUDE COMK1.LIST  
INCLUDE COMK6.LIST  
INCLUDE COMK12.LIST  
INCLUDE COMT12.LIST

COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VAR5Z,TOTVT2,NOFLD2,  
AVAR2,COVAR2,CLSID2,SURNO2,SUROS2,FLDSV2,VERTX2,  
FETVC2(30),SUBVC2(75),SUBPTR(75),CLSV2(60),  
KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),  
GRPCMK(61),GROUPS(124)  
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,  
HISFIL,HISKEY,TRFORM,ERITP,ERPKEY,MAPUNT,NOFILE,  
DRUMAD,DRMADS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL  
NHSTUN,NHSTFI,SCTRUN,MAPFIL  
DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,  
CRDUNT,PRUNT,RANDIO  
COMMON/SCITFR/PSCALE,XYSCALE,CLRVEC(30),NCLRCH,CLRKEY,LOG,  
FREQ,XMAX,YMAX,XMIN,YMIN,BCKGND,XHI,XLO,YLO,YSIZ,  
YHI,YSIZ,NRINS,SYMTX(32),RMATX(60),RVFC(30),NRVCHN,NOFEAT  
SCALKY,MENADR,FLDADR,PNTADR,IDADR,NC,BMFEAT,BMCOMB  
NOVEC,TOTMNS,SIZE,DRMID,DRMID1,DRMCLR,DRMCR1,DRMTNS,DRMTN1,  
DRMCT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD  
DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE  
NOS(IH)

SCITFR IS A COMMON BLOCK LOADED ONLY WITH THE SCATTER PLOT  
PROCESSOR

PSCALE - KEY INDICATING THAT THE TRANSFORMED DATA IS TO BE RESCALED  
XYSCALE - KEY INDICATING THAT THE PIXEL FREQUENCY PLOT DATA IS  
TO BE RESCALED TO A RANGE OF 100  
CLRVEC - ARRAY CONTAINING THE COLOR CHANNELS  
NCLRCH - NO. OF COLOR CHANNELS  
CLRKEY - KEY INDICATING THE MANNER THAT THE COLORS ARE BEING

DEFINED :  
= 1 - INPUT BY STAT FILE  
= 2 - USER INPUT  
= 3 - RADIANCE VALUES FROM IMAGE TAPE- ON NDIM FILE  
= 4 - FIELD MEANS - ON NDIM FILE

LOG - KEY INDICATING LOG(2) OF FREQUENCY IS TO BE OUTPUT ON  
LINE PRINTER PLOT  
FREQ - KEY INDICATING FREQUENCY IS TO BE OUTPUT ON LINE PRINT-  
ER PLOT  
YMAX - MAXIMUM VALUE OF 1ST COMPONENT OF TRANS. DATA  
YMAX - MAXIMUM VALUE OF 2ND COMPONENT OF TRANS. DATA  
YMIN - MINIMUM VALUE OF 1ST COMPONENT OF TRANS. DATA  
YMIN - MINIMUM VALUE OF 2ND COMPONENT OF TRANS. DATA  
RCKGND - COLOR FOR TAPE BACKGROUND  
XHI - UPPER LIMIT OF SCAN LINE PARAMETER  
XLO - LOWER LIMIT OF SCAN LINE PARAMETER  
YHI - UPPER LIMIT OF LINE NO. PARAMETER  
YLO - LOWER LIMIT OF LINE NO. PARAMETER  
YSIZ - NO. OF SAMPLES PER SCAN LINE

CONTINUE  
YSIZ - NO. OF LINES TO OUTPUT ON TAPE  
NRINS - NO. OF BIN LEVELS OR SYMBOLS FOR PIXEL FREQ. PLOT  
SYMTX - ARRAY CONTAINING SYMBOLS FOR PIXEL FREQ. PLOT  
RMATX - ARRAY CONTAINING R-MATRIX  
RVFC - ARRAY CONTAINING ADDITIVE VECTOR  
NRVCHN - NO. OF ADDITIVE VECTOR ELEMENTS  
NOFEAT - NOFEAT + NCLRCH  
SCALKY - KEY INDICATING THE MANNER OF COLLECTING THE MIN AND MAX  
VALUES :

= 1 - USER INPUT  
= 2 - COMPUTE FROM NDIM FILE  
MENADR - ADDRESS FOR STORING MEANS  
FLDADR - ADDRESS FOR STORING FIELD INFO  
IDADR - ADDRESS FOR STORING IDS  
NC - NO. OF CHANNELS FOR COLORS TO BE OUTPUT ON TAPE  
BMFEAT - NO. OF CHANNELS IN B-MATRIX  
BMCOMB - NO. OF LINEAR COMB.  
NOVEC - NO. OF VECTORS ON NDIM FILE  
TOTMNS - NO. OF MEAN ELEMENTS

SET00010  
SET00020  
SET00030  
SET00040  
SET00050  
SET00060  
SET00070  
SET00080  
SET00090  
SET00100  
SET00110  
SET00120  
SET00130  
SET00140  
SET00150  
SET00160  
SET00170  
SET00180  
SET00190  
SET00200  
SET00210  
SET00220  
SET00230  
SET00240  
SET00250  
SET00260  
SET00270  
SET00280  
SET00290  
SET00300  
SET00310  
SET00320  
SET00330  
SET00340  
SET00350  
SET00360  
SET00370  
SET00380  
SET00390  
SET00400  
SET00410  
SET00420  
SET00430  
SET00440  
SET00450  
SET00460  
SET00470  
SET00480  
SET00490  
SET00500  
SET00510  
SET00520  
SET00530  
SET00540  
SET00550  
SET00560  
SET00570  
SET00580  
SET00590  
SET00600  
SET00610  
SET00620  
SET00630  
SET00640  
SET00650  
SET00660  
SET00670  
SET00680  
SET00690  
SET00700  
SET00710  
SET00720  
SET00730  
SET00740  
SET00750  
SET00760  
SET00770  
SET00780  
SET00790

FILE: SETADR

```
C* SIZE - NO. OF WORDS FOR PACKED HISTOGRAMMED VECTOR. SF T00800
C* DRMID - BEGINNING DRUM ADDRESS FOR STORING IDS. SF T00810
C* DRMID1 - SUMMING DRUM ADDRESS FOR RETRIV. IDS SF T00820
C* DRMCLR - BEGINNING DRUM ADDRESS FOR STORING COLORS SF T00830
C* DRMCR1 - SUMMING DRUM ADDRESS FOR RETRIV. COLORS SF T00840
C* DRMTNS - BEGINNING DRUM ADDRESS FOR STORING TRANS. DATA SF T00850
CONTINUE SF T00860
C* DRMTN1 - SUMMING DRUM ADDRESS FOR RETRIV. TRANS. DATA SF T00870
C* DRMCNT - BEGINNING DRUM ADDRESS FOR STORING FREQ. SF T00880
C* DRMCT1 - SUMMING DRUM ADDRESS FOR RETRIV. FREQ. SF T00890
C* DRMVFC - BEGINNING DRUM ADDRESS FOR STORING VECTORS SF T00900
C* DRMVC1 - SUMMING DRUM ADDRESS FOR RETRIV. VECTORS SF T00910
C* VECTR1 - ADDRESS IN 'ARRAY' FOR STORING TRANSF. VECTOR SF T00920
C* DATA1 - SAME ADDRESS AS VECTR1 - USED FOR CREATING A SCAN LINE SF T00930
C* - OF DATA OUTPUT TO THE SCATTER PLOT TAPE SF T00940
NVFC - NO. OF VECTORS TO READ FROM DRUM AT ONE TIME SF T00950
NOREAD - NO. OF READS TO DRUM SF T00960
LREAD - NO. OF VECTOR TO READ ON LAST DRUM READ SF T00970
DRMPTR - BEGINNING DRUM ADDRESS FOR STORING POINTERS SF T00980
FETVEC - ARRAY FOR CONTAINING FETVC2 AND CLRVEC CHANNELS SF T00990
DRMPLT - BEGINNING ADDRESS FOR STORING FREQ. PLOT IMAGE SF T01000
CSCALE - KEY INDICATING THAT XSIZ OR YSIZ PARAMETERS HAVE BEEN SF T01010
INPUT SF T01020
CSEND SF T01030
DIMENSION BUFF(1) SF T01040
C SF T01050
PEAD(NHSTUN,END=150) NOFLD2,NOSUB,TOTVT2,NOVEC SF T01060
IF (CLRKEY .EQ. 1) GO TO 100 SF T01070
IF (CLRKEY .EQ. 2) GO TO 100 SF T01080
IF (CLRKEY .EQ. 3) GO TO 120 SF T01090
IF (CLRKEY .EQ. 4) GO TO 130 SF T01100
C SF T01110
100 MENADR = 1 SF T01120
FLDADR = MENADR + 60*NC SF T01130
VERTX2 = FLDADR + 4*NOFLD2 SF T01140
DATA1 = FLDADR SF T01150
GO TO 133 SF T01160
C SF T01170
120 FLDADR = 1 SF T01180
VERTX2 = FLDADR + NOFLD2*4 SF T01190
DATA1 = FLDADR SF T01200
GO TO 133 SF T01210
C SF T01220
130 MENADR = 1 SF T01230
FLDADR = MENADR + TOTMNS SF T01240
VERTX2 = FLDADR + 4*NOFLD2 SF T01250
DATA1 = FLDADR SF T01260
C SF T01270
COMPUTE MAXIMUM NO. OF VECTORS ARRAY MAY HOLD AT ONE TIME SF T01280
C SF T01290
133 VECTR1 = DATA1 SF T01300
NVFC = (TOP - DATA1) / 2 SF T01310
VECTR2 = LIMIT / SIZE SF T01320
IF (VECTR2 .LT. NVFC) NVFC = VECTR2 SF T01330
IF (NOVEC .LT. NVFC) GO TO 135 SF T01340
NOREAD = NOVEC / NVFC SF T01350
IF (MOD(NOVEC,NVEC) .NE. 0) NOREAD = NOREAD + 1 SF T01360
LREAD = MOD(NOVEC,NVEC) SF T01370
IF (LREAD .EQ. 0) LREAD = NVFC SF T01380
GO TO 140 SF T01390
135 NOREAD = 1 SF T01400
NVFC = NOVEC SF T01410
LREAD = NVFC SF T01420
C SF T01430
ADDRESSES FOR HIGH SPEED DRUM SF T01440
C SF T01450
140 DRMVFC = DRMADR SF T01460
DRMID = DRMVFC + NOVEC*SIZE SF T01470
DRMCNT = DRMID + NOVEC SF T01480
IF (CLRKEY .EQ. 3) DRMCLR = DRMCNT + NOVEC SF T01490
IF (CLRKEY .EQ. 3) DRMCR1 = DRMCNT SF T01500
DRMTNS = DRMCLR + NOVEC SF T01510
DRMPTR = DRMTNS + NOVEC*2 SF T01520
DRMPLT = DRMADR + NOVEC SF T01530
TOTDPM = DRMPLT SF T01540
SF T01550
SF T01560
SF T01570
SF T01580
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: SETADR

```
IF (LOG .EQ. 1 .OR. FREQ .EQ. 1) TOTDRM = DRMPLT + 10201
IF ((TOTDRM-DRUMAD) .LE. DRMWDS) GO TO 143
WRITE(A,142)TOTDRM,DRMWDS
142 FORMAT(/' NOT ENOUGH DRUM SPACE. '// TOTAL WORDS OF DRUM SPACE =',
* I12/' TOTAL WORDS OF DRUM SPACE AVAILABLE =',I12)
CALL CMERR
143 CONTINUE
DRMVC1 = DRMVEC
DRMID1 = DRMID
DRMCR1 = DRMCLA
DRMCT1 = DRMCNT
DRMTN1 = DRMTNS
DRMPT1 = DRMPTR
C
C
C ZERO OUT AREA OF DRUM
IF (LOG .EQ. 0 .AND. FREQ .EQ. 0) GO TO 145
DO 147 I=1,10201
147 RUFF(I) = 0
CALL RWRITE(DRMPLT,BUFF,10201,ISTAT)
146 IF (ISTAT .EQ. 1) GO TO 146
145 RETURN 1
C
150 DEWIND NHSTUN
RETURN 2
END
```

SET01590  
SET01600  
SET01610  
SET01620  
SET01630  
SET01640  
SET01650  
SET01660  
SET01670  
SET01680  
SET01690  
SET01700  
SET01710  
SET01720  
SET01730  
SET01740  
SET01750  
SET01760  
SET01770  
SET01780  
SET01790  
SET01800  
SET01810  
SET01820  
SET01830  
SET01840



FILE: SET11

```
C
C CALL REREAD(30,H0)
C NOW READ CARD INTO BUFFER
105 READ(21,106)(ACARD(I),I=1,20)
106 FORMAT(20A4)
WRITE(30,106)(ACARD(I),I=1,20)
REWIND 30
READ(30,110)CODE1,CARD
110 FORMAT(A4,6X,62A1)
REWIND 30
COL = 0

C
C WRITE(6,120)CODE1,CARD
120 FORMAT(1X,A4,6X,62A1)

C
C DO 130 I=1,NPUT
IF (CODE1 .EQ. CODE(I)) GO TO (150,180,210,250,290,300,340,
* 350,370,390,400,410,415,420,455,460,470,477).I
130 CONTINUE
WRITE(6,140)
140 FORMAT(' INVALID CONTROL CARD -- IGNORED')
GO TO 105

C
C CHANNEL CARD -- NEEDED ONLY IF STATS FILE IS INPUT
C
150 M=NXTCHR(CARD,COL)
IF (M .NE. BLANK) GO TO 160
153 WRITE(6,155)
155 FORMAT(' ERROR ON CHANNELS CARD')
GO TO 105
160 COL = COL - 1
NOFEAT = NUMBER(CARD,COL,FETVEC,NOFEAT)
CALL ORDER(FETVEC,NOFEAT)
NC = NOFEAT
CLPKFY = 1
GO TO 105

C
C STAT FILE CARD
C
180 M = NXTCHR(CARD,COL)
IF (M .EQ. BLANK) GO TO 105
IF (M .EQ. URCD) GO TO 190
IF (M .EQ. FRCD) GO TO 200
185 WRITE(6,187)
187 FORMAT(' ERROR ON STAT FILE CARD')
GO TO 105
190 J = FIND12(CARD,COL,EQUOM)
IF (J .NE. 2) GO TO 185
M = NUMBER(CARD,COL,SAVTAP,ZERO)
COL = COL - 1
CLPKFY = 1
GO TO 180
200 J = FIND12(CARD,COL,EQUOM)
IF (J .NE. 2) GO TO 185
M = NUMBER(CARD,COL,STAFIL,ZERO)
COL = COL - 1
GO TO 180

C
C N-DIM HISTOGRAM FILE
C
210 M = NXTCHR(CARD,COL)
IF (M .EQ. BLANK) GO TO 105
IF (M .EQ. URCD) GO TO 230
IF (M .EQ. FRCD) GO TO 240
220 WRITE(6,225)
225 FORMAT(' ERROR ON N-DIM HISTOGRAM FILE CARD')
GO TO 105
230 J = FIND12(CARD,COL,EQUOM)
IF (J .NE. 2) GO TO 220
M = NUMBER(CARD,COL,NHSTUN,ZERO)
COL = COL - 1
GO TO 210
240 J = FIND12(CARD,COL,EQUOM)
IF (J .NE. 2) GO TO 220
M = NUMBER(CARD,COL,NHSTFI,ZERO)
COL = COL - 1
GO TO 210

C
C PIXEL FREQ. PLOT CARD
```

```
SET00800
SET00810
SET00820
SET00830
SET00840
SET00850
SET00860
SET00870
SET00880
SET00890
SET00900
SET00910
SET00920
SET00930
SET00940
SET00950
SET00960
SET00970
SET00980
SET00990
SET01000
SET01010
SET01020
SET01030
SET01040
SET01050
SET01060
SET01070
SET01080
SET01090
SET01100
SET01110
SET01120
SET01130
SET01140
SET01150
SET01160
SET01170
SET01180
SET01190
SET01200
SET01210
SET01220
SET01230
SET01240
SET01250
SET01260
SET01270
SET01280
SET01290
SET01300
SET01310
SET01320
SET01330
SET01340
SET01350
SET01360
SET01370
SET01380
SET01390
SET01400
SET01410
SET01420
SET01430
SET01440
SET01450
SET01460
SET01470
SET01480
SET01490
SET01500
SET01510
SET01520
SET01530
SET01540
SET01550
SET01560
SET01570
SET01580
```

FILE: SET11

C

```

250 M = NXTCHR(CARD,COL)
    IF (M.EQ.BLANK) GO TO 105
    IF (M.EQ.LRCD) GO TO 270
    IF (M.EQ.FRCD) GO TO 280
    IF (M.EQ.RHCD) GO TO 285
255 WRITE(6,250)
260 FORMAT(' ERROR ON OPTION CARD')
    GO TO 105
270 LOG = 1
    GO TO 287
280 FRFO = 1
    GO TO 287
285 XYSCL = 1
287 J = FIND12(CARD,COL, EQUOM)
    IF (J.EQ.3) GO TO 250
    GO TO 105

```

C

COLOR CODES

```

290 NCLSTR = VECSCN(MEANS(ADR*1),NOCHAN,CARD,COL) + NCLSTR
    ADP = NCLSTR*NOCHAN
    NC = NOCHAN
    NOSIIR2 = NCLSTR
    CLPKEY = 2
    GO TO 105

```

C

TAPE SIZE CARD

```

300 M = NXTCHR(CARD,COL)
    IF (M.EQ.BLANK) GO TO 105
    IF (M.EQ.XRCD) GO TO 320
    IF (M.EQ.YRCD) GO TO 330
310 WRITE(315)
315 FORMAT(' ERROR ON TAPE SIZE CARD')
    GO TO 105
320 M = NXTCHR(CARD,COL)
    J = FIND12(CARD,COL, EQUOM)
    IF (M.EQ.SHCD) CSCALE = 1
    IF (J.NE.2) GO TO 310
    MM = NUMBER(CARD,COL,J,ZERO)
    IF (M.EQ.HHCD) XHI = J
    IF (M.EQ.LHCD) XLO = J
    IF (M.EQ.SHCD) XSIZ = J
    COL = COL - 1
    GO TO 300
330 M = NXTCHR(CARD,COL)
    IF (M.NE.SHCD) SCSALE = 1
    J = FIND12(CARD,COL, EQUOM)
    IF (J.NE.2) GO TO 310
    MM = NUMBER(CARD,COL,J,ZERO)
    IF (M.EQ.HHCD) YHI = J
    IF (M.EQ.LHCD) YLO = J
    IF (M.EQ.SHCD) YSIZ = J
    COL = COL - 1
    GO TO 300

```

C

SYMBOLS CARD

```

340 M = NXTCHR(CARD,COL)
    IF (M.EQ.BLANK) GO TO 105
    IF (M.EQ.KOMMA) GO TO 340
    ICNT = ICNT + 1
    SYMMIX(ICNT) = M
    GO TO 340

```

C

MODULE STAT CARD DECK

```

350 CALL CPDSTA(BUFF, TOP)
    GO TO 105

```

C

DATE CARD

```

370 M = NXTCHR(CARD,COL)
    IF (M.EQ.BLANK) GO TO 105
    READ(30,380) DATE
380 FORMAT(10X,15A4)
    REWIND 30
    GO TO 105

```

SET01590  
SET01600  
SET01610  
SET01620  
SET01630  
SET01640  
SET01650  
SET01660  
SET01670  
SET01680  
SET01690  
SET01700  
SET01710  
SET01720  
SET01730  
SET01740  
SET01750  
SET01760  
SET01770  
SET01780  
SET01790  
SET01800  
SET01810  
SET01820  
SET01830  
SET01840  
SET01850  
SET01860  
SET01870  
SET01880  
SET01890  
SET01900  
SET01910  
SET01920  
SET01930  
SET01940  
SET01950  
SET01960  
SET01970  
SET01980  
SET01990  
SET02000  
SET02010  
SET02020  
SET02030  
SET02040  
SET02050  
SET02060  
SET02070  
SET02080  
SET02090  
SET02100  
SET02110  
SET02120  
SET02130  
SET02140  
SET02150  
SET02160  
SET02170  
SET02180  
SET02190  
SET02200  
SET02210  
SET02220  
SET02230  
SET02240  
SET02250  
SET02260  
SET02270  
SET02280  
SET02290  
SET02300  
SET02310  
SET02320  
SET02330  
SET02340  
SET02350  
SET02360  
SET02370

ORIGINAL PAGE IS  
OF POOR QUALITY



FILE: SET11

```
C COMMENT CARD
390 M = NXTCHR(CARD,COL)
IF (M.EQ.BLANK) GO TO 105
READ(30,340)COMMENT
REWIND 30
GO TO 105

C
C
C HED1
400 M = NXTCHR(CARD,COL)
IF (M.EQ.BLANK) GO TO 105
READ(30,380) HED1
REWIND 30
GO TO 105

C
C
C HED2
410 M = NXTCHR(CARD,COL)
IF (M.EQ.BLANK) GO TO 105
READ(30,390) HED2
REWIND 30
GO TO 105

C
C
C SCATTER PLOT TAPE CARD
415 M = NXTCHR(CARD,COL)
IF (M.EQ.BLANK) GO TO 105
IF (M.EQ.UHCD) GO TO 418
416 WRITE(6,417)
417 FORMAT(' ERROR ON SCATTER PLOT TAPE CARD')
GO TO 105
418 J = FIND12(CARD,COL,EGUCOM)
IF (J.EQ.0) GO TO 416
M = NUMP(CARD,COL,SCRUN,ZERO)
GO TO 105

C
C
C
C
C B - MATRIX CARD
RMKEY = 1 FOR CARDS
RMKEY = 2 FOR FILE
420 M = NXTCHR(CARD,COL)
IF (M.EQ.BLANK) GO TO 433
IF (M.EQ.CHCD) GO TO 450
IF (M.EQ.FHCD) GO TO 450
433 WRITE(6,435)
435 FORMAT(' ERROR ON B-MATRIX CARD')
GO TO 105
440 RMKEY = 1
CALL BMFIL(BMATRX,BMCOMB,BMFEAT,BMVEC,BMKEY)
GO TO 105
450 RMKEY = 2
CALL BMFIL(BMATRX,BMCOMB,BMFEAT,BMVEC,BMKEY)
GO TO 105

C
C
C BACK GROUND COLOR CARD
455 M = NXTCHR(CARD,COL)
IF (M.EQ.BLANK) GO TO 105
IF (M.EQ.HHCD) BCKGND = 0
IF (M.EQ.WHCD) BCKGND = 255
GO TO 105

C
C
C R VECTOR
460 NBVCHN = FINTNM(CARD,COL,BVEC(NBVCHN+1),30) + NBVCHN
GO TO 105

C
C
C SCALING CARD
470 M = NXTCHR(CARD,COL)
IF (M.EQ.BLANK) GO TO 105
IF (M.EQ.FHCD) GO TO 471
IF (M.EQ.YHCD) GO TO 472
IF (M.EQ.XHCD) GO TO 473
IF (M.EQ.WHCD) GO TO 476
GO TO 474
471 SCALRY = 2
GO TO 105
471 J = FIND12(CARD,COL,EGUCOM)
```

```
SET02380
SET02390
SET02400
SET02410
SET02420
SET02430
SET02440
SET02450
SET02460
SET02470
SET02480
SET02490
SET02500
SET02510
SET02520
SET02530
SET02540
SET02550
SET02560
SET02570
SET02580
SET02590
SET02600
SET02610
SET02620
SET02630
SET02640
SET02650
SET02660
SET02670
SET02680
SET02690
SET02700
SET02710
SET02720
SET02730
SET02740
SET02750
SET02760
SET02770
SET02780
SET02790
SET02800
SET02810
SET02820
SET02830
SET02840
SET02850
SET02860
SET02870
SET02880
SET02890
SET02900
SET02910
SET02920
SET02930
SET02940
SET02950
SET02960
SET02970
SET02980
SET02990
SET03000
SET03010
SET03020
SET03030
SET03040
SET03050
SET03060
SET03070
SET03080
SET03090
SET03100
SET03110
SET03120
SET03130
SET03140
SET03150
SET03160
```

FILE: SET11

```

IF (J .EQ. 3) GO TO 470
NO TO 105
472 M = NITCHR(CARD,COL)
M = NITCHR(CARD,COL)
J = FIND12(CARD,COL,EOUCOM)
IF (J .NE. 2) GO TO 474
J = FLTNUM(CARD,COL,NUM,1)
IF (M .EQ. TRCD) YMIN = NUM
IF (M .EQ. ABCD) YMAX = NUM
SCALY = 1
GO TO 470
473 M = NITCHR(CARD,COL)
M = NITCHR(CARD,COL)
J = FIND12(CARD,COL,EOUCOM)
IF (J .NE. 2) GO TO 474
J = FLTNUM(CARD,COL,NUM,1)
IF (M .EQ. TRCD) XMIN = NUM
IF (M .EQ. ABCD) XMAX = NUM
SCALX = 1
GO TO 470
474 WRITE(6,475)
475 FORMAT(' ERROR ON SCALING CARD')
NO TO 105
476 RSCALE = 1
GO TO 471

*END*

477 CONTINUE

480 IF (BMKEY .LE. 0) GO TO 490
IF (LOG .EQ. 1) GO TO 481
IF (FRFQ .EQ. 0) GO TO 482
481 IF (RSCALE .EQ. 1) GO TO 482
WRITE(6,480)
480 FORMAT(' DATA MUST BE RESCALED BEFORE PIXEL FREQUENCY PLOT OPTION
MAY BE SELECTED')
LOG = 0
FRFQ = 0
482 CONTINUE
CALL WPTBM(BMATRIX,BMCOMB,BMFEAT,BMVEC)
GO TO 490
483 WRITE(6,485)NOFET2,BMFEAT
485 FORMAT(' NO. OF PLOTTING CHANNELS, NO. OF B-MATRIX CHANNELS:
. , . RESPECTIVELY')
CALL CMERR

490 CONTINUE
CHANGE MEANS TO FLOATING PT NO
IF (INCLSTR .EQ. 0) GO TO 497
NO 495 I=1,ADR
495 MENS(I) = MFANS(I)
497 CONTINUE
COMPUTE NO. OF BINS
IF (ICNT .GT. 0) NBINS = ICNT/2
READ HEADER REC FROM N-BEM HIST FILE
READ(NHSTUN) TOTMNS,SIZE,NOFET2,(FETVC2(I),I=1,NOFET2),
NCLRCH,(CLRVEC(I),I=1,NCLRCH)

DEFAULT STAT CHANNELS
IF (CLPKEY .NE. 1) GO TO 499
IF (NOFFAT .NE. 0) GO TO 499
NO 499 I=1,NOFET2
IF (I .GT. 4) GO TO 499
NOFAT = I
NC = I
499 FETVFC(I) = FETVC2(I)
CONTINUE
IF (BMKEY .NE. 0 .AND. NOFET2 .NE. BMFEAT) GO TO 483

```

ORIGINAL PAGE IS OF POOR QUALITY

SET03178  
SET03180  
SET03190  
SET03208  
SET03210  
SET03220  
SET03230  
SET03240  
SET03250  
SET03260  
SET03270  
SET03280  
SET03290  
SET03300  
SET03310  
SET03320  
SET03330  
SET03340  
SET03350  
SET03360  
SET03370  
SET03380  
SET03390  
SET03400  
SET03410  
SET03420  
SET03430  
SET03440  
SET03450  
SET03460  
SET03470  
SET03480  
SET03490  
SET03500  
SET03510  
SET03520  
SET03530  
SET03540  
SET03550  
SET03560  
SET03570  
SET03580  
SET03590  
SET03600  
SET03610  
SET03620  
SET03630  
SET03640  
SET03650  
SET03660  
SET03670  
SET03680  
SET03690  
SET03700  
SET03710  
SET03720  
SET03730  
SET03740  
SET03750  
SET03760  
SET03770  
SET03780  
SET03790  
SET03800  
SET03810  
SET03820  
SET03830  
SET03840  
SET03850  
SET03860  
SET03870  
SET03880  
SET03890  
SET03900  
SET03910  
SET03920  
SET03930  
SET03940  
SET03950

FILE: SET11

C ARE COLOR CODES COMING FROM TAPE

IF (NCLRCH .EQ. 0) GO TO 505

CLRKEY = 3

NC = NCLRCH

505 IF (TOTMNS .EQ. 0) GO TO 500

CRKEY = 4

NC = NOFET2

500 CONTINUE

C

POSITION TO DESIRED FILE

CALL FSRFL(NHSTUN,NHSTFI,ISTAT)

IF (ISTAT .EQ. 0) GO TO 520

WRITE(6,510)NHSTFI,ISTAT

510 FORMAT(' ERROR IN POSITIONING N-DIM HIST FILE TO FILE')

\* 15// ISTAT = ',I5)

REWIND NHSTUN

CALL CMERR

520 CONTINUE

C

WRITE(6,700)

IF (FRFQ .EQ. 1) WRITE(6,710)

IF (LOG .EQ. 1) WRITE(6,720)

IF (RMKEY .GT. 0) WRITE(6,740)

IF (PCKGND .EQ. 0) WRITE(6,750)

IF (PCKGND .EQ. 255) WRITE(6,760)

IF (RSCALF .EQ. 0 .AND. RMKEY .GT. 0) WRITE(6,770)

IF (RSCALF .EQ. 2 .AND. RMKEY .GT. 0) WRITE(6,780)

IF (SCALFY .EQ. 1) WRITE(6,790)

IF (CLRKEY .EQ. 1) WRITE(6,800) (FETVEC(I), I=1,NOFET)

IF (CLRKEY .EQ. 2) WRITE(6,810)

IF (CLRKEY .EQ. 3) WRITE(6,820)

IF (CLRKEY .EQ. 4) WRITE(6,830)

700 FORMAT('// USER HAS SELECTED THE FOLLOWING OPTIONS :',//)

710 FORMAT(' LINE PRINTER PIXEL FREQ. PLOT')

720 FORMAT(' LINE PRINTER LOG OF PIXEL FREQ. PLOT')

740 FORMAT(' TRANSFORM DATA')

750 FORMAT(' HACK GROUND COLOR FOR SPECTRAL PLOTS WILL BE BLACK')

760 FORMAT(' HACK GROUND COLOR FOR SPECTRAL PLOTS WILL BE WHITE')

770 FORMAT(' TRANSFORMED DATA WILL NOT BE RESCALED')

790 FORMAT(' USER HAS INPUT RESCALING FACTORS')

780 FORMAT(' RESCALING FACTORS WILL BE COMPUTED FROM N-DIM HIST FILE')

800 FORMAT(' COLOR CODES ARE COMING FROM STAT FILE USING CHANNELS '

\* 4(1X,I2)

810 FORMAT(' COLOR CODES ARE COMING FROM USER INPUT')

820 FORMAT(' COLOR CODES WILL BE COMPUTED FROM FIELD MEANS')

830 FORMAT(' COLOR CODES ARE COMING FROM RADIANCE VALUES STORED ON N-

DIM HIST FILE')

RETURN

C

END

SET03960  
SET03970  
SET03980  
SET03990  
SET04000  
SET04010  
SET04020  
SET04030  
SET04040  
SET04050  
SET04060  
SET04070  
SET04080  
SET04090  
SET04100  
SET04110  
SET04120  
SET04130  
SET04140  
SET04150  
SET04160  
SET04170  
SET04180  
SET04190  
SET04200  
SET04210  
SET04220  
SET04230  
SET04240  
SET04250  
SET04260  
SET04270  
SET04280  
SET04290  
SET04300  
SET04310  
SET04320  
SET04330  
SET04340  
SET04350  
SET04360  
SET04370  
SET04380  
SET04390  
SET04400  
SET04410  
SET04420  
SET04430  
SET04440  
SET04450  
SET04460  
SET04470

FILE: SORTVC

```

C*
C* SURROUTINE SORTVC(HIST,PNTR,ICOL,NOVEC,IRG,IEN,II)
C*
C* SORTVC SORTS THE ARRAY HIST INTO DESCENDING ORDER
C*
C* HIST - THE ARRAY TO BE ORDERED
C* PNTR - ARRAY CONTAINING POINTERS
C* ICOL - THE COLUMN WITHIN HIST THAT IS TO BE ORDERED
C* NOVEC - NO. OF VECTORS TO SORT
C* IRG - ARRAY CONTAINING BEGINNING POINTER FOR EACH BLOCK SORTED
C* IEN - ARRAY CONTAINING ENDING POINTER FOR EACH BLOCK SORTED
C* II - NO. OF BLOCK BEING SORTED
C*
C* IMPLICIT INTEGER (A-Z)
C
C REAL HIST(ICOL,NOVEC),NUMBR
C
C DIMENSION PNTR(1)
C DIMENSION IRG(1),IEN(1)
C
C IB = 1
C IE = 1
C NEX = IB
C DO 200 J=1,NOVEC
C
C NUMBR = HIST(ICOL,J)
C COMPARE AGAINST LARGEST NUMBER
C
C IF (NUMBR .GE. HIST(ICOL,IB)) GO TO 100
C
C COMPARE AGAINST SMALLEST NUMBER
C
C IF (NUMBR .LE. HIST(ICOL,IE)) GO TO 120
C
C COMPARE AGAINST NEXT LARGEST NUMBER
C INTRY = PNTR(NEX)
C NOVEC1 = J - 1
C DO 80 I=1,NOVEC1
C PAST = NEX
C NEX = PNTR(NEX)
C IF (NUMBR .GE. HIST(ICOL,NEX)) GO TO 130
C IF (NUMBR .GT. HIST(ICOL,INTRY)) GO TO 130
C PAST = INTRY
C INTRY = PNTR(INTRY)
C 80 CONTINUE
C WRITE(6,90)
C 90 FORMAT(' ERROR IN SORTING VECTORS')
C STOP
C
C LARGEST NUMBER FOUND THUS FAR
C
C 100 PNTR(J) = IB
C IB = J
C NEX = IB
C GO TO 200
C
C SMALLEST NUMBER FOUND THUS FAR
C
C 120 PNTR(IE) = J
C IE = J
C GO TO 200
C
C
C 130 PNTR(J) = NEX
C NEX = IB
C PNTR(PAST) = J
C 200 CONTINUE
C
C IRG(II) = IB
C IEN(II) = IE
C RETURN
C
C END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: STOFIL

```

SUBROUTINE STOFIL(LIMIT,MEANS,BUFF)
C C C
STOFIL READS AND STORES NDIM FILE ON DRUM
C C C
IMPLICIT INTEGER (A-Z)
C C C
INCLUDE CMRK12.LIST
INCLUDE COMMK6.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,RMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWD5,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVFL
* .NHSTUN,NHSTFI,SCRUN,MAPFIL
* .DOTUNT,DOTFIL,NCHPAS,TRNSFL,RMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRTUNT,RANDIO
COMMON/SCATTER/RSCALE,XYSCL,CLRVEC(30),NCLRCH,CLRKEY,LOG,
* FREQ,XMAX,YMAX,XMIN,YMIN,RCKGND,XHI,XLO,YLO,XSIZ,
* YHI,YSIZ,NBINS,SYMTX(32),RMATX(60),RVEC(30),NBVCHN,NOFEAT
* .SCALKY,MFNADR,FLDADR,PNTADR,IDADR,NC,RMFEAT,BMCOMP
* .NOVEC,TOTMNS,SIZ,DRMID,DRMID1,DRMCLR,DRMCRI,DRMTNS,DRMTN1,
* DRMCNT,DRMCT1,DRMVC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD
* DRMPT,DRMPT1,FETVEC(16),DRMPLT,CSCALE
* .NOSUB
C C C
CSEND
C
DIMENSION BUFF(1),ISTAT(4,2),ISTA(3,2),MEANS(1)
DIMENSION REMDR(4)
C C C
READ MEANS INTO CORE
IF (CLRKEY .EQ. 4) READ(NHSTUN)(MEANS(I),I=1,TOTMNS)
AMT = 0
TOTAL = NOVEC*SIZE
DO 100 I=1,4
REMD = TOTAL - LIMIT*I
IF (REMD .GT. 0) GO TO 90
REMDR(I) = TOTAL - LIMIT*(I-1)
NTPRDS = I
GO TO 105
90 REMDR(I) = LIMIT
100 CONTINUE
105 CONTINUE
C C C
READ IN REC 4 -- DATA VECTORS
DO 130 I=1,NTPRDS
NOWRDS = REMDR(I)
READ (NHSTUN)(BUFF(J),J=1,NOWRDS)
C C C
DUMP ON HIGH SPEED FRUM
CALL RWRITE(DRMVC1,BUFF(1),NOWRDS,ISTAT(I,1))
DRMVC1 = DRMVC1 + NOWRDS
130 CONTINUE
IF (CLRKEY .EQ. 3) NOREC = 3
IF (CLRKEY .NE. 3) NOREC = 2
C C C C C
READ IN REC 5 -- ID ARRAY
READ IN REC 6 -- COUNTERS
READ IN REC 7 -- COLOR CODES
DO 200 K=1,NOREC
NOWRDS = NOVEC
READ(NHSTUN)(BUFF(J),J=1,NOWRDS)
C C C
DUMP ON HIGH SPEED DRUM
IF (K .EQ. 1) GO TO 170
IF (K .EQ. 2) GO TO 180
IF (K .EQ. 3) GO TO 190
170 CALL RWRITE(DRMID1,BUFF(1),NOWRDS,ISTA(K,1))
DRMID1 = DRMID1 + NOWRDS
GO TO 200
180 CALL RWRITE(DRMCT1,BUFF(1),NOWRDS,ISTA(K,1))
DRMCT1 = DRMCT1 + NOWRDS
GO TO 200
190 CALL RWRITE(DRMCR1,BUFF(1),NOWRDS,ISTA(K,1))
DRMCR1 = DRMCR1 + NOWRDS
C
200 CONTINUE

```

ST000010  
ST000020  
ST000030  
ST000040  
ST000050  
ST000060  
ST000070  
ST000080  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00060  
  
ST000100  
ST000110  
ST000120  
ST000130  
ST000140  
ST000150  
ST000160  
ST000170  
ST000180  
ST000190  
ST000200  
ST000210  
ST000220  
ST000230  
ST000240  
ST000250  
ST000260  
ST000270  
ST000280  
ST000290  
ST000300  
ST000310  
ST000320  
ST000330  
ST000340  
ST000350  
ST000360  
ST000370  
ST000380  
ST000390  
ST000400  
ST000410  
ST000420  
ST000430  
ST000440  
ST000450  
ST000460  
ST000470  
ST000480  
ST000490  
ST000500  
ST000510  
ST000520  
ST000530  
ST000540  
ST000550  
ST000560  
ST000570  
ST000580  
ST000590  
ST000600  
ST000610  
ST000620  
ST000630  
ST000640  
ST000650  
ST000660

FILE: STOFIL

CALL FSBSFL (NMSTUN,1,ISTAT1)  
RETURN  
END

ST000670  
ST000680  
ST000690

ORIGINAL PAGE IS  
OF POOR QUALITY

~~16-27~~  
318



FILE: UNPCKV

```

C      SURROUTINE UNPCKV(PLOT, TNSDAT, NVECT)
C      UNPCKV UNPACKS TWO 8 BITS BYTES AND STORES THE VALUES INTO TWO
C      FLOATING POINT WORDS
C      IMPLICIT INTEGER (A-Z)
C      REAL TNSDAT(2, NVECT)
C
C      INCLUDE CMHKL2.LIST
C      COMMON /SCATTER/ RSCALE, XYSCLF, CLRVEC(30), NCLRCH, CLRKEY, LOG,
C      * FREQ, XMAX, YMAX, XMIN, YMIN, RCKGND, XHI, XLO, YLO, XSIZ,
C      * YHI, YSIZ, NRINS, SYMMTX(32), RMATX(60), HVEC(30), NRVCHN, NOFEAT
C      * SCALKY, MENADR, FLADW, PNTADR, IDADR, NC, BMFEAT, BCOMR
C      * NOVEC, TOTYNS, SIZE, DRMID, DRMID1, DRMCLR, DRMCR1, DRMTNS, DRMTN1,
C      * DRMCNT, DRMCT1, DRMVEC, DRMVC1, VECTR1, DATA1, NVEC, NOREAD, LREAD
C      * DRMPTR, DRMPT1, FETVEC(16), DRMPLT, CSCALE
C      * NOSUB
C
C      DIMENSION PLOT(1)
C      LOGICAL *1 LDUM(4), LLDUM(4)
C      EQUIVALENCE (LDUM(1), IDUM), (LLDUM(1), IIDUM)
C
C      DO 100 I=1, NVECT
C      IDUM=PLOT(I)
C      IIDUM=0
C      LLDUM(4)=LDUM(2)
C      VALUE2=IIDUM
C      LLDUM(4)=LDUM(1)
C      VALUE1=IIDUM
100  TNSDAT(2, I) = VALUE2
      TNSDAT(1, I) = VALUE1
      RETURN
      END
UNP00010
UNP00020
UNP00030
UNP00040
UNP00050
UNP00060
UNP00070
UNP00080
UNP00090
UNP00100
UNP00110
UNP00120
UNP00130
UNP00140
UNP00150
UNP00160
UNP00170
UNP00180
UNP00190
UNP00200
UNP00210
UNP00220
UNP00230
UNP00240
UNP00250
UNP00260
UNP00270
UNP00280
UNP00290
UNP00300
UNP00310
UNP00320
UNP00330
UNP00340
UNP00350
UNP00360
```

ORIGINAL PAGE IS  
OF POOR QUALITY



FILE: VECSCN

```
FUNCTION VECSCN(VECTR,NVCELT,CARD,COL)                                VEC00010
C* VECSCN CONVERTS ALPHA CHARACTERS TO INTEGERS                       VEC00020
C* RETURNS THE NO. OF ELEMENTS WITHIN A SET OF PARENTHESIS          VEC00030
C* RETURNS THE NO. OF PARENTHESIS                                    VEC00040
C* IMPLICIT INTEGER (A-Z)                                           VEC00050
C                                                                      VEC00060
C DIMENSION VECTR(1),COMMA(2),CARD(1)                                VEC00070
C                                                                      VEC00080
C DATA STAR /'.'/,BLANK/'.'/,LPBCD/'(.'/,RPRCD/'('.'/,KOMMA/','.'/ VEC00090
C DATA COMMA/','.'/,KOMMA/','.'/                                  VEC00100
C                                                                      VEC00110
C NTIMES = 0                                                         VEC00120
C TOTNUM = 0                                                         VEC00130
C                                                                      VEC00140
C 80 TOTNUM = TOTNUM + 1                                             VEC00150
C NVCELT = NVCELT + 1                                              VEC00160
C 100 M = NATCHR(CARD,COL)                                          VEC00170
C                                                                      VEC00180
C IF (M.EQ.BLANK) GO TO 140                                         VEC00190
C IF (M.EQ.LPBCD) GO TO 100                                         VEC00200
C IF (M.EQ.RPRCD) GO TO 130                                         VEC00210
C IF (M.EQ.KOMMA) GO TO 120                                         VEC00220
C IF (M.EQ.STAR) GO TO 135                                          VEC00230
C                                                                      VEC00240
C CHANGING NUMBER FROM ALPHA MODE TO INTEGER MODE                 VEC00250
C                                                                      VEC00260
C 110 CALL I4A1RN(CARD(COL),1,NUM)                                   VEC00270
C NUMR = 10*NUMR + NUM                                              VEC00280
C IF (NUM.LT.0 .OR. NUM.GT.9) GO TO 150                             VEC00290
C GO TO 100                                                         VEC00300
C                                                                      VEC00310
C FOUND A COMMA                                                    VEC00320
C                                                                      VEC00330
C 120 VECTR(TOTNUM) = NUMR                                          VEC00340
C NUMR = 0                                                         VEC00350
C GO TO 80                                                         VEC00360
C                                                                      VEC00370
C FOUND A ')'                                                      VEC00380
C                                                                      VEC00390
C 130 VECTR(TOTNUM) = NUMR                                          VEC00400
C NUMR = 0                                                         VEC00410
C IF (NTIMES.EQ.0) GO TO 133                                        VEC00420
C DO 132 I=1,NTIMES                                               VEC00430
C DO 132 J=1,NVCELT                                               VEC00440
C JJ = TOTNUM + (I-1)*NVCELT + J                                  VEC00450
C 132 VECTR(JJ) = VECTR(TOTNUM-NVCELT+J)                          VEC00460
C TOTNUM = TOTNUM + NVCELT                                         VEC00470
C NTIMES = 0                                                       VEC00480
C                                                                      VEC00490
C* 133 J = FIND12(CARD,COL,COMMA)                                   VEC00500
C IF (J.EQ.-1) GO TO 140                                           VEC00510
C NVCELT = 0                                                       VEC00520
C GO TO 80                                                         VEC00530
C                                                                      VEC00540
C FOUND A MULTIPLICATIVE FACTOR                                    VEC00550
C                                                                      VEC00560
C 135 NTIMES = NUMR - 1                                            VEC00570
C NUMR = 0                                                         VEC00580
C GO TO 100                                                         VEC00590
C                                                                      VEC00600
C FINISHED SCANNING CARD                                          VEC00610
C                                                                      VEC00620
C 140 VECSCN = TOTNUM/NVCELT                                       VEC00630
C RETURN                                                           VEC00640
C                                                                      VEC00650
C 150 VECSCN = -1                                                  VEC00660
C WRITE(A,200)                                                     VEC00670
C 200 FORMAT(' ERROR OCCURRED SCANNING VECTOR CARD' )           VEC00680
C RETURN                                                           VEC00690
C END                                                             VEC00700
C                                                                      VEC00710
C                                                                      VEC00720
C                                                                      VEC00730
```

## 17. DOTDATA PROCESSOR

FILE: DOTDAT

```
C   DOTDAT IS THE DRIVER ROUTINE FOR THE DOTDATA PROCESSOR
C   SURROUTINE DOTDAT(ARRAY, TOP)
C   DIMENSION ARRAY(1)
    CALL SFT13
    CALL DOT5(ARRAY(1), ARRAY(5001), ARRAY(6001), TOP)
    RETURN
    END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: DOTS

```
C
C
C   DOTS IS THE CO-ORDINATOR FOR CREATING THE DOT DATA FILE
C
C   SURROUTINE DOTS(DATA, FIELDS, VERTEX, TOP)
C   IMPLICIT INTEGER (A-Z)
C   DIMENSION DATA(SIZE, 1), IDATA(10000)
C   DIMENSION FIELDS(4, 1), FL(12), VERTEX(1)
C   INCLUDE COMBK1.LIST
C   INCLUDE COMBK4.LIST
C   INCLUDE COMBK14.LIST
C   INCLUDE COMBK16.LIST
C
C   COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARS22, TOTVT2, NOFLD2,
C   *   AVAR2, COVAR2, CLSID2, SURNO2, SURDS2, FLDSV2, VERTX2,
C   *   FETVC2(30), SURVC2(75), SURPTH(75), CLSVC2(60),
C   *   KEPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),
C   *   GRPCHK(61), GROUPS(124)
C   COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
C   *   HISFIL, HISKEY, TRFORM, ERPTP, ERPKY, MAPUNT, NOFILE,
C   *   DRUMAD, DRMWD, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
C   *   NHSTUN, NHSTFI, SCTRUN, MAPFIL
C   *   DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
C   *   CRDUNT, PRTUNT, HANDIO
C   COMMON /DOTVEC/TYPE, CATNAM(60), NOCAT, TOTVEC, FLDINF(6), PRTKEY
C   *   SIZE, LACIE
C   COMMON/ISOLNK/SUNANG(8), ISUNT, ISUNC, SMSTR, SMSTP, SMINC, LINSKP
CSEND
C
C   DATA BLANK/ ' '
C   NOSUN = 8
C   ISUNT = 1
C   ISUNC = 0
C   STAMNT = 1
C   SWITCH = 0
C   IPT = 1
C   IF(LACIE.NE.0)IPT=0
C
C*** CODE ADDED NOV 21, 1978 TO SUPPORT LIST PROCESSING
C
C   INIT = 0
C   ZERO = 0
C   SWCHG = 0
C   NOCAT = 0
C   NOFLD2=0
C   TYPE = 1
C
C   INITIALIZE IMAGE DATA TAPE
C
C   90 CALL TAPHDR(DATAPE, DATFIL)
C
C   DO A0 I=1,SIZE
C   DO A0 J=1,TOTVEC
C   A0 DATA(I,J) = 0
C   TOTVT2 = 0
C   TOTVEC = 0
C   A5 IF(LACIE.EQ.)CALL FLDLAC(FIELDS, STAMNT, &100, &510, &520, IPT,
C   * VERTEX)
C
C*** ADDED NOV 21, 1978 IN SUPPORT OF LIST PROCESSING
C
C   IF(LACIE.GT.1) CALL LISTLC(FIELDS, STAMNT, &100, &510,
C   * &520, SWCHG, INIT, LACIE, ZERO, IPT, VERTEX)
C   CALL FLDTYP(FIELDS, STAMNT, &100, &510, &520, IPT, VERTEX)
C
C   100 LINSTR = FLDINF(1)
C   LINEND = FLDINF(2)
C   LININC = FLDINF(3)
C   SAMSTR = FLDINF(4)
C   SAMEND = FLDINF(5)
C   SAMINC = FLDINF(6)
C   FIELDS(2, NOFLD2) = NOCAT
C   TOTVT2 = FIELDS(4, NOFLD2) + TOTVT2
C
C   ILINE = (LINEND-LINSTR)/LININC + 1
C   NSAMP = (SAMEND-SAMSTR)/SAMINC + 1
C
C   POSITION IMAGE TAPE FOR THIS FIELD
C   CALL FLDINT(FLDINF(1), FETVC2, NOFET2)
C
C   DOT00010
C   DOT00020
C   DOT00030
C   DOT00040
C   DOT00050
C   DOT00060
C   DOT00070
C   DOT00080
C   DOT00090
C   DOT00100
C   DOT00110
C   DOT00120
C   DOT00130
C   DOT00140
C   DOT00150
C   DOT00160
C   DOT00170
C   DOT00180
C   DOT00190
C   DOT00200
C   DOT00210
C   DOT00220
C   DOT00230
C   DOT00240
C   DOT00250
C   DOT00260
C   DOT00270
C   DOT00280
C   DOT00290
C   DOT00300
C   DOT00310
C   DOT00320
C   DOT00330
C   DOT00340
C   DOT00350
C   DOT00360
C   DOT00370
C   DOT00380
C   DOT00390
C   DOT00400
C   DOT00410
C   DOT00420
C   DOT00430
C   DOT00440
C   DOT00450
C   DOT00460
C   DOT00470
C   DOT00480
C   DOT00490
C   DOT00500
C   DOT00510
C   DOT00520
C   DOT00530
C   DOT00540
C   DOT00550
C   DOT00560
C   DOT00570
C   DOT00580
C   DOT00590
C   DOT00600
C   DOT00610
C   DOT00620
C   DOT00630
C   DOT00640
C   DOT00650
C   DOT00660
C   DOT00670
C   DOT00680
C   DOT00690
C   DOT00700
C   DOT00710
C   DOT00720
C   DOT00730
C   DOT00740
C   DOT00750
C   DOT00760
C   DOT00770
C   DOT00780
C   DOT00790
```



FILE: DOTS

```
1      30X,'DATA'//
1      ISTART=1
1      IEND=10
799    CONTINUE
1      IKT=0
1      DO 800 II=1,TOTVEC
1      IKT=IKT+1
1      IF (IEND.GT.NOFET2) IEND=NOFET2
1      IF (II.NE.1.AND.IKT.EQ.1) WRITE(6,810)
810    FORMAT(1H,5(/))
1      IF (IKT.NE.1) GO TO 820
1      WRITE(A,700)
1      WRITE(A,690)
720    FORMAT(37X,10(A1,'CH(',I2,')'))
820    CONTINUE
1      WRITE(A,710) II,(DATA(I,II),I=1,4),(DATA(4+JJ,II),JJ=ISTART,IEND)
710    FORMAT(1X,I3,1H.,3X,I4,3X,I4,2X,I2,6X,I2,8X,10(13,4X))
1      WRITE(6,712)
712    FORMAT( )
800    CONTINUE
1      IF (NOFFT2.GT.10) GO TO 830
1      GO TO 840
830    CONTINUE
1      IF (ITWO.EQ.1) GO TO 840
1      ITWO=1
1      ISTART=IEND+ISTART
1      IEND=NOFET2
1      GO TO 799
840    CONTINUE
1      IF (SWTCH.EQ.1) GO TO 90
1      RETURN
1      END
```

DOT01590  
DOT01600  
DOT01610  
DOT01620  
DOT01630  
DOT01640  
DOT01650  
DOT01660  
DOT01670  
DOT01680  
DOT01690  
DOT01700  
DOT01710  
DOT01720  
DOT01730  
DOT01740  
DOT01750  
DOT01760  
DOT01770  
DOT01780  
DOT01790  
DOT01800  
DOT01810  
DOT01820  
DOT01830  
DOT01840  
DOT01850  
DOT01860  
DOT01870  
DOT01880  
DOT01890  
DOT01900  
DOT01910

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: FLDLAC

```

C          FIELDS - CATEGORY NAME AND DOT TYPE FOR DOT 1 STORED IN          FLD00010
C          FIELD(1,1) AND FIELD(4,1)                                       FLD00020
C          STAMNT - INITIALLY SET TO 1. SWITCHED TO INDICATE DOTS BEING     FLD00030
C          TAKEN FROM CURRENTLY READ CARD.                                  FLD00040
C          IPT - INITIALLY SET TO 1. INDEX NUMBER FOR FIELD VERTEX INFORMATION FLD00050
C          VERTEX - VERTEX INFORMATION FOR EACH DOT.                         FLD00060
C                                                                              FLD00070
C          SUBROUTINE FLDLAC(FIELDS,STAMNT,*,*,*,IPT,VERTEX)                 FLD00080
C          IMPLICIT INTEGER (A-Z)                                          FLD00090
C          LOGICAL*1 LCARD(300),LCATNM(4)                                   FLD00100
C          REAL DIM                                                         FLD00110
C          DIMENSION FIELDS(4,1),VERTEX(1),CARD(75),NDOTS(30)             FLD00120
C          DIMENSION ACARD(80)                                             FLD00130
C          LOGICAL SWITCH                                                  FLD00140
C          DATA SWITCH/.TRUE./,SWCHG/0/,ENDBCD/'SEN'/,                   FLD00150
C          *CATNM/' '/,                                                    FLD00160
C          INCLUDE CMHK14                                                  FLD00170
C          INCLUDE CMHK1                                                  FLD00180
C          COMMON/INFORM/NOCL52,NOSUM2,NOFET2,VARSZ2,TOTVT2,NOFLD2,        FLD00190
C          AVAR2,CVAR2,CLS102,SUBNO2,SURDS2,FLOSZ2,VERTX2,                FLD00200
C          FETVC2(30),SURVC2(75),SUMPTR(75),CLSYC2(60),                  FLD00210
C          KFPPTS(60),NOGWP,GWPNAM(60),GWPDEX(61),                       FLD00220
C          GRPCHK(6),GROUPS(124)                                          FLD00230
C          COMMON /DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY   FLD00240
C          ,SIZE ,LACIE                                                  FLD00250
C                                                                              FLD00260
CSEND  EQUIVALENCE (LCATNM(1),CATNM),(LCARD(1),CARD(1))                  FLD00270
C          IF (STAMNT.EQ.2)GO TO 30                                       FLD00280
C          IF (.NOT.SWITCH)GO TO 20                                       FLD00290
C          CALL WFWF41(30,80)                                             FLD00300
C          READ(21,103)(ACARD(I),I=1,80)                                  FLD00310
C          103  FORMAT(A0A1)                                              FLD00320
C          WRITE(30,103)(ACARD(I),I=1,80)                                FLD00330
C          REWIND 30                                                       FLD00340
C          READ(30,1000)ID,TYPES,CARD                                     FLD00350
C          REWIND 30                                                       FLD00360
C          1000 FORMAT(A3,1X,I1,75A1)                                     FLD00370
C          IF (TYPE.EQ.TYPES)GO TO 20                                     FLD00380
C          IF (SWCHG.NE.0)GO TO 40                                       FLD00390
C          TYPE = TYPES                                                  FLD00400
C                                                                              FLD00410
C          READ CARD                                                       FLD00420
C                                                                              FLD00430
C          20  COL = 0                                                    FLD00440
C          CATNM = NATCHW(CARD,COL)                                       FLD00450
C          IF NEXT CHAR IS NOT A CAT. NAME, CORRECT COL COUNT TO READ NUM FLD00460
C          IF (CATNM.GT.0)GO TO 21                                       FLD00470
C          LINDEX=4*COL+1                                                FLD00480
C          LCATNM(2)=LCARD(LINDEX)                                       FLD00490
C          COL=COL+1                                                       FLD00500
C          IF (CATNM.EQ.CATNM)GO TO 23                                     FLD00510
C          NOCAT=NOCAT + 1                                               FLD00520
C          CATNAM(NOCAT)=CATNM                                           FLD00530
C          CATNM = CATNM                                                  FLD00540
C          GO TO 23                                                       FLD00550
C          21  COL=COL - 1                                               FLD00560
C          23  NDCARD=0                                                  FLD00570
C          CALL NUMRD(NDOTS,NDCARD,CARD,COL)                              FLD00580
C          IF (NDCARD.EQ.0)GO TO 10                                       FLD00590
C          ICNT = 0                                                       FLD00600
C          STAMNT = ?                                                     FLD00610
C          SWITCH = .TRUE.                                               FLD00620
C          GO TO 100                                                       FLD00630
C                                                                              FLD00640
C          TEST FOR END OF DOTS TO BE PROCESSED ON CARD                 FLD00650
C          30  IF (ICNT.LT.NDCARD)GO TO 100                               FLD00660
C          READ NEXT CARD                                                FLD00670
C                                                                              FLD00680
C          STAMNT = 1                                                    FLD00690
C          ICNT = 0                                                       FLD00700
C          READ(21,103)(ACARD(I),I=1,80)                                  FLD00710
C          WRITE(30,103)(ACARD(I),I=1,80)                                FLD00720
C          REWIND 30                                                       FLD00730
C          READ(30,1000)ID,TYPES,CARD                                     FLD00740
C          REWIND 30                                                       FLD00750
C          IF (ID.EQ.FNDHCD)RETURN 3                                       FLD00760
C          IF (TYPE.EQ.TYPES)GO TO 20                                       FLD00770
C          FLD00780
C          FLD00790

```

FILE: FLDLAC

```
SWITCH = .FALSE.  
SWCHG = SWCHG + 1  
IF (SWCHG.GT.1) GO TO 40  
TYPE = TYPES  
IPT = 0  
C***** CHANGED) JUNE 28 1978  
RETURN 2  
C  
C 100 ICNT = ICNT + 1  
NOFLD2 = NOFLD2 + 1  
C  
C COMPUTE LINE INCREMENT  
NN = NDOTS(ICNT)  
NI = IABS(NN) / 10000000  
LI = IABS(NN) - NI * 10000000  
IF (LI.GE.10000000) NI = NI + 1  
C  
C COMPUTE SAMPLE INCREMENT  
KK=1  
IF (NN.LT.0) KK=-1  
LI = NI * KK  
N2 = NN - LI + 10000000  
N3 = IABS(N2) / 10000  
SI = IABS(N2) - N3 * 10000  
IF (SI.GE.1000) N3 = N3 + 1  
KK=1  
IF (N2.LT.0) KK=-1  
SI = N3 * KK  
LACI = N2 - SI * 10000  
LR = (LACI - 1) / 19  
LP = (LR + 1) * 10  
LS = LR - 1  
LS = LS / 10  
LS = 10 * (LACI - (LS * 19))  
L = LR - LI  
S = LS + SI  
C  
C STORE DOT INFO  
FIELDS(1,NOFLD2) = CATNM  
FIELDS(4,NOFLD2) = 2  
FLDINF(1) = L  
FLDINF(2) = L  
FLDINF(3) = L  
FLDINF(4) = S  
FLDINF(5) = S  
FLDINF(6) = 1  
IF (IPT.NE.0) GO TO 35  
IPT = -3  
C 35 IPT = IPT + 4  
VERTEX(IPT) = S  
VERTEX(IPT+1) = L  
VERTEX(IPT+2) = S  
VERTEX(IPT+3) = L  
RETURN 1  
C 40 WRITE(4,2000)  
2000 FORMAT(//4X,'ERROR HAS OCCURRED IN READING LACIE FORMATTED DOT CAR  
DS - SUBROUTINE FLDLAC - EXIT TAKEN')  
RETURN 3  
END
```

FLD000800  
FLD000810  
FLD000820  
FLD000830  
FLD000840  
FLD000850  
FLD000860  
FLD000870  
FLD000880  
FLD000890  
FLD000900  
FLD000910  
FLD000920  
FLD000930  
FLD000940  
FLD000950  
FLD000960  
FLD000970  
FLD000980  
FLD000990  
FLD001000  
FLD001010  
FLD001020  
FLD001030  
FLD001040  
FLD001050  
FLD001060  
FLD001070  
FLD001080  
FLD001090  
FLD001100  
FLD001110  
FLD001120  
FLD001130  
FLD001140  
FLD001150  
FLD001160  
FLD001170  
FLD001180  
FLD001190  
FLD001200  
FLD001210  
FLD001220  
FLD001230  
FLD001240  
FLD001250  
FLD001260  
FLD001270  
FLD001280  
FLD001290  
FLD001300  
FLD001310  
FLD001320  
FLD001330  
FLD001340  
FLD001350  
FLD001360  
FLD001370  
FLD001380  
FLD001390  
FLD001400  
FLD001410  
FLD001420

FILE: FLDTYP

```

SUBROUTINE FLDTYP(FIELDS,STAMNT,*,*,*,IPT,VERTEX)
  IMPLICIT INTEGER (A-Z)
  INCLUDE COMRK1.LIST
  INCLUDE COMRK1A.LIST
  COMMON/INFORM/NOCLS2,NOSURP,NOFET2,VARSTP,TOTVTP,NOFLD2,
  AVAR2,COVAR2,CLSID2,SUNO2,SUBO2,FLOS2,VERTX2,
  FETVC2(30),SURVC2(75),SUBTR(75),CLSYC2(60),
  KFPPTS(60),NOGRP,GRPNAM(40),GRPDEX(41),
  GRPCHR(41),GROUPS(124)
  COMMON /DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),PRKEY
  * ,SIZE,LACIE
  DIMENSION FIELDS(4,1),VERTEX(1)
  DIMENSION CARD(62)
  LOGICAL SWITCH
  DATA SWITCH,TRUE, /
  IPT = IPT + FIELDS(4,NOFLD2)*2
  GO TO 100,100,STAMNT
80  I = LAREAD(FIELDS(1,NOFLD2+1),VERTEX(IPT),FLDINF(1),FIELDS(4,NOFLD2
  * 2,1))
  WAS CLASS,SUBCLASS,FIELD, OR SEND* ENCOUNTERED
  IF (I) :FO: (-4) GO TO 90
  IF (I) :FO: (-1) GO TO 120
  IF (I) :FO: (1) GO TO 130
  IF (I) :FO: (0) GO TO 140
  TYPE CARD
  *0 IF (SWITCH) GO TO 100
  STAMNT = 2
  RETURN 2
100 READ(30,115)CARD
115 FORMAT(10X,62A1)
  REWIND 30
  COL = 0
  ZERO = 0
  ME = NUMBER(CARD,COL,TYPE,ZERO)
  IPT = 1
  NOFLD2 = 0
  NOCAT = 0
  SWITCH = .FALSE.
  GO TO 80
  CLASSNAME CARD
110 FORMAT(10X,A4)
120 NOCAT = NOCAT + 1
  READ(30,110)CATNAM(NOCAT)
  REWIND 30
  GO TO 80
  FIELD CARD
130 NOFLD2 = NOFLD2 + 1
  STAMNT = 1
  RETURN 1
  SEND*
140 SWITCH = .TRUE.
  RETURN 3
  END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

FLD0000610  
 FLD0000620  
 FLD0000630  
 FLD0000640  
 FLD0000650  
 FLD0000660  
 FLD0000670  
 FLD0000680  
 FLD0000690  
 FLD0000700  
 FLD0000710  
 FLD0000720  
 FLD0000730  
 FLD0000740  
 FLD0000750  
 FLD0000760  
 FLD0000770  
 FLD0000780  
 FLD0000790  
 FLD0000800  
 FLD0000810  
 FLD0000820  
 FLD0000830  
 FLD0000840  
 FLD0000850  
 FLD0000860  
 FLD0000870  
 FLD0000880  
 FLD0000890  
 FLD0000900  
 FLD0000910  
 FLD0000920  
 FLD0000930  
 FLD0000940  
 FLD0000950  
 FLD0000960  
 FLD0000970  
 FLD0000980  
 FLD0000990  
 FLD0001000  
 FLD0001010  
 FLD0001020  
 FLD0001030  
 FLD0001040  
 FLD0001050  
 FLD0001060  
 FLD0001070  
 FLD0001080  
 FLD0001090  
 FLD0001100  
 FLD0001110  
 FLD0001120  
 FLD0001130  
 FLD0001140  
 FLD0001150  
 FLD0001160  
 FLD0001170  
 FLD0001180  
 FLD0001190  
 FLD0001200  
 FLD0001210  
 FLD0001220  
 FLD0001230  
 FLD0001240  
 FLD0001250  
 FLD0001260  
 FLD0001270  
 FLD0001280  
 FLD0001290  
 FLD0001300  
 FLD0001310  
 FLD0001320  
 FLD0001330  
 FLD0001340  
 FLD0001350  
 FLD0001360  
 FLD0001370  
 FLD0001380  
 FLD0001390  
 FLD0001400  
 FLD0001410  
 FLD0001420  
 FLD0001430  
 FLD0001440  
 FLD0001450  
 FLD0001460  
 FLD0001470  
 FLD0001480  
 FLD0001490  
 FLD0001500  
 FLD0001510  
 FLD0001520  
 FLD0001530  
 FLD0001540  
 FLD0001550  
 FLD0001560  
 FLD0001570  
 FLD0001580  
 FLD0001590  
 FLD0001600  
 FLD0001610  
 FLD0001620  
 FLD0001630  
 FLD0001640  
 FLD0001650  
 FLD0001660  
 FLD0001670



```

SUBROUTINE SFT13
IMPLICIT INTEGER (A-Z)
DIMENSION CODE(10),CARD(62),EQUOM(3),ACARD(20)
DIMENSION SLASH(2)
DATA SLASH /1,1//
DATA CODE /'CHAN', 'DATA', 'DOTF',
* 'OPT1', 'DATE', 'COMM', 'MED1', 'MED2', 'END' /
DATA EQUOM /2,2,2//
DATA D /'D', 'BLNK' /
DATA L /'L' /
INCLUDE COMAK1.LIST
INCLUDE COMAK4.LIST
INCLUDE COMAK6.LIST
INCLUDE CMK14.LIST
COMMON /INFORM /NOCL52, NOSUR2, NOFET2, VARS72, TOTVT2, NOFLO2,
* AVAR2, COVAR2, CLS102, SUMNO2, SIMOS2, FLDSV2, VERTX2,
* FETVC2(30), SURVC2(75), SURPTR(75), CLSYC2(60),
* KFPTS(40), NOGRP, GRPNAM(60), GRPDEX(61),
* GRPCHK(4), GHOUPS(124)
DIMENSION HED(15), HED2(15), DATE(3), COMENT(15)
EQUIVALENCE (HED(1), HED(4)), (DATE(1), HED(22)),
* (HED2(1), HED(30)), (COMENT(1), HED(48))
COMMON /GLOBAL /HEAD(63), MAPTAP, DATE, SAVTAP, RMFILE, RMKEY,
* HISFIL, HISKEY, TRFORM, ERPTP, ERPKEY, MAPUNT, NOFILE,
* DRUMAD, DRUMDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
* NHSTUN, NHSTFI, SCTRUN, MAPFIL
* DOTUNT, DOTFIL, NCHPAS, TRANSF, BMTFFL, HISTFL, PCHUNT,
* PRUNT, PRUNT, RANDIO
COMMON /DOTVEC /TYPE, CATNAM(60), NOCAT, TOTVEC, FLDINF(6), PRTKEY
* SIZE, LACIE

```

CSEND

```

ZERO = 0
NOFET2 = 0
FIELD = 1
PRTKEY = 0
NPUNT = 9
LACIE = 0

```

C

```

100 WRITE(6,100)
FORMAT(/1X,'INPUT SUMMARY'//)
SET UP REREAD BUFFER
RRUNIT = 30
CALL REREAD(RRUNIT,80)

```

C

PUT CARD IN BUFFER

```

105 READ(2,103) (ACARD(I),I=1,20)
103 FORMAT(20A4)
WRITE(30,107) (ACARD(I),I=1,20)
REWIND RRUNIT

```

C

```

READ(30,110) CODE1,CARD
REWIND RRUNIT
COL = 0
WRITE(4,120) CODE1,CARD
170 FORMAT(1X,A4,6X,62A1)
110 FORMAT(A4,6X,62A1)
DO 130 I=1,NPUNT
IF (CODE1.EQ.CODE(I)) GO TO (150,180,210,330,370,
* 390,400,410,420),I
130 CONTINUE
120 WRITE(4,140)
140 FORMAT(' INVALID CONTROL CARD - IGNORED ')
GO TO 105

```

C

CHANNEL CARD

```

150 M = NKTCHW(CARD,COL)
IF (M.EQ.0) GO TO 155
IF (M.FU,BLNK) GO TO 105
152 WRITE(4,153)
153 FORMAT(' ERMUR ON DATA CARD')
GO TO 105
154 J = FIND2(CARD,COL,EQUOM)
IF (J.NE.?) GO TO 152
NOFET2 = NUMBER(CARD,COL,FETVC2,NOFET2)

```

T00010  
T00020  
T00030  
T00040  
T00050  
T00060  
T00070  
T00080  
T00090  
T00100  
T00110  
T00120  
T00130  
T00140  
T00150  
T00160  
T00170  
T00180  
T00190  
T00200  
T00210  
T00220  
T00230  
T00240  
T00250  
T00260  
T00270  
T00280  
T00290  
T00300  
T00310  
T00320  
T00330  
T00340  
T00350  
T00360  
T00370  
T00380  
T00390  
T00400  
T00410  
T00420  
T00430  
T00440  
T00450  
T00460  
T00470  
T00480  
T00490  
T00500  
T00510  
T00520  
T00530  
T00540  
T00550  
T00560  
T00570  
T00580  
T00590  
T00600  
T00610  
T00620  
T00630  
T00640  
T00650  
T00660  
T00670  
T00680  
T00690  
T00700  
T00710  
T00720  
T00730  
T00740  
T00750  
T00760  
T00770  
T00780  
T00790

FILE: SET13

```

CALL ORDER(FETVC2,NOFET2)
GO TO 105
C
C
C
DATA FILE CARD
180 M = NXTCHR(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
IF (M.EQ.U) GO TO 190
IF (M.EQ.FF) GO TO 200
185 WRITE(6,187)
187 FORMAT(' ERROR ON DATA FILE CARD')
GO TO 105
190 J = FIND12(CARD,COL,EQUCOM)
IF (J.NE.2) GO TO 185
M = NUMBER(CARD,COL,DATAPE,ZERO)
COL = COL
GO TO 180
200 J = FIND12(CARD,COL,EQUCOM)
IF (J.NE.2) GO TO 185
M = NUMBER(CARD,COL,DATFIL,ZERO)
DATFIL = DATFIL - 1
COL = COL - 1
GO TO 180

```

```

C
C
C
DOT FILE CARD
210 M = NXTCHR(CARD,COL)
IF (M.FO.OO) GO TO 213
IF (M.FO.BLNK) GO TO 105
GO TO 215
213 J = FIND12(CARD,COL,SLASH)
IF (J.EQ.-1) GO TO 215
214 M = NXTCHR(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
IF (M.EQ.U) GO TO 230
IF (M.EQ.FF) GO TO 240
215 WRITE(6,220)
220 FORMAT(' ERROR ON DOT FILE CARD')
GO TO 105
230 J = FIND12(CARD,COL,EQUCOM)
IF (J.NE.2) GO TO 215
M = NUMBER(CARD,COL,DOTUNT,ZERO)
COL = COL - 1
GO TO 214
240 J = FIND12(CARD,COL,EQUCOM)
IF (J.NE.2) GO TO 215
M = NUMBER(CARD,COL,DOTFIL,ZERO)
DOTFIL = DOTFIL - 1
COL = COL - 1
GO TO 214

```

```

C
C
C
OPTION CARD
330 M = NXTCHR(CARD,COL)
IF (M.FO.BLNK) GO TO 105
IF (M.FO.P) GO TO 340
IF (M.EQ.L) GO TO 345

```

```

C
C
C
*** CODE ADDED NOV 21, 1978 IN SUPPORT OF LIST PROCESSING
IF (M.EQ.U) GO TO 350

```

```

C
C
C
333 WRITE(6,335)
335 FORMAT(' ERROR ON OPTION CARD')
GO TO 105
340 PRKEY = 1
GO TO 105
345 LACIE = 1
GO TO 105
350 M = NUMBER(CARD,COL,LACIE,ZERO)
GO TO 105

```

```

C
C
C
DATE CARD
370 M = NXTCHR(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
READ(30,380)DATE
380 FORMAT(10X,15A4)
REWIND BRUNIT

```

SET00800  
SET00810  
SET00820  
SET00830  
SET00840  
SET00850  
SET00860  
SET00870  
SET00880  
SET00890  
SET00900  
SET00910  
SET00920  
SET00930  
SET00940  
SET00950  
SET00960  
SET00970  
SET00980  
SET00990  
SET01000  
SET01010  
SET01020  
SET01030  
SET01040  
SET01050  
SET01060  
SET01070  
SET01080  
SET01090  
SET01100  
SET01110  
SET01120  
SET01130  
SET01140  
SET01150  
SET01160  
SET01170  
SET01180  
SET01190  
SET01200  
SET01210  
SET01220  
SET01230  
SET01240  
SET01250  
SET01260  
SET01270  
SET01280  
SET01290  
SET01300  
SET01310  
SET01320  
SET01330  
SET01340  
SET01350  
SET01360  
SET01370  
SET01380  
SET01390  
SET01400  
SET01410  
SET01420  
SET01430  
SET01440  
SET01450  
SET01460  
SET01470  
SET01480  
SET01490  
SET01500  
SET01510  
SET01520  
SET01530  
SET01540  
SET01550  
SET01560  
SET01570  
SET01580

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: SET13

C	GO TO 105	SET01590
C	COMMENT CARD	SET01600
C	390 M = NXTCHR(CARD,COL)	SET01610
	IF (M.EQ. BLNK ) GO TO 105	SET01620
	READ(30,380)COMENT	SET01630
	REWIND RRUNIT	SET01640
	GO TO 105	SET01650
C	HED1	SET01660
C	400 M = NXTCHR(CARD,COL)	SET01670
	READ(30,380) HED1	SET01680
	REWIND RRUNIT	SET01690
	GO TO 105	SET01700
C	HED2	SET01710
C	410 M = NXTCHR(CARD,COL)	SET01720
	READ(30,380) HED2	SET01730
	REWIND RRUNIT	SET01740
	GO TO 105	SET01750
C	*END*	SET01760
C	420 CONTINUE	SET01770
	IF (NOFET2.NE. 0) GO TO 440	SET01780
	DO 430 I=1,30	SET01790
	FETVC2(I) = I	SET01800
	430 CONTINUE	SET01810
	NOFET2 = I	SET01820
C	440 SIZE = 4 + NOFET2	SET01830
C	WRITE(6,1000)	SET01840
	IF (NOFET2.NE. 0) WRITE(6,1010) (FETVC2(I),I=1,NOFET2)	SET01850
	IF (PRTKEY.EQ. 1) WRITE(6,1030)	SET01860
	1040 FORMAT(' LACIE FORMATTED DOT CARDS USED AS EOD-LARSYS FIELD CARDS'	SET01870
	IF(LACIE.EQ.1)WRITE(6,1040)	SET01880
	1000 FORMAT('///' USER HAS REQUESTED THE FOLLOWING OPTIONS ://)	SET01890
	1010 FORMAT(' SELECTED CHANNELS ARE',30I3)	SET01900
	1030 FORMAT(' PRINT DATA VECTORS')	SET01910
C	RETURN	SET01920
C	END	SET01930
		SET01940
		SET01950
		SET01960
		SET01970
		SET01980
		SET01990
		SET02000
		SET02010
		SET02020
		SET02030
		SET02040
		SET02050
		SET02060
		SET02070

~~17-10~~  
331

ORIGINAL PAGE IS  
OF POOR QUALITY

## 18. LABEL PROCESSOR

### FILE LABEL

C	SURROUTINE LABEL (ARRAY, TOP)	LAB00010
	IMPLICIT INTEGER (A-Z)	LAB00020
	DIMENSION FLDSAV(2000), ARRAY(1)	LAB00030
C	GO READ CONTROL CARDS	LAB00040
	CALL SET14 (ARRAY, TOP, EXIT)	LAB00050
C	READ IN REQUIRED FILES	LAB00060
	CALL FILER0 (ARRAY, TOP, NOFLD, TOTVRT, FLDSAV(1), FLDSAV(1001))	LAB00070
C	READY TO PERFORM USER REQUESTS	LAB00080
	CALL LABLR (ARRAY, TOP, NOFLD, TOTVRT, FLDSAV(1), FLDSAV(1001),	LAB00090
	FLDSAV(1), EXIT)	LAB00100
C	FINISHED -- SEND* CARD	LAB00110
	100 READ(21, 100) CARD	LAB00120
	FORMAT(A4)	LAB00130
	RETURN	LAB00140
	END	LAB00150
		LAB00160
		LAB00170
		LAB00180
		LAB00190
		LAB00200
		LAB00210
		LAB00220
		LAB00230
		LAB00240

ORIGINAL PAGE IS  
OF POOR QUALITY



FILE ALLKIN

```

C      FIND ALL DOTS IN CLUSTER I
C      DO 150 J=1,TOTDT2
C      IF (CLSTNO(J).NE. I) GO TO 150
C      K = DOTS(4,J)
C      CATGRY(K) = CATGRY(K) + 1
C      NODOT = NODOT + 1
C      DOTNUM(NODOT) = J
150    CONTINUE
C      WERE DOTS - 1 OF A KIND
C      WERE DOTS- MEXED
C      WERE DOTS-- NULL SET
C      DO 160 J=1,NOCAT
C      IF (CATGRY(J).LT. MAX) GO TO 160
C      MAX = CATGRY(J)
C      CATNUM = J
160    CONTINUE
C      IF (MAX .NE. 0) GO TO 170
C      DEFAULT TO K-NEAREST NEIGHBOR
C      ITER = 1
C      TAB1 = TABADR + (I-1)*TOTDT2
C      SWTCH = 1
C      WRITE(6,165) I
165    FORMAT(//,3X, ' ** DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE FOR CL
C      USTER **',15)
C      CALL KNEAR(DOTS,DUM,DUM,CATVEC,ITER,TAB1,SWTCH,CATNUM,I,MEANS,
C      1 DOTSUM)
C      GO TO 300
C      LABEL BY MAJORITY
170    CONTINUE
C      PRINT CLUSTER INFORMATION
C      TAB1 = TABADR + (I-1)*TOTDT2
C      CALL RREAD(TAB1,DISTNC,TOTDT2,ISTAT)
C      GO TO CODE FOR A TIE
C      GO TO 390
C      WRITE DOT DETAILS
175    WRITE(6,1100) I,CATNAM(CATNUM)
1100   FORMAT(/,5X,12,8X,1A4)
C      DO 110 J=1,NODOT
C      DST=DOTNUM(J)
C      K=DOTS(4,DST)
C      IF (J.EQ.1) WRITE(6,1201) CATNAM(K),DOTVEC(DST),DISTNC(DST).
1201   1 CLSTNO(DST)
C      FORMAT(1H+,T24,1A4,2X,13,4X,F7.2,4X,I2)
C      IF (J.GT.1) WRITE(6,1200) CATNAM(K),DOTVEC(DST),DISTNC(DST).
1200   1 CLSTNO(DST)
C      FORMAT(23X,1A4,2X,13,4X,F7.2,4X,I2)
C      DOTSUM(I,K)=DOTSUM(I,K)+1
110    CONTINUE
C      IF (TIE .EQ. 0) GO TO 300
C      WRITE(6,211)
211    FORMAT(23X, 'A TIE OCCURRED. THE FOLLOWING DOT(S) WERE DISCARDED
C      * ://)
C      DO 213 J=1,TIE
C      INDEX1 = TIES(J)
C      LABELS = DOTS(4,INDEX1)
213    WRITE(6,1200) CATNAM(LABELS),DOTVEC(INDEX1),DISTNC(INDEX1),

```

```

ALL00770
ALL00780
ALL00790
ALL00800
ALL00810
ALL00820
ALL00830
ALL00840
ALL00850
ALL00860
ALL00870
ALL00880
ALL00890
ALL00900
ALL00910
ALL00920
ALL00930
ALL00940
ALL00950
ALL00960
ALL00970
ALL00980
ALL00990
ALL01000
ALL01010
ALL01020
ALL01030
ALL01040
ALL01050
ALL01060
ALL01070
ALL01080
ALL01090
ALL01100
ALL01110
ALL01120
ALL01130
ALL01140
ALL01150
ALL01160
ALL01170
ALL01180
ALL01190
ALL01200
ALL01210
ALL01220
ALL01230
ALL01240
ALL01250
ALL01260
ALL01270
ALL01280
ALL01290
ALL01300
ALL01310
ALL01320
ALL01330
ALL01340
ALL01350
ALL01360
ALL01370
ALL01380
ALL01390
ALL01400
ALL01410
ALL01420
ALL01430
ALL01440
ALL01450
ALL01460
ALL01470
ALL01480
ALL01490
ALL01500
ALL01510
ALL01520

```

FILE ALLKIN

C	1 CLSTNO(INDEX1)	ALL01530
C	300 CONTINUE	ALL01540
C	WRITE DOT SUMMARY	ALL01550
C	WRITE(6,2222)	ALL01560
2222	FORMAT(1H1,20X,'CLUSTER LABELING SUMMARY',/)	ALL01570
	WRITE(6,1300)	ALL01580
1300	FORMAT(3X,'CLUSTER',20X,'NUMBER OF DOTS USED (BY CATEGORY NAME)',/)	ALL01590
	ISTR1=1	ALL01600
	IEND=NOCAT	ALL01610
	IF (IEND.GT.15) IEND=15	ALL01620
	WRITE(6,1305)	ALL01630
1305	FORMAT(3X,'NUMBER',3X,'LABEL',7X,50(1H-))	ALL01640
	WRITE(6,1330)	ALL01650
1330	FORMAT(1H+,T20,'TOTAL')	ALL01660
	WRITE(6,1310) (CATNAM(IJ),IJ=1,IEND)	ALL01670
1310	FORMAT(30X,1A4,14(3X,1A4))	ALL01680
	DO 600 I=1,NOSUB2	ALL01690
	TOTAL=0	ALL01700
	DO 550 J=1,NOCAT	ALL01710
550	TOTAL=TOTAL+DOTSUM(I,J)	ALL01720
	K=CATVEC(I)	ALL01730
	WRITE(6,1320) I,CATNAM(K),TOTAL,(DOTSUM(I,J),J=1,IEND)	ALL01740
1320	FORMAT(//,5X,12,6X,1A4,15,2X,15(2X,15))	ALL01750
600	CONTINUE	ALL01760
602	IF (IEND.EQ.NOCAT) GO TO 650	ALL01770
	ISTR1=IEND+1	ALL01780
	IEND=NOCAT	ALL01790
	IF (IEND.GT.ISTR1+14) IEND=ISTR1+14	ALL01800
	WRITE(6,1350)	ALL01810
1350	FORMAT(//)	ALL01820
	WRITE(6,1300)	ALL01830
	WRITE(6,1305)	ALL01840
	WRITE(6,1310) (CATNAM(IJ),IJ=ISTR1,IEND)	ALL01850
	DO 610 I=1,ITER	ALL01860
	K=CATVEC(I)	ALL01870
	WRITE(6,1340) I,CATNAM(K),(DOTSUM(I,J),J=ISTR1,IEND)	ALL01880
1340	FORMAT(//,5X,12,6X,1A4,7X,15(2X,15))	ALL01890
610	CONTINUE	ALL01900
	GO TO 602	ALL01910
C	GROUP LABELED CLUSTERS	ALL01920
C	K = 0	ALL01930
650	DO 310 I=1,NOCAT	ALL01940
	DO 310 J=1,NOSUB2	ALL01950
	IF (CATVEC(J) .NE. I) GO TO 310	ALL01960
	SUBNO(I) = SUBNO(I) + 1	ALL01970
	K = K + 1	ALL01980
	SUBVEC(K) = J	ALL01990
310	CONTINUE	ALL02000
	RETURN	ALL02010
C	CODE FOR A TIE	ALL02020
C	390 IF (MAX .EQ. 0) GO TO 175	ALL02030
	DO 400 II=1,NOCAT	ALL02040
	IF (II .EQ. CATNUM) GO TO 400	ALL02050
	IF (MAX .EQ. CATGRY(II)) GO TO 410	ALL02060
400	CONTINUE	ALL02070
	CATVEC(I) = CATNUM	ALL02080
	GO TO 175	ALL02090
C	410 TIE = TIE + 1	ALL02100
	MAXDST = 0	ALL02110
	DO 420 J=1,NODOT	ALL02120
	NO = DOTNU+1(J)	ALL02130
	DST = DISTNC(NO)	ALL02140
	IF (MAXDST .GT. DST) GO TO 420	ALL02150
	DOTNO = NO	ALL02160
	MAXDST = DST	ALL02170
	INDEX = J	ALL02180
	LABEL = DOTS(4,NO)	ALL02190
420	CONTINUE	ALL02200
		ALL02210
		ALL02220
		ALL02230
		ALL02240
		ALL02250
		ALL02260
		ALL02270
		ALL02280

FILE ALLKIN

```
C      TIES(TIE) = DOTNO
C      CATGRY(LABEL) = CATGRY(LABEL) - 1
      NODOT = NODOT - 1
      IF (INDEX .EQ. (NODOT + 1)) GO TO 445
C      DO 440 II=INDEX,NODOT
440    DOTNUM(II) = DOTNUM(II+1)
C      445 CONTINUE
      MAX = 0
      DO 470 II=1,NOCAT
      IF (CATGRY(II) .LE. MAX) GO TO 470
      MAX = CATGRY(II)
      CATNUM = II
470    CONTINUE
      GO TO 390
      END
```

```
ALL02290
ALL02300
ALL02310
ALL02320
ALL02330
ALL02340
ALL02350
ALL02360
ALL02370
ALL02380
ALL02390
ALL02400
ALL02410
ALL02420
ALL02430
ALL02440
ALL02450
ALL02460
ALL02470
```



FILE: ASCEND

```
SUBROUTINE ASCEND(SCN,LNCAT,PTR1,PTR2)
  IMPLICIT INTEGER(A-X)
  DIMENSION PTR1(LNCAT),PTR2(LNCAT)
  REAL SCN(LNCAT),SAVE
  J=0
60  J=J+1
   IF (J.GT.LNCAT)GO TO 90
   IF (J.FD.LNCAT)GO TO 75
   IF (SCN(J) .GT. SCN(J+1)) GO TO 70
   GO TO 60
C
70  SAVE=SCN(J)
   SCN(J)=SCN(J+1)
   SCN(J+1)=SAVE
C
   SAVE1=PTR1(J)
   PTR1(J)=PTR1(J+1)
   PTR1(J+1)=SAVE1
C
   SAVE2=PTR2(J)
   PTR2(J)=PTR2(J+1)
   PTR2(J+1)=SAVE2
75  K=J
A0  IF (K.EQ.1)GO TO 60
   IF (SCN(K) .GT. SCN(K-1)) GO TO 60
C
   SAVE=SCN(K-1)
   SCN(K-1)=SCN(K)
   SCN(K)=SAVE
C
   SAVE1=PTR1(K-1)
   PTR1(K-1)=PTR1(K)
   PTR1(K)=SAVE1
C
   SAVE2=PTR2(K-1)
   PTR2(K-1)=PTR2(K)
   PTR2(K)=SAVE2
   K=K-1
   GO TO A0
90  CONTINUE
   RETURN
   END
```

ASC00010  
ASC00020  
ASC00030  
ASC00040  
ASC00050  
ASC00060  
ASC00070  
ASC00080  
ASC00090  
ASC00100  
ASC00110  
ASC00120  
ASC00130  
ASC00140  
ASC00150  
ASC00160  
ASC00170  
ASC00180  
ASC00190  
ASC00200  
ASC00210  
ASC00220  
ASC00230  
ASC00240  
ASC00250  
ASC00260  
ASC00270  
ASC00280  
ASC00290  
ASC00300  
ASC00310  
ASC00320  
ASC00330  
ASC00340  
ASC00350  
ASC00360  
ASC00370  
ASC00380  
ASC00390  
ASC00400  
ASC00410  
ASC00420

FILE: CLRKEY

```

SUBROUTINE CLRKEY(XSIZ, IDATA, NOSUB2, CH, MEANS, NC)
C
C*
C*   CLRKEY ADDS THE COLOR KEYS TO A UNIVERSAL FORMAT TAPE
C*   THE COLORS ARE OUTPUT AS SQUARES IMAGES (10X10)
C*   IMPLICIT INTEGER (A-Z)
C*   DIMENSION IDATA(XSIZ,CH)
C*   REAL MEANS(NC,NOSUB2)
C
C   LSTLIN = 0
C   LINE = 0
C   TOTKEY = 0
C   NKFYS = XSIZ/11
C   NOKEY = NOSUB2
C
C   90 DO 100 J=1,CH
C*   WRITE A SCAN LINE OF ZEROS - USED FOR SEPARATING THE THE COLORS
C*
C*   DO 100 I=1,XSIZ
C   100 IDATA(I,J) = 0
C
C   CALL WRTLN(IDATA,LSTLIN)
C   LINE = LINE + 1
C
C   110 IF (NKEYS .LE. NOKEY) NOKEY = NKEYS
C   KK = 0
C
C   DO 150 I=1,NOKEY
C   TOTKEY = TOTKEY + 1
C   DO 140 J=1,NC
C   DO 130 K=1,10
C   KK = (I-1)*11 + K
C   130 IDATA(KK,J) = MEANS(J,TOTKEY) + 0.5
C   140 CONTINUE
C*
C*   WRITE A SCAN LINE OF COLORS
C*
C*   150 CONTINUE
C*   NOKEY = NOSUB2 - TOTKEY
C
C   DO 160 I=1,10
C   IF (NOKEY .LE. 0 .AND. I .EQ. 10) LSTLIN = -1
C   160 CALL WRTLN(IDATA,LSTLIN)
C   LINE = LINE + 10
C   IF (NOKEY .LE. 0) GO TO 170
C   GO TO 90
C
C   170 CONTINUE
C
C   RETURN
C   END

```

CLR00010  
CLR00020  
CLR00030  
CLR00040  
CLR00050  
CLR00060  
CLR00070  
CLR00080  
CLR00090  
CLR00100  
CLR00110  
CLR00120  
CLR00130  
CLR00140  
CLR00150  
CLR00160  
CLR00170  
CLR00180  
CLR00190  
CLR00200  
CLR00210  
CLR00220  
CLR00230  
CLR00240  
CLR00250  
CLR00260  
CLR00270  
CLR00280  
CLR00290  
CLR00300  
CLR00310  
CLR00320  
CLR00330  
CLR00340  
CLR00350  
CLR00360  
CLR00370  
CLR00380  
CLR00390  
CLR00400  
CLR00410  
CLR00420  
CLR00430  
CLR00440  
CLR00450  
CLR00460  
CLR00470  
CLR00480  
CLR00490  
CLR00500  
CLR00510  
CLR00520

ORIGINAL PAGE IS  
OF POOR QUALITY



FILE: CLSMAP

```

3 SMALL = 2**30 + (2**30-2)
DO 4 I = 1, NOCAT
  DUM = CATNAM(I) - 2**31
  IF (DUM.LE.SMALLS) GO TO 4
  IF (DUM.GE.SMALL) GO TO 4
  III = I
  SMALL = DUM
4 CONTINUE
  COLATE(III) = II
  SMALLS = SMALL
  IF (DELETE(III).EQ.0) GO TO 6
  II = II + 1
  IF (II.LE.KEPT) GO TO 3
7 CONTINUE
C
C      ASSIGN CATEGORY SYMBOL TO EACH CLUSTER
C
DO 10 I=1,NOSUB2
  CATNUM = CATSUR(I)
  CLSSYM(I) = SYMBOL(CATNUM)
  IF (CATNUM.GT.NOCAT) GO TO 9
  DUM = COLATE(CATNUM)
  DELETE(I) = DUM
  COLORS(I) = COLOR(DUM + 1)
  GO TO 10
9  COLORS(I) = COLOR(64 + KEPT - CATNUM)
  DELETE(I) = 63 + KEPT - CATNUM
10 CONTINUE
   CLSSYM (NOSUB2 + 1) = IPND
   CLSSYM (NOSUB2 + 2) = IPND
  CALL SFTHMG(68,0,68)
  WRITE(6,HEAD)
C
C      CALL MAPHND(NOCAT,CLSSYM,CATNAM,CATVEC,SUBDES,CATSUB)
C
C      WRITE(6,5)(TITLE(I,SWTCH),I=1,3)
C      S FORMAT(/50X,344,'CLUSTER MAP')
C
  ISTRT = FLDINF(4)
  IEND = FLDINF(5)
  SAMINC = FLDINF(6)
  LININC = FLDINF(3)
  LINSTR = FLDINF(1)
  LINEND = FLDINF(2)
  ILINE = (LINEND - LINSTR)/LININC + 1
  NSAMP = (IEND-ISTRT)/SAMINC + 1
  PTS = 0
C
C      POSITION TAPE
C
  REWIND OMAPUN
  IF (OMAPFI.NE.0) CALL FSFMLE(OMAPUN,OMAPFI,ISTAT)
  CALL WRTHED(1,1,NSAMP,1,OMAPUN)
C
C      PRINT LINE PRINTER MAP
C
  IPFLAG = 1
  GO TO 500
14 CONTINUE
  II = 0
C
  IPTS = NSAMP
  IF (IPTS.GT.110) IPTS = 110
  IF (NSAMP.LE.110) GO TO 15
  IPN = NSAMP - 110
15 CONTINUE
  DO 300 I=LINSTR,LINEND,LININC
  II = II + 1
  PIXADR = DRUMADR + (II-1)*NSAMP
  CALL RREAD(PIXADR,IR,NSAMP,ISTAT)
20 IF (ISTAT.EQ.1) GO TO 20
C
C      DRUM ADDRESS MAPADR
C
  DO 30 J=1,NSAMP
  L = IP(J)
  IF (L.NE.0) GO TO 25
  INDATA(J) = 0
  
```

CLS00800  
 CLS00810  
 CLS00820  
 CLS00830  
 CLS00840  
 CLS00850  
 CLS00860  
 CLS00870  
 CLS00880  
 CLS00890  
 CLS00900  
 CLS00910  
 CLS00920  
 CLS00930  
 CLS00940  
 CLS00950  
 CLS00960  
 CLS00970  
 CLS00980  
 CLS00990  
 CLS01000  
 CLS01010  
 CLS01020  
 CLS01030  
 CLS01040  
 CLS01050  
 CLS01060  
 CLS01070  
 CLS01080  
 CLS01090  
 CLS01100  
 CLS01110  
 CLS01120  
 CLS01130  
 CLS01140  
 CLS01150  
 CLS01160  
 CLS01170  
 CLS01180  
 CLS01190  
 CLS01200  
 CLS01210  
 CLS01220  
 CLS01230  
 CLS01240  
 CLS01250  
 CLS01260  
 CLS01270  
 CLS01280  
 CLS01290  
 CLS01300  
 CLS01310  
 CLS01320  
 CLS01330  
 CLS01340  
 CLS01350  
 CLS01360  
 CLS01370  
 CLS01380  
 CLS01390  
 CLS01400  
 CLS01410  
 CLS01420  
 CLS01430  
 CLS01440  
 CLS01450  
 CLS01460  
 CLS01470  
 CLS01480  
 CLS01490  
 CLS01500  
 CLS01510  
 CLS01520  
 CLS01530  
 CLS01540  
 CLS01550  
 CLS01560  
 CLS01570  
 CLS01580

ORIGINAL PAGE IS OF POOR QUALITY

FILE: CLSMAP

```

GO TO 30
25 OUT(J) = CLSSYM(L)
30 IDATA(J) = COLORS(L)
WRITE IDATA OUT TO TAPE
CALL WRTLN(IDATA,LSTLIN)
WRITE REMAINDER OF PIXELS ON DRUM FOR SUBSEQUENCE WRITING
MAPDRM = MAPADR + (I1-1)*IPD
CALL RWRITE(MAPDRM,OUT(I1)),IPD,ISTAT)
50 WRITE(6,60)I.(OUT(IK),IK=I,IPTS)
60 FORMAT(2X,I5,2X,110A1)
300 CONTINUE
XSIZ = NSAMP
CH = 1
DO #0 I = 1,NOSUB2
RELCLR(I) = COLORS(I)
80 CONTINUE
CALL CLRKEY(XSIZ, IDATA, NOSUB2, CH, RELCLR, CH)
ISTART = 1
IPD = IPD
I = 0
305 IPTS = IPTS + IPTS
IF (IPTS .GE. NSAMP) GO TO 360
IF (IPD .GT. 110) IPTS = 110
IF (IPD .LE. 110) IPTS = IPD
IENDS = ISTART + IPTS - 1
IPD = IPD - IPTS
PRINT REST OF MAP
IPFLAG = 2
GO TO 500
308 CONTINUE
DO 350 I=LINSTR,LINEND,LININC
I1 = I + 1
MAPDRM = MAPADR + (I1-1)*IPD
CALL RREAD(MAPDRM,OUT,IPD,ISTAT)
310 IF (ISTAT .EQ. 1) GO TO 310
350 WRITE(6,240)I.(OUT(IK),IK=ISTART,IENDS)
240 FORMAT(2X,I6,1X,110A1)
ISTART = IENDS + 1
GO TO 305
FINISHED
360 CONTINUE
WRITE(6,370)
370 FORMAT(1H, ' NEW ORDERING AND COLOR KEY CODES')
WRITE(6,375)
375 FORMAT(/// 23X, ' OLD NEW CAT COLOR ORDER COLORS'///)
DO 11 I = 1,NOSUB2
WRITE(6,340) I,CATSUB(I),DELETE(I),COLORS(I)
340 FORMAT(20X,4(16,6X))
11 CONTINUE
CALL SFTMRG(68,4,62)
RETURN
500 CONTINUE
J = 0
DO #1 IJ = ISTRT, IEND, SAMINC
J = J + 1
COL(1,J) = IJ/100
COL(2,J) = MOD(IJ,100)/10
COL(3,J) = MOD(IJ,10)
IF (J .EQ. 110) GO TO #2
#1 CONTINUE
#2 SAMEN = IJ
ISTRT = SAMEN + SAMINC
JPTS = J
WRITE(6,95)
DO #3 IJ = 1, 3
#3 WRITE(6,90) (COL(IJ, J), J = 1, JPTS)
90 FORMAT(9X,11011)
95 WRITE(6,95)
95 FORMAT(//)

```

FILE: CLSMAP

IF (IPFLAG .EQ. 1) GO TO 14  
IF (IPFLAG .EQ. 2) GO TO 308  
END

CLS02380  
CLS02390  
CLS02400

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: CNMAP

```
C SURROUTINE CNMAP(DOTS,CNDSUR,CATVEC) CN000010
C CN000020
C FLAGS THE CONDITIONAL CLUSTERS CN000030
C CN000040
C IMPLICIT INTEGER (A-Z) CN000050
C DIMENSION CNDSUR(60),DISTNC(250),DOTS(SIZE,TOTDT2),CATVEC(1) CN000060
C REAL DSTN,DISTNC CN000070
C REAL T CN000080
C INCLUDE COMPK1,LIST CN000090
C INCLUDE CMK15,LIST CN000100
C COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VAR5Z2,TOTVT2,NOFLO2, CN000110
C AVAR2,CNVAR2,CLSID2,SUBNO2,SUBDS2,FLDSV2,VERTX2, CN000120
C FETVC2(30),SURVC2(75),SURPTR(75),CLSVZ2(60), CN000130
C KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61), CN000140
C GRPCHK(41),GROUPS(124) CN000150
C COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60), CN000160
C SURRAY(120),PTR(60),CATPTR(250),CATDOT(500), CN000170
C DOTVEC(250),COND,MIX,PROC,MAPKEY,DOTKEY,STATKY, CN000180
C SUNANG,T,NFARST,DIST,NOFEAT,FETVEC(30),OMAPUN,OMAPP1, CN000190
C OSAVTP,OSTAFI,NOSUN,ANGLE(8),SIZE,TOTDT2,FLDINF(6), CN000200
C CLSSYM(62),STADRS,MEANAD,TABADR,MAPADR,SUNCOR(30), CN000210
C ODOTUN,ODOTF1,MANSTA,MANDOT,OSPUNT,NSPFIL,DSPKEY,PRNSTS, CN000220
C PRINT,FLDNAM,VERTX(22),NOVRT,NSUN,ANGLES(8) CN000230
C ,TOTDT3,FLDADR,VTXADR CN000240
CSEND CN000120
C CN000130
C FIND ALL CONDITIONAL CLUSTERS CN000140
C CN000150
C NEXT = 63 CN000160
C DO 100 I=1,NOSUR2 CN000170
100 CNDSUR(I) = CATVEC(I) CN000180
C DO 200 I=1,NOSUB2 CN000190
C TAB1 = TABADR * (I-1)*TOTDT2 CN000200
C CALL RREAD(TAB1,DISTNC,TOTDT2,ISTAT) CN000210
110 IF (ISTAT .EQ. 1) GO TO 110 CN000220
C CN000230
C DSTN = 255. CN000240
C LABEL = CATVEC(I) CN000250
C DO 120 J=1,TOTDT2 CN000260
C IF (DSTN .LT. DISTNC(J)) GO TO 120 CN000270
C IF (DOTS(4,J) .NE. LABEL) GO TO 120 CN000280
C DSTN = DISTNC(J) CN000290
120 CONTINUE CN000300
C CN000310
C CN000320
C COMPARE THRESHOLD VALUE T CN000330
C CN000340
C IF (DSTN .LF. T) GO TO 200 CN000350
C CN000360
C FLAG AS CONDITIONAL CN000370
C CN000380
C NEXT = NEXT - 1 CN000390
C CNDSUR(I) = NEXT CN000400
C CN000410
200 CONTINUE CN000420
C CN000430
C RETURN CN000440
C END CN000450
```

FILE: CRDSCN

```
C      FUNCTION CRDSCN(CARD,GRPDEX,GRPNAM,GROUPS,NOGRP,GRPTR)          CRD00010
C      IMPLICIT INTEGER (A-H,O-Z)                                     CRD00020
C      DIMENSION BUFL(8), CARD(62), COMVEC(2), NUMVEC(30)           CRD00030
C      DIMENSION GRPDEX(1),GRPNAM(1),GROUPS(1)                     CRD00040
C      LOGICAL*1 L1(4),L2(32)                                       CRD00050
C      EQUIVALENCE (WRD1,L1(1)),(L2(1),BUFL(1))                   CRD00060
C      DATA BLANK/' ', COMMA/','/, COMVEC/1.,'/                  CRD00070
C      -----                                                    CRD00080
C      -----                                                    CRD00090
C      -----                                                    CRD00100
C      -----                                                    CRD00110
C      -----                                                    CRD00120
C      -----                                                    CRD00130
C      COL = 0                                                       CRD00140
C      J = NXTCHR(CARD,COL)                                          CRD00150
C      IF (J.EQ.BLANK) GO TO 110                                     CRD00160
C      DO 10 I=1,8                                                  CRD00170
C      J2 = CARD(COL-1+I)                                          CRD00180
C      IF (J2.EQ.COMMA) GO TO 20                                    CRD00190
10  RUF(I) = J2                                                    CRD00200
C      GO TO 40                                                    CRD00210
C      -----                                                    CRD00220
C      20 DO 30 J=1,8                                               CRD00230
C      30 RUF(J) = BLANK                                           CRD00240
C      -----                                                    CRD00250
C      40 N = 1                                                     CRD00260
C      DO 50 I=1,4                                                 CRD00270
C      L1(I) = L2(N)                                               CRD00280
C      N = N + 4                                                    CRD00290
C      50 CONTINUE                                                CRD00300
C      GRPNAM(NOGRP+1) = WRD1                                       CRD00310
C      J = FIND12(CARD,COL,COMVEC)                                  CRD00320
C      IF (J.LE.0) GO TO 110                                       CRD00330
C      J = NUMBER(CARD,COL,NUMVEC,0)                                CRD00340
C      II = 0                                                       CRD00350
C      DO 90 I=1,J                                                 CRD00360
C      JJ = NUMVEC(I)                                              CRD00370
80  II = II+1                                                    CRD00380
C      NUMVEC(II) = JJ                                             CRD00390
C      90 CONTINUE                                                CRD00400
C      IF (II.LE.0) GO TO 110                                       CRD00410
C      -----                                                    CRD00420
C      NOGRP = NOGRP+1                                             CRD00430
C      GRPTR = GRPTR + 1                                           CRD00440
C      GRPDEX(NOGRP) = GRPTR                                       CRD00450
C      GROUPS(GRPTR) = II                                          CRD00460
C      DO 100 I=1,II                                               CRD00470
100 GROUPS(GRPTR+I) = NUMVEC(I)                                  CRD00480
C      GRPTR = GRPTR+II                                           CRD00490
C      CRDSCN = 0                                                  CRD00500
C      RETURN                                                       CRD00510
C      -----                                                    CRD00520
C      -----                                                    CRD00530
C      -----                                                    CRD00540
C      110 RETURN                                                 CRD00550
C      END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

18-13  
344



FILE: DOTDST

```
C      SURROUTINE DOTDST(MEANS,DOTS,TABLE,TOP)                                DOT00010
C      COMPUTES L1 OR L2 DISTANCES AND STORE ON DRUM                          DOT00020
C      IMPLICIT INTEGER (A-Z)                                                DOT00030
C      INCLUDE COMMK1.LIST                                                    DOT00040
C      INCLUDE COMMK4.LIST                                                    DOT00050
C      INCLUDE COMMK6.LIST                                                    DOT00060
C      INCLUDE COMMK15.LIST                                                   DOT00070
COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VAR572,TOTVT2,NOFLD2,                      DOT00080
*   AVAR2,COVAR2,CLS1D2,SUBNO2,SURDS2,FLDSV2,VERTX2,                          DOT00090
*   FETVC2(30),SUBVC2(75),SUHPTR(75),CLSV2(60),                              DOT00100
*   KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),                                  DOT00110
*   GRPCHK(61),GROUPS(124)                                                  DOT00120
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)                             DOT00130
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),                          DOT00140
2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))                                  DOT00150
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,                  DOT00160
*   HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,                       DOT00170
*   DRUMAD,DRM4DS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL                          DOT00180
*   ,NHSTUN,NHSTFI,SCTRUN,MAPFIL                                           DOT00190
*   ,DOTUNT,DOTFIL,NCHPAS,TRANSFL,BMTRFL,HISTFL,PCHUNT,                    DOT00200
*   CRDUNT,PRJUNT,HANDIO                                                    DOT00210
COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60),          DOT00220
*   SUBRAY(120),PTR(60),CATPTR(250),CATDOT(500),                            DOT00230
*   DOTVEC(250),COND,MIX,PROC,MAPKEY,DOTKEY,STATKY,                         DOT00240
*   SUNANG,T,NFARST,DIST,NOFEAT,FETVEC(30),OMAPUN,OMAPFI,                  DOT00250
*   OSAVTP,OSTAFI,NOSUN,ANGLE(A),SIZE,TOTDT2,FLDINF(6),                    DOT00260
*   CLSSYM(62),STADKS,MEANAD,TABADR,MAPADR,SUNCOR(30),                      DOT00270
*   ODOTUN,ODOTFI,MANSTA,MANDOT,OSPUNT,DSPFIL,DSPKEY,PRNSTS,              DOT00280
*   PRNDOT,FLDNAM,VERTEX(22),NOVRT,NSUN,ANGLES(8)                           DOT00290
*   ,TOTDT3,FLDADR,VTXADR                                                  DOT00300
CSEND DIMENSION DOTS(SIZE,1)                                                DOT00310
REAL TABLE(TOTDT2,1)                                                       DOT00320
DATA CLSTR/CLST1/,BLANK/0 0/                                               DOT00330
REAL MEANS(NOFET2,1),DSTN,DISTNC(250),SUN,SUNCOR                           DOT00340
TABADR - DRUM ADDRESS FOR STORING DISTANCE TABLE                          DOT00350
TAB1 = TABADR                                                                DOT00360
RETRIEVE SUN ANGLE CORRECTIONS                                              DOT00370
IF (SUNANG .EQ. 0) GO TO 19                                                 DOT00380
SWTCH = 1 - SUNANGLES ARE USER INPUT                                       DOT00390
      = 0 -SUN ANGLES ARE ON DOT FILE                                       DOT00400
IF (SUNANG .NE. 1) GO TO 18                                                 DOT00410
SWTCH = 0                                                                      DOT00420
CALL SUNFAC(SUNCOR,ANGLE,FETVEC,NOFEAT,SWTCH,DUMMY)                         DOT00430
GO TO 19                                                                       DOT00440
1A  SWTCH = 1                                                                    DOT00450
CALL SUNFAC(SUNCOR,ANGLES,FETVEC,NOFEAT,SWTCH,DUMMY)                       DOT00460
19  CONTINUE                                                                    DOT00470
DO 200 I=1,NOSUB2                                                            DOT00480
ZERO OUT ARRAY                                                                DOT00490
DO 20 IJ=1,TOTDT2                                                            DOT00500
20  DISTNC(IJ) = 0                                                            DOT00510
COMPUTE DISTANCE BETWEEN ALL DOTS FOR EACH CLUSTER                         DOT00520
DIST = 1 -- L1 DISTANCE                                                       DOT00530
      = 2 -- L2 DISTANCE                                                       DOT00540
DO 100 J=1,TOTDT2                                                            DOT00550
DSTN = 0                                                                      DOT00560
DO 50 K=1,NOFET2                                                            DOT00570
SUN = SUNCOR(K)                                                              DOT00580
DSTN = SUN*DOTS(4+K,J) - SUN*MEANS(K,I)                                     DOT00590
GO TO (30,40),DIST                                                            DOT00600
30  DISTNC(J) = DISTNC(J) + ABS(DSTN)                                       DOT00610
GO TO 50                                                                      DOT00620
DOT00630
DOT00640
DOT00650
DOT00660
DOT00670
DOT00680
DOT00690
DOT00700
DOT00710
DOT00720
DOT00730
DOT00740
DOT00750
DOT00760
DOT00770
DOT00780
DOT00790
```

FILE: DOTDST

```

40 DISTNC(J) = DISTNC(J) * DSTN**2
50 CONTINUE
C
100 IF (DIST .EQ. 2) DISTNC(J) = SORT(DISTNC(J))
    CALL PWRITE(TAB1,DISTNC,TOTDT2,ISTAT)
    TAB1 = TAB1 + TOTDT2
110 IF (ISTAT .EQ. 1) GO TO 110
200 CONTINUE
C
    WRITE(6,HEAD)
    WRITE(6,250)
C
250 FORMAT(/T50,'CLUSTER-DOT INTER-DISTANCE TABLE')
C
    SUR2 = NOSUR2
    TAB1 = TABADR
    NSUR2 = 1
    NOGRPS = TOP/TOTDT2
    IF (NOGRPS .GT. 15) NOGRPS = 15
245 IF (NOGRPS .GT. SUR2) NOGRPS = SUB2
    TOTWDS = TOTDT2 * NOGRPS
C
    CALL RREAD(TAB1,TABLE,TOTWDS,ISTAT)
    TSUR2 = NSUR2 + NOGRPS - 1
    WRITE(6,260)(CLSTR ,I=1,NOGRPS)
    WRITE(6,261)(BLANK,SURVC2(K),K=NSUB2,TSUB2)
260 FORMAT(/78X,15(4X,A4))
261 FORMAT(4X,'DOTS',3X,15(A1,'(',I2,')',3X))
    WRITE(6,262)
262 FORMAT( )
265 IF (ISTAT .EQ. 1) GO TO 265
    DO 266 I=1,TOTDT2
    WRITE(6,275)DOTVEC(I),(TABLE(I,J),J=1,NOGRPS)
275 FORMAT(4X,I3,1H.,15(1X,F7.2))
280 CONTINUE
    TAB1 = TAB1 + TOTWDS
    NSUR2 = TSUR2 + 1
    SUR2 = SUR2 - NOGRPS
    IF (SUR2 .GT. 0) GO TO 245
C
    RETURN
    END
DOT00800
DOT00810
DOT00820
DOT00830
DOT00840
DOT00850
DOT00860
DOT00870
DOT00880
DOT00890
DOT00900
DOT00910
DOT00920
DOT00930
DOT00940
DOT00950
DOT00960
DOT00970
DOT00980
DOT00990
DOT01000
DOT01010
DOT01020
DOT01030
DOT01040
DOT01050
DOT01060
DOT01070
DOT01080
DOT01090
DOT01100
DOT01110
DOT01120
DOT01130
DOT01140
DOT01150
DOT01160
DOT01170
DOT01180
DOT01190
DOT01200
```

ORIGINAL PAGE  
OF POOR QUALITY

FILE: DSPTAP

```
SUBROUTINE DSPTAP(SURNO,SURDES,FLDSAV,VERTX,CATVEC,SUBVEC,MEANS, DSP00010
* COVAR, TOP, DATA, NOFLD, TOTVRT) DSP00020
IMPLICIT INTEGER (A-Z) DSP00030
INCLUDE COMK1.LIST DSP00040
INCLUDE COMK4.LIST DSP00050
INCLUDE COMK6.LIST DSP00060
INCLUDE COMK15.LIST DSP00070
COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2, DSP00080
* AVAP2,COVAR2,CLSID2,SURNO2,SURDS2,FLDSV2,VERTX2, DSP00090
* FETVC2(30),SURVC2(75),SUBPTR(75),CLSV2(60), DSP00100
* KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61), DSP00110
* GRPCHK(61),GROUPS(124) DSP00120
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15) DSP00130
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)), DSP00140
2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48)) DSP00150
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAF,SAVTAP,9MFILE,9MKEY, DSP00160
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE, DSP00170
* DRUMAD,DRMWD5,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVFL DSP00180
* ,NHSTUN,NHSTFI,SCTRIUN,MAPFIL DSP00190
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, DSP00200
* CROUT,PRUNT,RANDIO DSP00210
COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60), DSP00220
* SURPAY(120),PTR(60),CATPTR(250),CATDOT(500), DSP00230
* DOTVEC(250),COND,MIX,PROC,MAPKEY,DOTKEY,STATKY, DSP00240
* SUNANG,T,NEARST,DIST,NOFEAT,FETVEC(30),UMAPUN,OMAPFI, DSP00250
* OSAVTP,OSTAFI,NOSUN,ANGLE(A),SIZE,TOTDT2,FLDINF(6), DSP00260
* CLSSYM(62),STADRS,MEANAD,TABADR,MAPADR,SUNCOR(30), DSP00270
* ODOTUN,ODOTFI,MANSTA,MANDOT,DSPUNT,DSPFIL,DSPKEY,PRNSTS, DSP00280
* PRNDOT,FLDNAM,VERTEX(22),NOVRT,NSUN,ANGLES(8) DSP00290
* ,TOTDT3,FLDADR,VTXADR DSP00300
C*END DSP00310
DIMENSION SURNO(1),SURDES(1),FLDSAV(4,1),VERTX(2,1),CATVEC(1) DSP00320
DIMENSION SUBVEC(1),DATA(1),IP(1000),NEWSUB(62) DSP00330
DIMENSION CLSVFC(60),DUMMY(100) DSP00340
REAL MEANS(NOFET2,1),COVAR(VARSZ2,1),VR(1000) DSP00350
C DSP00360
C POSITION TAPE DSP00370
C DSP00380
C CALL F88SFL(DSPUNT,DSPFIL,IST) DSP00390
C DSP00400
C RUN HEADER RECORD NO. 1 DSP00410
C DSP00420
C DO 10 I=1,100 DSP00430
10 DUMMY(I)=1 DSP00440
NOFLD2 = NOFLD DSP00450
TOTVT2 = TOTVRT DSP00460
NCAT = 0 DSP00470
C**** JULY 12 1978 NOCAT USED INSTEAD OF NCAT IN MAPTAP FIRST RECORD DSP00480
C DSP00490
C WRITE(DSPUNT)((DATE(I),I=1,2),(DUMMY(I),I=1,3),NOCAT,NOFLD2,NOSUB2, DSP00500
* NOFET2,TOTVT2,NOCAT,VARSZ2,(FETVC2(I),I=1,NOFET2) DSP00510
C DSP00520
C K = 0 DSP00530
DO 40 I=1,NOCAT DSP00540
III = SURNO(I) DSP00550
DO 40 J=1,III DSP00560
K = K + 1 DSP00570
40 CLSVEC(K) = I DSP00580
C RUN HEADER RECORD NO. 2 DSP00590
C WRITE(DSPUNT)(CATNAM(I),I=1,NOCAT),(CATNAM(I),I=1,NOCAT), DSP00600
* (SURNO(I),I=1,NOCAT),(SURDES DSP00610
* (I),I=1,NOSUR2),((FLDSAV(I,J),I=1,4),J=1,NOFLD2),((VERTX(I,J) DSP00620
* ,I=1,2),J=1,TOTVT2),(CLSVEC(I),I=1,NOSUB2),(CLSVEC(I),I=1,NOSUB2) DSP00630
* ,(DUMMY(I),I=1,NOCAT),(KFPPTS(I),I=1,NOSUR2) DSP00640
C DSP00650
C RUN HEADER RECORD NO. 3 DSP00660
C DSP00670
C MEAN1 = STADRS + VARSZ2 * NOSUB2 DSP00680
DO 100 J=1,NOSUB2 DSP00690
KK = SURVEC(J) DSP00700
MEANS2 = MEAN1 + NOFET2*(KK-1) DSP00710
COVAR1 = STADRS + VARSZ2*(KK-1) DSP00720
CALL RPFAD(COVAR1,COVAR(1,J),VARSZ2,ISTAT) DSP00730
50 IF (ISTAT.EQ. 1) GO TO 50 DSP00740
CALL RPFAD(MEANS2,MEANS(1,J),NOFET2,ISTAT) DSP00750
60 IF (ISTAT.EQ. 1) GO TO 60 DSP00760
C DSP00770
C 100 CONTINUE DSP00780
C DSP00790
```

FILE: DSPTAP

```
      WRITE(DSPUNT) ((COVAR(I,J),I=1,VARSZ2),J=1,NOSUB2), ((MEANS(I,J),
C      * I=1,NOFET2),J=1,NOSUB2)
C      RUN HEADER RECORD NO. 4
C      WRITE(DSPUNT) ((COVAR(I,J),I=1,VARSZ2),J=1,NOSUB2),
C      * (DUMMY(I),I=1,NOSUB2), (DUMMY(I),I=1,NOSUB2)
C      FIELD RECORD
      LINSTR = FLDINF(1)
      LINEND = FLDINF(2)
      LININC = FLDINF(3)
      SAMSTR = FLDINF(4)
      SAMEND = FLDINF(5)
      SAMINC = FLDINF(6)
      PTS = (SAMEND-SAMSTR)/SAMINC + 1
      LINES = (LINEND-LINSTR)/LININC + 1
C      WRITE(DSPUNT) (FLDINF(I),I=1,6),PTS,LINES,FLDNAM,NOVRT,
C      * (VERTEX(I),I=1,NOVRT), (VERTEX(I+NOVRT),I=1,NOVRT)
C      NEWSUB -- NEW SUBCLASS NUMBERS
      DO 120 I=1,NOSUB2
      K = SURVEC(I)
120  NEWSUR(K) = I
      DO 130 I=1,PTS
130  VR(I) = 0.0
      MAP = DMUMAD
      ILINE = LINES
      NOLINE = TOP/PTS
135  IF (NOLINE .GT. ILINE) NOLINE = ILINE
      TOTPIX = NOLINE * PTS
      CALL WREAD(MAP,DATA,TOTPIX,ISTAT2)
      MAP = MAP + TOTPIX
137  IF (ISTAT2 .EQ. 1) GO TO 137
      DO 150 I=1,NOLINE
      II = II + 1
      N = LINSTR + LININC*(II-1)
C      DO 140 J=1,PTS
      IDUM=(I-1)*PTS+J
      JJ = DATA(IDUM)
140  IR(J) = NEWSUR(JJ)
C      WRITE(DSPUNT)N,(IR(K),K=1,PTS),(VR(K),K=1,PTS)
150  CONTINUE
      ILINE = ILINE - NOLINE
      IF (ILINE .LE. 0) GO TO 155
155  N = 0
      WRITE(DSPUNT)N,(IR(I),I=1,PTS),(VR(I),I=1,PTS)
      PTS = 0
      WRITE(DSPUNT) (FLDINF(I),I=1,6),PTS,LINES,FLDNAM,NOVRT,
      * (VERTEX(I),I=1,NOVRT), (VERTEX(I+NOVRT),I=1,NOVRT)
      ENDFILE DSPUNT
C      RETURN
      END
```

DSP00800  
DSP00810  
DSP00820  
DSP00830  
DSP00840  
DSP00850  
DSP00860  
DSP00870  
DSP00880  
DSP00890  
DSP00900  
DSP00910  
DSP00920  
DSP00930  
DSP00940  
DSP00950  
DSP00960  
DSP00970  
DSP00980  
DSP00990  
DSP01000  
DSP01010  
DSP01020  
DSP01030  
DSP01040  
DSP01050  
DSP01060  
DSP01070  
DSP01080  
DSP01090  
DSP01100  
DSP01110  
DSP01120  
DSP01130  
DSP01140  
DSP01150  
DSP01160  
DSP01170  
DSP01180  
DSP01190  
DSP01200  
DSP01210  
DSP01220  
DSP01230  
DSP01240  
DSP01250  
DSP01260  
DSP01270  
DSP01280  
DSP01290  
DSP01300  
DSP01310  
DSP01320  
DSP01330  
DSP01340  
DSP01350  
DSP01360  
DSP01370  
DSP01380  
DSP01390  
DSP01400

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: FILERD

```

SURROUTINE FILFRD (ARRAY, TOP, NOFLD, TOTVRT, FLDSAV, VERTX)
READS IN ALL NEEDED FILES

DRUM ADDRESSES
DRUMAD - BEGINNING ADDRESS FOR MAPFIL
STADRS - BEGINNING ADDRESS FOR COVAR AND MEANS
TABADR - BEGINNING ADDRESS FOR DISTANCE TABLE
MAPADR - BEGINNING ADDRESS FOR NSAMP-110 PTS OF COND. OR
        MIXED CLUSTER MAP
FLDADR - BEGINNING ADDRESS FOR FIELD INFO
VTXADR - BEGINNING ADDRESS FOR VERTICES
COVAR2 - BEGINNING ADDRESS FOR DOT DATA

IMPLICIT INTEGER (A-Z)
LIMIT = 5000
INCLUDE COMRK1.LIST
INCLUDE COMRK6.LIST
INCLUDE COMRK15.LIST
COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
*   AVAR2, COVAR2, CLSID2, SURNO2, SUROS2, FLDSV2, VERTX2,
*   FETVC2(30), SURVC2(75), SURPTR(75), CLSYC2(60),
*   KFPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),
*   GRPCHK(61), GROUPS(124)
COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
*   HISFIL, HISKEY, TRFORM, ERIPTR, ERPKEY, MAPUNT, NOFILE,
*   DRUMAD, DRUMDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
*   , NHSTUN, NHSTFI, SCTRUN, MAPFIL
*   , DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
*   CRDUNT, PRTUNT, WAND10
COMMON /LAHS/NOCAT, CATNAM(60), NOCL2, CLSNM2(60), NOCAT2, CATNM2(60),
*   SURRAY(120), PTR(60), CATPTR(250), CATDOT(500),
*   DOTVEC(250), COND, MIX, PROC, MAPKEY, DOTKEY, STATKY,
*   SUNANG, T, NEARST, DIST, NOFFAT, FETVEC(30), OMAPUN, OMAFFI,
*   OSAVTP, USTAFI, NOSUN, ANGLE(8), SIZE, TOTDT2, FLDINF(6),
*   CLSSYM(62), STADRS, MEANAD, TABADR, MAPADR, MAPADR, SUNCOR(30),
*   ODOTUN, ODOTFI, MANSTA, MANDOT, DSPUNT, DSPFIL, DSPKEY, PRNSTS,
*   PRNDOT, FLDNAM, VERTEX(22), NOVRT, NSUN, ANGLES(8)
*   , TOTDT3, FLDADR, VTXADR

CSEND
DIMENSION ARPAY(1)
DIMENSION FLDSAV(4,1), VERTX (2,1)
DIMENSION FETVC3(30)

COVAR2 = 1

READ IN MAPFIL AND STORE ON THE DRUM
IF (MAPKEY .EQ. 0) GO TO 100
I = LAREAD(FLDNAM, VERTEX, FLDINF, NOVRT)
NOLINE = (FLDINF(2) - FLDINF(1))/FLDINF(3) + 1
NSAMP = (FLDINF(5) - FLDINF(4))/FLDINF(6) + 1
TOTPIX = NOLINE*NSAMP

CALL STOMAP(NOLINE, NSAMP, ARRAY, TOP, DRUMAD)
READ IN STAT FILE
100 IF (STATKY .EQ. 0) GO TO 200
CALL REDSAV(ARRAY, TOP, HMFLG)
STASIZ = (VARSZ2 + NOFET2) * NOSUB2
STADRS = DRUMAD + TOTPIX
CALL RWRITE(STADRS, ARPAY(COVAR2), STASIZ, ISTAT)
110 IF (ISTAT .EQ. 1) GO TO 110

PEAD IN DOTFIL
200 IF (DOTKEY .EQ. 0) RETURN

TOTSTO = TOP - COVAR2
IF (TOTSTO .GE. LIMIT) GO TO 220

```

FIL00010  
FIL00020  
FIL00030  
FIL00040  
FIL00050  
FIL00060  
FIL00070  
FIL00080  
FIL00090  
FIL00100  
FIL00110  
FIL00120  
FIL00130  
FIL00140  
FIL00150  
FIL00160  
FIL00170  
FIL00180  
FIL00190  
FIL00200  
FIL00210  
FIL00220  
FIL00230  
FIL00240  
FIL00250  
FIL00260  
FIL00270  
FIL00280  
FIL00290  
FIL00300  
FIL00310  
FIL00320  
FIL00330  
FIL00340  
FIL00350  
FIL00360  
FIL00370  
FIL00380  
FIL00390  
FIL00400  
FIL00410  
FIL00420  
FIL00430  
FIL00440  
FIL00450  
FIL00460  
FIL00470  
FIL00480  
FIL00490  
FIL00500  
FIL00510  
FIL00520  
FIL00530  
FIL00540  
FIL00550  
FIL00560  
FIL00570  
FIL00580  
FIL00590  
FIL00600  
FIL00610  
FIL00620  
FIL00630  
FIL00640  
FIL00650  
FIL00660  
FIL00670  
FIL00680  
FIL00690  
FIL00700  
FIL00710  
FIL00720  
FIL00730  
FIL00740  
FIL00750  
FIL00760  
FIL00770  
FIL00780  
FIL00790

FILE: FILERD

```
WRITE(6,210)
210 FORMAT(' NOT ENOUGH CORE TO STORE DOTFIL')
CALL CMERK
C
220 TYPST = 3
CALL RDDOTS (ARRAY(COVAR2), DOTVEC, TOTDT3, TYPST, SIZE, TOTDT2, NOCAT,
1 CATNAM, NOFEAT, FETVEC, NOFET3, FETVC3, NOSUN, ANGLE, NOFLD,
2 TOTVRT, FLDSAV, VERTX, DUMMY)
CALL WRFLD (FLDSAV, VERTX, NOFLD, 2, CATNAM, DUMMY)
TABADR = STADR + (VARSZ2 * NOFEAT) * NOSUB2
MAPADR = TABADR + NOSUB2 * TOTDT2
TOTAL = MAPADR
FLDADR = TOTAL
IF (MIX .EQ. 0 .AND. COND .EQ. 0) GO TO 222
FLDADR = TOTAL + (NSAMP - 110) * NOLINE
IF (FLDADR .LT. TOTAL) FLDADR = TOTAL
TOTAL = FLDADR
222 IF (DSPKEY .EQ. 0) GO TO 225
VTXADR = TOTAL + 4 * NOFLD
TOTAL = VTXADR + TOTVRT * 2
C
225 CONTINUE
IF (TOTAL .LE. (DRUMAD + DRMWDS)) GO TO 230
WRITE(6,400)
400 FORMAT(' NOT ENOUGH CORE DRUM SPACE OF CLUSTER MAP INFO')
230 CONTINUE
C
C
C RETURN
C
C END
```

FIL00800  
FIL00810  
FIL00820  
FIL00830  
FIL00840  
FIL00850  
FIL00860  
FIL00870  
FIL00880  
FIL00890  
FIL00900  
FIL00910  
FIL00920  
FIL00930  
FIL00940  
FIL00950  
FIL00960  
FIL00970  
FIL00980  
FIL00990  
FIL01000  
FIL01010  
FIL01020  
FIL01030  
FIL01040  
FIL01050  
FIL01060  
FIL01070  
FIL01080  
FIL01090  
FIL01100

FILE KNEAR

```
      SURROUTINE KNEAR(DOTS,SUBVEC,SUBNO,CATVEC,ITER,TAB1,SWTCH,
      * CATNUM,CLUNUM,MEANS,DOTSUM)
      LABELS BY THE K-NEAREST NEIGHBOR PROCEDURE
      IMPLICIT INTEGER (A-Z)
      REAL DISTNC(250)
      DIMENSION DOTS(SIZE,1),SUBNO(1),SUBVEC(60)
      DIMENSION CATVEC(1),DOTNAM(250),DOTSUM(60,60)
      DIMENSION CATGRY(60),DOTCLU(250)
      INCLUDE COMRK1,LIST
      INCLUDE COMBK4,LIST
      INCLUDE CMRK15,LIST
      INCLUDE COMBK6,LIST
      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
      * AVAR2,COVAR2,CLSID2,SURNO2,SUBDS2,FLDSV2,VRTX2,
      * FETVC2(30),SURVC2(75),SURPTR(75),CLSVC2(60),
      * KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
      * GRPCHK(61),GROUPS(124)
      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
      EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
      * (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
      2 COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60),
      * SURREY(120),PTR(60),CATPTR(250),CATDOT(500),
      * DOTVEC(250),COND,MIX,PROC,MAPKEY,DOTKEY,STATKY,
      * SUNANG,T,NEARST,DIST,NOFEAT,FETVEC(30),OMAPUN,OMAPFI,
      * OSAVTP,OSTAFI,NOSUN,ANGLE(8),SIZE,TOTDT2,FLDINF(6),
      * CLASSYM(62),STADRS,MEANAD,TABADR,MAPADR,SUNCOR(30),
      * ODOTUN,ODOTFI,MANSTA,MANDOT,DSPUNT,DSPFIL,DSPKEY,PRNSTS,
      * PRNDOT,FLDNAM,VERTEX(22),NOVRT,NSUN,ANGLES(8)
      * TOTDT3,FLDACR,VTXADR
      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVAP,BMFILE,BMKEY,
      * HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
      * DRUMAD,DRMWS,PAGSIZ,DAFIL,STAFIL,ASAV,ASAVFL
      * NHSTUN,NHSTFI,SCRUN,MAPFIL
      * DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PEHUNT,
      * CRDUNT,PRUNT,RANDIO
      NO OF DOTS TO COMPARE
      REAL MEANS(NOFET2,1)
      DIMENSION DOTVC2(250),TIES(250)
      SAVE DOTVEC
      DO 5 I=1,TOTDT2
      5 DOTVC2(I) = DOTVEC(I)
      FIND CLUSTER NUMBER OF DOTS IF MAP AVAILABLE
      IF (MAPKEY.EQ.0)GO TO 15
      NSAMP=(FLDINF(5)-FLDINF(4))/FLDINF(6)+1
      DO 12 I=1,TOTDT2
      ILINE=DOTS(2,I)
      ISAMP=DOTS(1,I)
      PIXADR=DRUMAD+(ILINE/FLDINF(3)-1)*NSAMP+ISAMP/FLDINF(6)-1
      CALL RREAD(PIXADR,NUMBER,1,ISTAT)
      DOTCLU(I)=NUMBER
      CONTINUE
      12 IF (SWTCH.EQ.1)GO TO 6
      IF (SWTCH.EQ. 0) WRITE(6,HEAD)
      WRITE(6,10)NEARST
      10 FORMAT(/T43,'LABELING BY',I3,'-NEAREST NEIGHBOR PROCEDURE'/)
      KNGHRR = NEARST
      WRITE(6,1111)
      1111 FORMAT(20X,'CLUSTER LABELING DETAILS',/)
      WRITE(6,1000)
      1000 FORMAT(3X,'CLUSTER',2X,'CLUSTER',3X,'DOT',4X,'DOT',
      1 7X,'DOT',6X,'DOT',/,3X,'NUMBER',4X,'LABEL',3X,'LABEL',
      2 2X,'NUMBER',2X,'DISTANCE',2X,'CLUSTER',/)
      DO 500 I=1,ITER
      500
```

FILE KNEAR

```

C      READ IN DISTANCES FROM DRUM ONE CLUSTER AT A TIME
C      TIE = 0
C      II = I
C      CALL RREAD(TAB1,DISTNC,TOTDT2,ISTAT)
50  IF (ISTAT.EQ.1) GO TO 50
C      TAB1 = TAB1 + TOTDT2
C      SORT DISTANCES IN ASCENDING ORDER
C      DO 55 J=1,TOTDT2
C      DOTVEC(J) = DOTVC2(J)
55  DOTNAM(J) = DOT5(4,J)
C      CALL ASCEND(DISTNC,TOTDT2,DOTNAM,DOTVEC)
C      REINITIALIZE ARRAYS
57  DO 60 J=1,NOCAT
60  CATGRY(J) = 0
C      MAX = 0
C      DO 70 JJ=1,KNGHBR
C      L = DOTNAM(JJ)
C      RETRIEVE CATEGORY NO.
C      CATGRY(L) = CATGRY(L) + 1
C      IF (CATGRY(L) .LE. MAX) GO TO 70
C      MAX = CATGRY(L)
C      CATNUM = L
70  CONTINUE
C      CHECK FOR A TIE
C      IF (KNGHBR .EQ. 1) GO TO 100
C      DO 80 III=1,NOCAT
C      IF (III .EQ. CATNUM) GO TO 80
C      IF (IMAX .EQ. CATGRY(III)) GO TO 90
80  CONTINUE
C      NO TIES OCCURRED
C      GO TO 100
C      A TIE OCCURRED - DECREASE K-DOTS BY 1 AND REPEAT PROCESS
90  KNGHRR = KNGHBR - 1
C      TIE = TIE + 1
C      TIES(TIE) = KNGHRR + 1
C      GO TO 57
C      ASSIGN CLUSTER TO CATEGORY
100 IF (SWTCH .EQ. 1) II = CLUNUM
C      CATVEC(II) = CATNUM
C      PRINT CLUSTER INFORMATION
C      WRITE(6,1100) II,CATNAM(CATNUM)
1100 FORMAT(/,5X,12,8X,1A4)
C      DO 110 J=1,KNGHBR
C      K=DOTNAM(J)
C      L=DOTCLU(DOTVEC(J))
C      IF (J.EQ.1) WRITE(6,1201) CATNAM(K),DOTVEC(J),DISTNC(J)
1201 FORMAT(1H+.T24,1A4,2X,13,4X,F7.2)
C      IF (J.GT.1) WRITE(6,1200) CATNAM(K),DOTVEC(J),DISTNC(J)
1200 FORMAT(23X,1A4,2X,13,4X,F7.2)
1201 IF (DOTKEY.EQ.1) WRITE(6,1210) L
1210 FORMAT(1H+.T4R,12)
C      DOTSUM(II,K)=DOTSUM(II,K)+1
110  CONTINUE
C      IF (TIE .EQ. 0) GO TO 490
C      WRITE(6,185)
185  FORMAT(/)
C      WRITE(6,190)
190  FORMAT(23X,'A TIE OCCURRED.',3X,'THE FOLLOWING DOT(S) WERE DISCARDKNE01510

```

KNE00770  
 KNE00780  
 KNE00790  
 KNE00800  
 KNE00810  
 KNE00820  
 KNE00830  
 KNE00840  
 KNE00850  
 KNE00860  
 KNE00870  
 KNE00880  
 KNE00890  
 KNE00900  
 KNE00910  
 KNE00920  
 KNE00930  
 KNE00940  
 KNE00950  
 KNE00960  
 KNE00970  
 KNE00980  
 KNE00990  
 KNE01000  
 KNE01010  
 KNE01020  
 KNE01030  
 KNE01040  
 KNE01050  
 KNE01060  
 KNE01070  
 KNE01080  
 KNE01090  
 KNE01100  
 KNE01110  
 KNE01120  
 KNE01130  
 KNE01140  
 KNE01150  
 KNE01160  
 KNE01170  
 KNE01180  
 KNE01190  
 KNE01200  
 KNE01210  
 KNE01220  
 KNE01230  
 KNE01240  
 KNE01250  
 KNE01260  
 KNE01270  
 KNE01280  
 KNE01290  
 KNE01300  
 KNE01310  
 KNE01320  
 KNE01330  
 KNE01340  
 KNE01350  
 KNE01360  
 KNE01370  
 KNE01380  
 KNE01390  
 KNE01400  
 KNE01410  
 KNE01420  
 KNE01430  
 KNE01440  
 KNE01450  
 KNE01460  
 KNE01470  
 KNE01480  
 KNE01490  
 KNE01500  
 KNE01510



FILE KNEAR

```

*ED*/)
DO 200 JJ=1,TIE
J = TIES(JJ)
K = DOTNAM(J)
L=DOTCLU(DOTVEC(J))
WRITE(6,1200)CATNAM(K),DOTVEC(J),DISTNC(J)
IF (DOTKEY.EQ.1)WRITE(6,1210)L
200 CONTINUE
    KNGHBR=KNGHBR+TIE
C
490 CONTINUE
IF (SWTCH .NE. 1) GO TO 500
C
RESTORE DOTVEC
C
DO 210 J=1,TOTDT2
210 DOTVEC(J) = DOTVC2(J)
500 CONTINUE
IF (SWTCH .EQ. 1) RETURN
C
WRITE DOT SUMMARY
2222 WRITE(6,2222)
    FORMAT(1H1,20X,'CLUSTER LABELING SUMMARY',/)
    WRITE(6,1300)
1300 FORMAT(3X,'CLUSTER',20X,'NUMBER OF DOTS USED (BY CATEGORY NAME)')
    ISTRT=1
    IEND=NOCAT
    IF (IEND.GT.15) IEND=15
    WRITE(6,1305)
    FORMAT(3X,'NUMBER',3X,'LABEL',7X,50(1H-))
1305 WRITE(6,1330)
    FORMAT(1H*,T20,'TOTAL:')
1330 WRITE(6,1310) (CATNAM(IJ),IJ=1,IEND)
1310 FORMAT(30X,1A4,14(3X,1A4))
    DO 600 I=1,ITER
    TOTAL=0
    DO 550 J=1,NOCAT
    TOTAL=TOTAL+DOTSUM(I,J)
    K=CATVEC(I)
    WRITE(6,1320) I,CATNAM(K),TOTAL,(DOTSUM(I,J),J=1,IEND)
1320 FORMAT(//,5X,I2,6X,1A4,15,2X,15(2X,15))
600 CONTINUE
602 IF (IEND.EQ.NOCAT)GO TO 650
    ISTRT=IEND+1
    IEND=NOCAT
    IF (IEND.GT. ISTRT+14) IEND=ISTRT+14
1350 WRITE(6,1350)
    FORMAT(//)
    WRITE(6,1300)
    WRITE(6,1305)
    WRITE(6,1310) (CATNAM(IJ),IJ=ISTRT,IEND)
    DO 610 I=1,ITER
    K=CATVEC(I)
    WRITE(6,1340) I,CATNAM(K),(DOTSUM(I,J),J=ISTRT,IEND)
1340 FORMAT(//,5X,I2,6X,1A4,7X,15(2X,15))
610 CONTINUE
    GO TO 602
C
GROUP LABELED CLUSTER ACCORDING TO CATEGORY
C
650 K = 0
    DO 510 I=1,NOCAT
    DO 510 J=1,NOSUB2
C
IF (CATVEC(J) .NE. I) GO TO 510
SUBNO(I) = SUBNO(I) + 1
K = K + 1
SUBVEC(K) = J
C
510 CONTINUE
C
RETURN
END

```

KNE 011530  
KNE 011540  
KNE 011550  
KNE 011560  
KNE 011570  
KNE 011580  
KNE 011590  
KNE 011600  
KNE 011610  
KNE 011620  
KNE 011630  
KNE 011640  
KNE 011650  
KNE 011660  
KNE 011670  
KNE 011680  
KNE 011690  
KNE 011700  
KNE 011710  
KNE 011720  
KNE 011730  
KNE 011740  
KNE 011750  
KNE 011760  
KNE 011770  
KNE 011780  
KNE 011790  
KNE 01800  
KNE 01810  
KNE 01820  
KNE 01830  
KNE 01840  
KNE 01850  
KNE 01860  
KNE 01870  
KNE 01880  
KNE 01890  
KNE 01900  
KNE 01910  
KNE 01920  
KNE 01930  
KNE 01940  
KNE 01950  
KNE 01960  
KNE 01970  
KNE 01980  
KNE 01990  
KNE 02000  
KNE 02010  
KNE 02020  
KNE 02030  
KNE 02040  
KNE 02050  
KNE 02060  
KNE 02070  
KNE 02080  
KNE 02090  
KNE 02100  
KNE 02110  
KNE 02120  
KNE 02130  
KNE 02140  
KNE 02150  
KNE 02160  
KNE 02170  
KNE 02180  
KNE 02190  
KNE 02200  
KNE 02210  
KNE 02220  
KNE 02230  
KNE 02240  
KNE 02250  
KNE 02260

FILE: LABDOT

```

C      SUBROUTINE LABDOT(DOTS)                                LAB00010
C      LABDOT UPDATES DOTFIL                                  LAB00020
C      IMPLICIT INTEGER (A-Z)                                LAB00030
C      INCLUDE CMRK15.LIST                                    LAB00040
C      COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60), LAB00050
C      SUBRAY(120),PTR(60),CATPTR(250),CATDOT(500),          LAB00060
C      DOTVEC(250),COND,MIX,PHOC,MAPKEY,DOTKEY,STATKY,      LAB00070
C      SUNANG,T,NEAPST,NIST,NOFEAT,FETVEC(30),OMAPUN,OMAPFI, LAB00080
C      OSAVTP,OSTAFT,NOSUN,ANGLE(A),SIZE,TOTDT2,FLODINF(6), LAB00090
C      CLSSYM(62),STADRS,MEANAD,TAHADP,MAPADR,SUNCOR(30),  LAB00100
C      ODOTUN,ODOTFI,MANSTA,MANDOT,OSPUNT,OSPFIL,OSPKEY,PRNSTS, LAB00110
C      PRNDOT,FLDNAM,VERTEX(22),NOVKT,NSUN,ANGLES(B)       LAB00120
C      ,TOTDT3,FLOADR,VTXADR                                 LAB00130
CSEND
C      DIMENSION DOTS(SIZE,1),CATNO(60)                     LAB00140
C      CHECK CATEGORY NAMES FOR NEW ENTRIES                 LAB00150
C      DO 100 I=1,NOCAT2                                     LAB00160
C      DO 90 J=1,NOCAT                                       LAB00170
C      IF (CATNM2(I) .EQ. CATNAM(J))GO TO 95                 LAB00180
C      CONTINUE                                              LAB00190
C      NOCAT = NOCAT + 1                                     LAB00200
C      INSERT NEW CATEGORY                                   LAB00210
C      CATNAM(NOCAT) = CATNM2(I)                             LAB00220
C      CATNO(I) = NOCAT                                       LAB00230
C      GO TO 100                                             LAB00240
C      95 CATNO(I) = J                                       LAB00250
C      100 CONTINUE                                          LAB00260
C      DO 150 I=1,NOCAT2                                     LAB00270
C      RETRIEVE BEGIN. AND END. POINTER                      LAB00280
C      IB = CATPTR(I) + 1                                     LAB00290
C      IE = IR + CATDOT(IB-1) - 1                           LAB00300
C      DO 120 J=IB,IE                                       LAB00310
C      RETRIEVE DOT NO AND CHANGE CATEGORY NO FOR DOT     LAB00320
C      K = CATDOT(J)                                         LAB00330
C      120 DOTS(4,K) = CATNO(I)                              LAB00340
C      150 CONTINUE                                          LAB00350
C      RETURN                                               LAB00360
C      END                                                  LAB00370
C      LAB00380
C      LAB00390
C      LAB00400
C      LAB00410
C      LAB00420
C      LAB00430
C      LAB00440
C      LAB00450
C      LAB00460
C      LAB00470
C      LAB00480
C      LAB00490
C      LAB00500
C      LAB00510
C      LAB00520
C      LAB00530
C      LAB00540
C      LAB00550

```

ORIGINAL PAGE IS  
OF POOR QUALITY

~~18-23~~  
354

FILE LABLR

```

C      SURROUTINE LABLR (ARRAY, TOP, NOFLD, TOTVRT, FLDSAV, VERTX, MEANS, EXITT) LAB00010
C      IMPLICIT INTEGER (A-Z) LAB00020
C      LIMIT = 3135 LAB00030
C      INCLUDE COMBK1.LIST LAB00040
C      INCLUDE COMBK6.LIST LAB00050
C      INCLUDE COMBK15.LIST LAB00060
C      COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2, LAB00070
C      AVAR2, COVAR2, CLSID2, SURN02, SUBDS2, FLOSV2, VERTX2, LAB00080
C      FETVC2(30), SURVC2(75), SURPTR(75), CLSVC2(60), LAB00090
C      KEPPYS(60), NOGRP, GRPNAM(60), GRPDEX(61), LAB00100
C      GRPCHK(61), GROUPS(124) LAB00110
C      COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPP, SAVTAP, BMFILE, BMKEY, LAB00120
C      HISFIL, HISKEY, TRFOHM, ERIPTP, ERPKEY, MAPUNT, NOFILE, LAB00130
C      DRUMAD, DRHMDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL LAB00140
C      NHSTUN, NHSTFI, SCTRUN, MAPFIL LAB00150
C      DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT, LAB00160
C      CRDUNT, PRTUNT, RANDIO LAB00170
C      COMMON /LABS/NOCAT, CATNAM(60), NOCL2, CLSNM2(60), NOCAT2, CATNM2(60), LAB00180
C      SUBRAY(120), PTR(60), CATPTR(250), CATDOT(500), LAB00190
C      DOTVEC(250), COND, MIX, PROC, MAPKEY, DOTKEY, STATKY, LAB00200
C      SUNANG, T, NEARST, DIST, NOFEAT, FETVEC(30), OMAPUN, OMAPP1, LAB00210
C      OSAVTP, OSTAFI, NOSUN, ANGLE (R), SIZE, TOTDT2, FLDINF(6), LAB00220
C      CLSSYM(62), STADRS, MEANAD, TABADR, MAPADR, SUNCOR(30), LAB00230
C      ODOTUN, ODOTFI, MANSTA, MANDOT, DSPUNT, DSPFIL, DSPKEY, PRNSTS, LAB00240
C      PRNDOT, FLDNAM, VERTEX(22), NOVRT, NSUN, ANGLES(8) LAB00250
C      TOTDT3, FLDAOR, VTXAOR LAB00260
C      CSEND LAB00270
C      DIMENSION FLOSAV(4,1), VERTX(2,1), ARRAY(1), SUBVEC(60), SUBNOS(60) LAB00280
C      DIMENSION N(60), CATVEC(60), SUBNAM(60), CNDSUB(60), MIXSUB(60) LAB00290
C      REAL MEANS(NOFET2,1) LAB00300
C      DIMENSION DOTVC2(250), DOTSUM(60,60) LAB00310
C      DIMENSION TABLE(3135) LAB00320
C      DATA FLANK/' '/ LAB00330
C      SWITCH = 2 LAB00340
C      INITIALIZE DOTSUM LAB00350
C      DO 10 I=1,60 LAB00360
C      DO 10 J=1,60 LAB00370
C      DOTSUM(I,J)=0 LAB00380
C      CONTINUE LAB00390
C      MANUALLY RELABEL STATS LAB00400
C      IF (MANSTA .EQ. 0) GO TO 20 LAB00410
C      CALL MANORD (ARRAY (CLSID2), CLSVC2, SUBVEC, NOCLS2, SUBNOS, NOSUB2) LAB00420
C      UPDATE INFO IN ARRAY LAB00430
C      CALL REORDER (ARRAY, SUBVEC, N) LAB00440
C      OUTPUT REORDERED STATS LAB00450
C      CALL LABMAN (OSAVTP, OSTAFI, NOCLS2, NOSUR2, NOFET2, NOFLD2, TOTVT2, LAB00460
C      FETVC2, ARRAY (FLDSV2), ARRAY (VERTX2), ARRAY (CLSID2), LAB00470
C      SUBNOS, ARRAY (SUBDS2), N, STADRS, VARSZ2, PUNCH, SUBVEC, PRNSTS, SWITCH) LAB00480
C      20 IF (MANDOT .EQ. 0) GO TO 40 LAB00490
C      UPDATE DOTFIL LAB00500
C      CALL LABDOT (ARRAY (COVAR2)) LAB00510
C      OUTPUT UPDATE DOTFIL LAB00520
C      CALL WRTDOT (TOTDT2, NOSUN, FLDSAV, VERTX, ANGLE, ARRAY (COVAR2), LAB00530
C      NOCAT, CATNAM, SIZE, NOFEAT, FETVEC, TOTVRT, NOFLD, LAB00540
C      ODOTUN, ODOTFI) LAB00550
C      IF (PHNDOT .EQ. 1) GOTO 798 LAB00560
C      30 CONTINUE LAB00570
C      EXECUTING A PROEDURE LAB00580

```



FILE LABLR

```

C
70  CATNO=NOCAT
    CALL NAMSTA(SURNAM,CATVEC,SUBNOS,NOSUR2,CATNAM,NOCAT)
    IF(CATNO.EQ.NOCAT)GO TO 75
    IF(EXITT.EQ.0)GO TO 75
C
    USER WISHES TO EXIT IF ANY CLASSES NOT USED BY LABEL
C
1000 WRITE(6,1000)NOCAT,CATNO
    FORMAT(//,1X,I3,' LABELS REMAINING OF ',I3,'. EXIT TAKEN')
    CALL EXIT
C
    OUTPUT LABELED STATS
75  CALL LABMAN(OSAVTP,OSTAFI,NOCAT,NOSUB2,NOFET2,NOFLD2,TOTVT2,
    * FETVC2,ARRAY(FLOSV2),ARRAY(VRTX2),CATNAM,SUBNOS,SUBNAM,
    * KEPPTS,STADRS,VAR22,PUNCH,SUBVEC,PRNSTS,SWTCH)
    IF(COND.EQ.0)GO TO 90
C
    FLAG CONDITIONAL CLUSTERS
    CALL CNDMAP(ARRAY(COVAR2),CNDSUB,CATVEC)
    OUTPUT CONDITIONAL MAP
C
    CALL CLSMAP(CNDSUB,1,SURNOS,SUBVEC,SUBNAM,CATVEC)
90  IF(MIX.EQ.0)GO TO 100
    IF(COND.NE.0)OMAPFI = OMAPFI + 1
C
    FLAG MIXED CLUSTERS
    CALL MIXMAP(ARRAY(COVAR2),MIXSUB,NOSUB2,CATVEC)
    OUTPUT MIXED MAP
    CALL CLSMAP(MIXSUB,2,SURNOS,SURVEC,SUBNAM,CATVEC)
    OUTPUT DISPLAY INTERFACE TAPE--MAPTAP
C
100 IF(DSPKEY.EQ.0)GO TO 110
    CALL RHEAD(FLOAD?,FLOSAV,TOTWDS,ISTAT)
103 IF(ISTAT.EQ.1)GO TO 103
    CALL RHEAD(VTXADR,VRTX,TOTVTS,ISTAT)
105 IF(ISTAT.EQ.1)GO TO 105
C
    CALL DSPTAP(SURNOS,SUBNAM,FLOSAV,VRTX,CATVEC,SUBVEC,MEANS,
    * ARRAY(COVAR2),TOP,ARRAY,NOFLD,TOTVRT)
110 CONTINUE
    RETURN
C
    CODE TO PRINT DOT DATA RECORD
C
798 CONTINUE
    ISTART=1
    IEND=10
799 CONTINUE
    IKT=0
    DO 800 II=1,TOTD;2
    IKT=IKT+1
    IF(IEND.GT.NOFEAT) IEND=NOFEAT
    IF(II.NE.1.AND.IKT.EQ.1)WRITE(6,810)
810 FORMAT(1H1,5(/))
    IF(IKT.NE.1)GO TO 820
    WRITE(6,700)
700 FORMAT(//)
    WRITE(6,690)
690 FORMAT(1X,'NO.',2X,'SAMPLE',2X,'LINE',2X,'TYPE',2X,'CATEGORY',
    3X,'DATA')
    WRITE(6,720)(BLANK,FETVC2(I),I=ISTART,IEND)
720 FORMAT(37X,10(A1,'CH',I2,''))
820 CONTINUE

```

LAB01530  
LAB01540  
LAB01550  
LAB01560  
LAB01570  
LAB01580  
LAB01590  
LAB01600  
LAB01610  
LAB01620  
LAB01630  
LAB01640  
LAB01650  
LAB01660  
LAB01670  
LAB01680  
LAB01690  
LAB01700  
LAB01710  
LAB01720  
LAB01730  
LAB01740  
LAB01750  
LAB01760  
LAB01770  
LAB01780  
LAB01790  
LAB01800  
LAB01810  
LAB01820  
LAB01830  
LAB01840  
LAB01850  
LAB01860  
LAB01870  
LAB01880  
LAB01890  
LAB01900  
LAB01910  
LAB01920  
LAB01930  
LAB01940  
LAB01950  
LAB01960  
LAB01970  
LAB01980  
LAB01990  
LAB02000  
LAB02010  
LAB02020  
LAB02030  
LAB02040  
LAB02050  
LAB02060  
LAB02070  
LAB02080  
LAB02090  
LAB02100  
LAB02110  
LAB02120  
LAB02130  
LAB02140  
LAB02150  
LAB02160  
LAB02170  
LAB02180  
LAB02190  
LAB02200  
LAB02210  
LAB02220  
LAB02230  
LAB02240  
LAB02250  
LAB02260  
LAB02270  
LAB02280

FILE LABLR

	IDUM=COVAR2*(II-1)*SIZE-1	LAB02290
	IDMM=COVAR2*(II-1)*SIZE+3	LAB02300
	WRITE(6,710)II,(ARRAY(IDUM+I),I=1,4),(ARRAY(IDMM+JJ),JJ=ISTART,IEND	LAB02310
	*D)	LAB02320
710	FORMAT(1X,I3,1H,.,3X,I4,3X,I4,2X,I2,6X,I2,8X,10(I3,4X))	LAB02330
	WRITE(6,712)	LAB02340
712	FORMAT( )	LAB02350
800	CONTINUE	LAB02360
	IF(NOFEAT.GT.10) GO TO 830	LAB02370
	GO TO 840	LAB02380
830	CONTINUE	LAB02390
	IF(ITWO.EQ.1) GO TO 840	LAB02400
	ITWO=1	LAB02410
	ISTART=IEND+1	LAB02420
	IEND=NOFEAT	LAB02430
	GO TO 799	LAB02440
840	CONTINUE	LAB02450
	GOTO 30	LAB02460
	END	LAB02470

ORIGINAL PAGE IS  
OF POOR QUALITY



FILE: MANORD

C  
END

MAN00800  
MAN00810

ORIGINAL PAGE IS  
OF POOR QUALITY

~~18-29~~  
360



FILE: MAPHND

```
C THIS ROUTINE PRINTS THE HEADER INFORMATION FOR THE CLASSIFICATION MAP IN CLASSIFY AND DISPLAY MAP00010
C MAP00020
C MAP00030
C MAP00040
C MAP00050
C MAP00060
C MAP00070
C MAP00080
C MAP00090
C MAP00100
C MAP00110
C MAP00120
C MAP00130
C MAP00140
C MAP00150
C MAP00160
C MAP00170
C MAP00180
C MAP00190
C MAP00200
C MAP00210
C MAP00220
C MAP00230
C MAP00240
C MAP00250
C MAP00260
C MAP00270
C MAP00280
C MAP00290
C MAP00300
C MAP00310
C MAP00320
C MAP00330
C MAP00340
C MAP00350
C MAP00360
C MAP00370
C MAP00380
C MAP00390
C MAP00400
C MAP00410
C MAP00420
C MAP00430
C MAP00440
C MAP00450
C MAP00460
C MAP00470
C MAP00480
C MAP00490
C MAP00500
C MAP00510
C MAP00520
C MAP00530
C MAP00540
C MAP00550
C MAP00560
C MAP00570

SURROUTINE MAPHND(INOCAT,CLSSYM,CATNAM,KATNO,SURDES,CATSUB)
  NOCAT -- NO. OF CATEGORIES
  CLSSYM -- SYMBOLS FOR CATEGORIES OR SURCLASSES
  CATNAM -- CATEGORY NAMES
  KATNO -- CATEGORY EACH CLASS WAS ASSIGNED TO
  CLSMTX -- CLASS NAMES
  SURNO -- NO. OF SURCLASSES IN EACH CLASS
  SURDES -- SURCLASS NAMES
  CLSVC2 -- CLASS EACH SUBCLASS WAS ASSIGNED TO (IN COMMON BLOCK INFORM)

  IMPLICIT INTEGER (A-Z)

  INCLUDE COMAK1,LIST
  COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VAR5Z2,TOTVT2,NOFLD2,
  * AVAR2,COVAR2,CLSID2,SURNO2,SURDS2,FLDSV2,VERTX2,
  * FETVC2(30),SUVVC2(75),SURPTR(75),CLSVC2(60),
  * KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
  * GRPCHK(61),GROUPS(124)

CSEND
  LOGICAL ISWTH
  DIMENSION CLSSYM(1),CATNAM(1),KATNO(1),SUBDES(1)
  DIMENSION CATSUB(1)

  PRINTS CATEGORY CLASSIFIER INFORMATION

  WRITE(6,200)
  200 FORMAT(//T54,'* LABELED CLUSTER MAP://T49,'( ** - DENOTES MIXED/COND
  * CLUSTER )//T50,' * - DENOTES DU/DU AREA',9X,')//T33,'LAREL',
  * T41,'SURCLASS//T31,'NO.',T35,'NAME',T64,'UNLABELED NO.',T40,
  * 'LABELED NO.',T94,'NAME',T101,'SYMBOL')

  JJ = 0
  KK = 0
  DO 68 I=1,NOCAT

  210 WRITE(6,210)I,CATNAM(I)
  FORMAT(/T31,I2,T37,A4)
  ISWTH = .TRUE.
  DO 63 J=1,NOSUR2
  IF (KATNO(J) .EQ. I) GO TO 64
  GO TO 63
  64 KK = KK + 1
  IF (ISWTH) GO TO 72
  WRITE(6,250)J,KK,SURDES(KK),CLSSYM(J)
  250 FORMAT( T69,I2,T84,I2,T94,A4,T105,A1)
  GO TO 75
  72 WRITE(6,240)J,KK,SURDES(KK),CLSSYM(J)
  240 FORMAT(1H+,T69,I2,T84,I2,T94,A4,T105,A1)
  ISWTH = .FALSE.
  75 IF (CATSUB(J) .GT. (NOCAT + 2)) WRITE(6,80)
  80 FORMAT(1H+,T92,2H**)
  63 CONTINUE
  68 CONTINUE
  RETURN
  END
```



FILE: REORDER

```

C      SURROUTINE REORDER (ARRAY, SURVEC, N)
C      REORDER ID INFORMATION IN ARRAY
C      IMPLICIT INTEGER (A-Z)
C      INCLUDE COMPK1.LIST
COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
*      AV4R2, COVAR2, CLSID2, SURNO2, SURDS2, FLDSV2, VERTX2,
*      FETVC2(30), SUHVC2(75), SUHPTR(75), CLSVC2(60),
*      KEPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),
*      GRPCHK(61), GROUPS(124)
C$END
DIMENSION ARRAY(1), SUBNAM(60), N(1)
DIMENSION SURVEC(1)
C      UPDATE SURCLASS NAMES
DO 100 I=1, NOSUR2
  K = SURVEC(I)
100  SURNAM(I) = ARRAY(SUBDS2 + K-1)
DO 110 I=1, NOSUR2
110  ARRAY(SUBDS2 + I-1) = SUBNAM(I)
C      UPDATE POPULATION ARRAY
DO 130 I=1, NOSUR2
  K = SURVEC(I)
130  N(I) = KEPPTS(K)
C      RETURN
C      END
RE000010
RE000020
RE000030
RE000040
RE000050
RE000060
COM00010
COM00020
COM00030
COM00040
COM00050
RE000080
RE000090
RE000100
RE000110
RE000120
RE000130
RE000140
RE000150
RE000160
RE000170
RE000180
RE000190
RE000200
RE000210
RE000220
RE000230
RE000240
RE000250
RE000260
RE000270
RE000280
RE000290
```

FILE SET14

```

SUBROUTINE SET14 (ARRAY, TOP, EXIT)
C*
C* SET14 READS IN THE CONTROL CARDS FOR THE LABEL PROCESSOR
C*
C* IMPLICIT INTEGER (A-Z)
C* DIMENSION CODE (21), CARD (62), EQUOM (3)
C* DATA CODE / 'CHAN', 'DATA', 'MAPF', 'DOTF', 'OPTI', 'EXCL',
C* 'STAT', 'DOTL', 'STAL', 'DIST', 'THRE', 'NEAR', 'PROC', 'MODU',
C* 'SUNA', 'MAPT', 'DATE', 'COMM', 'HED1', 'HED2', 'END' /
C IN THE CONVERSION FROM 1108 TO IBM, STATLA BECOMES STAL
C DIMENSION ARRAY(1), ACARD(20)
C DIMENSION SLASH(2), EQUVEC(2)
C DATA SLASH/1, '/'/
C DATA EQUVEC/1, '= '/
C DATA EQUOM/2, '=, ., , /
C DATA CODE/1, '/
C*
C* INCLUDE COMRK1, LIST
C* INCLUDE COMRK4, LIST
C* INCLUDE COMRK6, LIST
C* INCLUDE COMRK15, LIST
C* COMMON / INFORM / NOCL2, NOSUB2, NOFET2, VARSZ, TOTVT2, NOFLD2,
C* AVAR2, COVAR2, CLS102, SURNO2, SUBDS2, FLDSV2, VERTX2,
C* FETVC2(30), SURVC2(75), SURPTR(75), CLSVC2(60),
C* KEPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),
C* GRPCHK(61), GROUPS(124)
C* DIMENSION HED1(15), HED2(15), DATE(3), COMENT(15)
C* EQUIVALENCE (HED1(1), HEAD(4)), (DATE(1), HEAD(22)),
C* (HED2(1), HEAD(30)), (COMENT(1), HEAD(48))
C* COMMON / GLOHAL / HEAD(63), MAPTAP, DATAP, SAVTAP, RMFILE, RMKEY,
C* HISFIL, HISKEY, TRFORM, FRIP, ERKEY, MAPUNT, NOFILE,
C* DRUMAD, DRMWD, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
C* NHSTUN, NHSTFI, SCTRUN, MAPFIL
C* DOTUNT, DOTFIL, NCHPAS, TRNSFL, RMTRFL, HISTFL, PCHUNT,
C* CHDUNT, PRTUNT, RANDIO
C* COMMON / LAHS / NOCAT, CATNAM(60), NOCL2, CLSNM2(60), NOCAT2, CATNM2(60),
C* SURRAY(120), PTR(60), CATPTR(250), CATDOT(500),
C* DOTVEC(250), COND, MIX, PROC, MAPKEY, DOTKEY, STATKY,
C* SUNANG, T, NEARST, DIST, NOFEAT, FETVEC(30), OMAPUN, OMAPFI,
C* OSAVT, OSTAFI, NOSUN, ANGLE(8), SIZE, TOTDT2, FLDINF(6),
C* CLSSYM(62), STADRS, MEANAD, TABADR, MAPADR, SUNCOR(30),
C* ODOTUN, ODOTFI, MANSTA, MANDOT, DSPUNT, DSPFIL, DSPKEY, PRNSTS,
C* PRNDOT, FLDNAM, VERTEX(22), NOVPT, NSUN, ANGLE(8)
C* , TOTDT3, FLDADR, VTXADR
C$END
REAL T, SUNCOR
DATA BLANK / ' ', SBCD / 'S', DRCD / 'D', URCD / 'U',
* FRCD / 'F', IRCD / 'I', ORCD / 'O', COMMA / ',', CBCD / 'C',
* MRCD / 'M', BCD1 / '1', BCD2 / '2', KRCD / 'K', ABCD / 'A'
C DATA EBCD / 'E'
C INITIALIZE PARAMETERS
C
NPUT = 21
TOTDT3 = 0
COND = 0
MIX = 0
PROC = 1
PRNDOT = 0
PRNSTS = 0
MANSTA = 0
MANDOT = 0
MAPKEY = 0
DOTKEY = 0
DSPKEY = 0
STATKY = 0
NOSUN = 0
SUNANG = 0
T = 25.0
NEARST = 1
DIST = 1
NOFET2 = 0
NOFEAT = 0
PTR1 = 0
PTR2 = 0
NOCAT2 = 0
NOCL2 = 0
EXIT = 0

```

SET00010  
 SET00020  
 SET00030  
 SET00040  
 SET00050  
 SET00060  
 SET00070  
 SET00080  
 SET00090  
 SET00100  
 SET00110  
 SET00120  
 SET00130  
 SET00140  
 SET00150  
 SET00160  
 SET00170  
 SET00180  
 SET00190  
 SET00200  
 SET00210  
 SET00220  
 SET00230  
 SET00240  
 SET00250  
 SET00260  
 SET00270  
 SET00280  
 SET00290  
 SET00300  
 SET00310  
 SET00320  
 SET00330  
 SET00340  
 SET00350  
 SET00360  
 SET00370  
 SET00380  
 SET00390  
 SET00400  
 SET00410  
 SET00420  
 SET00430  
 SET00440  
 SET00450  
 SET00460  
 SET00470  
 SET00480  
 SET00490  
 SET00500  
 SET00510  
 SET00520  
 SET00530  
 SET00540  
 SET00550  
 SET00560  
 SET00570  
 SET00580  
 SET00590  
 SET00600  
 SET00610  
 SET00620  
 SET00630  
 SET00640  
 SET00650  
 SET00660  
 SET00670  
 SET00680  
 SET00690  
 SET00700  
 SET00710  
 SET00720  
 SET00730  
 SET00740  
 SET00750  
 SET00760

ORIGINAL PAGE IS  
OF POOR QUALITY



FILE SET14

```

210 M = NXTCHR(CARD,COL)
211 IF (M.EQ. IHCD) GO TO 213
    IF (M.EQ. OHCD) GO TO 220
    IF (M.EQ. BLANK) GO TO 105
212 WRITE(6,216)
216 FORMAT(' ERROR ON MAPFIL CARD')
    GO TO 105
C
213 J = FIND12(CARD,COL,SLASH)
    IF (J.EQ. -1) GO TO 212
    MAPKEY = 1
214 M = NXTCHR(CARD,COL)
    IF (M.EQ. COMMA) GO TO 214
    IF (M.EQ. FRCD) GO TO 215
    IF (M.EQ. URCD) GO TO 211
    J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 212
    M = NUMBER(CARD,COL,MAPUNT,ZERO)
    COL = COL - 1
    GO TO 214
215 J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 212
    M = NUMBER(CARD,COL,MAPFIL,ZERO)
    COL = COL - 1
    MAPFIL = MAPFIL - 1
    GO TO 214
220 J = FIND12(CARD,COL,SLASH)
    IF (J.EQ. -1) GO TO 212
221 M = NXTCHR(CARD,COL)
    IF (M.EQ. COMMA) GO TO 221
    IF (M.EQ. FRCD) GO TO 222
    IF (M.EQ. URCD) GO TO 211
    J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 212
    M = NUMBER(CARD,COL,OMAPUN,ZERO)
    COL = COL - 1
    GO TO 221
222 J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 212
    M = NUMBER(CARD,COL,OMAPFI,ZERO)
    COL = COL - 1
    OMAPFI = OMAPFI - 1
    GO TO 221
C
    DOTFIL CARD
C
250 M = NXTCHR(CARD,COL)
251 IF (M.EQ. IHCD) GO TO 254
    IF (M.EQ. OHCD) GO TO 260
    IF (M.EQ. BLANK) GO TO 105
252 WRITE(6,253)
253 FORMAT(' ERROR ON MAPFILE CARD')
    GO TO 105
C
254 J = FIND12(CARD,COL,SLASH)
    IF (J.EQ. -1) GO TO 252
    DOTKEY = 1
256 M = NXTCHR(CARD,COL)
    IF (M.EQ. COMMA) GO TO 256
    IF (M.EQ. FRCD) GO TO 258
    IF (M.EQ. URCD) GO TO 251
    J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 252
    M = NUMBER(CARD,COL,DOTUNT,ZERO)
    COL = COL - 1
    GO TO 256
258 J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 252
    M = NUMBER(CARD,COL,DOTFIL,ZERO)
    COL = COL - 1
    DOTFIL = DOTFIL - 1
    GO TO 256
C
260 J = FIND12(CARD,COL,SLASH)
    IF (J.EQ. -1) GO TO 252
262 M = NXTCHR(CARD,COL)

```

SET01530  
 SET01540  
 SET01550  
 SET01560  
 SET01570  
 SET01580  
 SET01590  
 SET01600  
 SET01610  
 SET01620  
 SET01630  
 SET01640  
 SET01650  
 SET01660  
 SET01670  
 SET01680  
 SET01690  
 SET01700  
 SET01710  
 SET01720  
 SET01730  
 SET01740  
 SET01750  
 SET01760  
 SET01770  
 SET01780  
 SET01790  
 SET01800  
 SET01810  
 SET01820  
 SET01830  
 SET01840  
 SET01850  
 SET01860  
 SET01870  
 SET01880  
 SET01890  
 SET01900  
 SET01910  
 SET01920  
 SET01930  
 SET01940  
 SET01950  
 SET01960  
 SET01970  
 SET01980  
 SET01990  
 SET02000  
 SET02010  
 SET02020  
 SET02030  
 SET02040  
 SET02050  
 SET02060  
 SET02070  
 SET02080  
 SET02090  
 SET02100  
 SET02110  
 SET02120  
 SET02130  
 SET02140  
 SET02150  
 SET02160  
 SET02170  
 SET02180  
 SET02190  
 SET02200  
 SET02210  
 SET02220  
 SET02230  
 SET02240  
 SET02250  
 SET02260  
 SET02270  
 SET02280

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE SET14

```

IF (M.EQ.COMMA)GO TO 262
IF (M.EQ.FRCD) GO TO 264
IF (M.NE.UHCD) GO TO 251
J = FIND12(CARD,COL,EQUVEC)
IF (J.EQ.-1) GO TO 252
M = NUMBER(CARD,COL,ODOTUN,ZERO)
COL = COL - 1
GO TO 262
264 J = FIND12(CARD,COL,EQUVEC)
IF (J.EQ.-1) GO TO 252
M = NUMBER(CARD,COL,ODOTFI,ZERO)
COL = COL - 1
ODOTFI = ODOTFI - 1
GO TO 262

CCC
OPTION CARD
330 M = NXTCHR(CARD,COL)
IF (M.EQ.BLANK)GO TO 105
IF (M.EQ.CBCD) GO TO 340
IF (M.EQ.SRCD) GO TO 345
IF (M.EQ.DRCD) GO TO 350
IF (M.EQ.MBCD) GO TO 363
IF (M.EQ.EHCD)GO TO 355
333 WRITE(6,335)
335 FORMAT(' ERROR ON OPTION CARD')
GO TO 105
340 COND = 1
GO TO 360
345 PRNSTS = 1
GO TO 360
350 PRNDOT = 1
GO TO 360
355 EXIT=1
GO TO 360
363 MIX = 1
360 J = FIND12(CARD,COL,EQUCOM)
IF (J.EQ.3) GO TO 330
IF (J.EQ.-1) GO TO 105

CCC
EXCLUDE CARD
365 TOTDT3 = NUMBER(CARD,COL,DOTVEC,TOTDT3)
CALL ORDER(DOTVEC,TOTDT3)
GO TO 105

CCC
STATFILE CARD
370 M = NXTCHR(CARD,COL)
371 IF (M.EQ.IBCD) GO TO 374
IF (M.EQ.OCDD) GO TO 380
IF (M.EQ.BLANK)GO TO 105
372 WRITE(6,373)
373 FORMAT(' ERROR ON STATF1 CARD')
GO TO 105

C
374 J = FIND12(CARD,COL,SLASH)
IF (J.EQ.-1) GO TO 372
STATKY = 1
375 M = NXTCHR(CARD,COL)
IF (M.EQ.CCMA)GO TO 375
IF (M.EQ.FBCD) GO TO 376
IF (M.NE.UHCD) GO TO 370
J = FIND12(CARD,COL,EQUVEC)
IF (J.EQ.-1) GO TO 372
M = NUMBER(CARD,COL,SAVTAP,ZERO)
COL = COL - 1
GO TO 375
376 J = FIND12(CARD,COL,EQUVEC)
IF (J.EQ.-1) GO TO 372
M = NUMBER(CARD,COL,STAFIL,ZERO)
COL = COL - 1
STAFIL = STAFIL - 1
GO TO 375
380 J = FIND12(CARD,COL,SLASH)

```

```

SET02290
SET02300
SET02310
SET02320
SET02330
SET02340
SET02350
SET02360
SET02370
SET02380
SET02390
SET02400
SET02410
SET02420
SET02430
SET02440
SET02450
SET02460
SET02470
SET02480
SET02490
SET02500
SET02510
SET02520
SET02530
SET02540
SET02550
SET02560
SET02570
SET02580
SET02590
SET02600
SET02610
SET02620
SET02630
SET02640
SET02650
SET02660
SET02670
SET02680
SET02690
SET02700
SET02710
SET02720
SET02730
SET02740
SET02750
SET02760
SET02770
SET02780
SET02790
SET02800
SET02810
SET02820
SET02830
SET02840
SET02850
SET02860
SET02870
SET02880
SET02890
SET02900
SET02910
SET02920
SET02930
SET02940
SET02950
SET02960
SET02970
SET02980
SET02990
SET03000
SET03010
SET03020
SET03030
SET03040

```

FILE SET14

```

IF (J.EQ.-1) GO TO 372
381 M = NXTCHR(CARD,COL)
    IF (M.EQ.COMMA) GO TO 381
    IF (M.EQ.FHCD) GO TO 382
    IF (M.NE.UHCD) GO TO 370
    J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ.-1) GO TO 372
    M = NUMBER(CARD,COL,OSAVTP,ZERO)
    COL = COL - 1
    GO TO 381
382 J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ.-1) GO TO 372
    M = NUMBER(CARD,COL,OSTAFI,ZERO)
    COL = COL - 1
    OSTAFI = OSTAFI - 1
    GO TO 381
C
C DOTLABEL CARD
390 M = CRDSCN(CARD,CATPTR,CATNM2,CATDOT,NOCAT2,PTR1)
    MANDOT = 1
    GO TO 105
C
C STATLABEL CARD
400 M = CRDSCN(CARD,PTR,CLSNM2,SUBRAY,NOCL2,PTR2)
    MANSTA = 1
    GO TO 105
C
C DISTANCE
410 M = NXTCHR(CARD,COL)
    M = NXTCHR(CARD,COL)
    IF (M.EQ.HCD1) DIST = 1
    IF (M.EQ.RCD2) DIST = 2
    GO TO 105
C
C THRESHOLD CARD
420 M = FLTNUM(CARD,COL,T,1)
    GO TO 105
C
C K NEAREST DOTS
430 J = NUMBER(CARD,COL,NEARST,ZERO)
    GO TO 105
C
C PROCEDURE CARD
K= NEAREST PROCEDURE = 1
ALL-OF-A-KIND = 2
MANUAL LABELING OF FILES = 3
440 M = NXTCHR(CARD,COL)
    IF (M.EQ.KRCD) GO TO 446
    IF (M.EQ.AHCD) GO TO 448
    IF (M.EQ.MBCD) GO TO 450
    WRITE(6,445)
445 FORMAT(' ERROR ON PROCEDURE CARD')
    GO TO 105
446 PROC = 1
    GO TO 105
448 PROC = 2
    GO TO 105
450 PROC = 3
    GO TO 105
C
C MODULE CARD DECK
460 CALL CRUSTA(ARRAY, TOP)
    STATKY = 1
    GO TO 105
C
C SUN ANGLE CARD

```

```

SET03050
SET03060
SET03070
SET03080
SET03090
SET03100
SET03110
SET03120
SET03130
SET03140
SET03150
SET03160
SET03170
SET03180
SET03190
SET03200
SET03210
SET03220
SET03230
SET03240
SET03250
SET03260
SET03270
SET03280
SET03290
SET03300
SET03310
SET03320
SET03330
SET03340
SET03350
SET03360
SET03370
SET03380
SET03390
SET03400
SET03410
SET03420
SET03430
SET03440
SET03450
SET03460
SET03470
SET03480
SET03490
SET03500
SET03510
SET03520
SET03530
SET03540
SET03550
SET03560
SET03570
SET03580
SET03590
SET03600
SET03610
SET03620
SET03630
SET03640
SET03650
SET03660
SET03670
SET03680
SET03690
SET03700
SET03710
SET03720
SET03730
SET03740
SET03750
SET03760
SET03770
SET03780
SET03790
SET03800

```

ORIGINAL IS  
OF GREATER QUANTITY

18-5T  
368





FILE SET14

```

C
C   PROCEDURE 1 AND 2 CANNOT BE SELECTED ALONG WITH PROCEDURE 3
C   IF (PROC.EQ. 3) GO TO 570
C   IF (NOFET2.EQ. 0 OR NOFEAT.EQ. 0) GO TO 570
C   IF (NOFET2.EQ. NOFEAT) GO TO 570
C   WRITE(6,2050)
C 570 IF (MANDOT.EQ. 0) GO TO 580
C     IF (PROC.EQ. 3) GO TO 580
C     WRITE(6,2060)
C     PROC. = 3
C 580 IF (MANSTA.EQ. 0) GO TO 590
C     IF (PROC.EQ. 3) GO TO 590
C     WRITE(6,2060)
C     PROC = 3
C 590 CONTINUE
C     WRITE(6,1000)
C     IF (NOFET2.NE. 0) WRITE(6,1010) (FETVC2(I),I=1,NOFET2)
C     IF (NOFEAT.NE. 0) WRITE(6,1020) (FETVEC(I),I=1,NOFEAT)
C     IF (MAPKEY.EQ. 1) WRITE(6,1030)
C     IF (MANDOT.EQ. 1) WRITE(6,1040)
C     IF (MANSTA.EQ. 1) WRITE(6,1050)
C     IF (DSPKEY.EQ. 1) WRITE(6,1055)
C     IF (PROC.EQ. 1) WRITE(6,1060)
C     IF (PROC.EQ. 2) WRITE(6,1070)
C     IF (DIST.EQ. 1) WRITE(6,1080)
C     IF (DIST.EQ. 2) WRITE(6,1090)
C     WRITE(6,2000)
C     WRITE(6,2010) NFAWST
C     IF (SUNANG.EQ.1) WRITE(6,2020)
C     IF (SUNANG.EQ.2) WRITE(6,2030)
C     IF (SUNANG.EQ.0) WRITE(6,2040)
C     IF (DOTKEY.EQ. 1) WRITE(6,2070)
C     IF (STATKY.EQ. 1) WRITE(6,2080)
C     IF (COND.EQ. 1) WRITE(6,2090)
C     IF (MIX.EQ. 1) WRITE(6,3000)
C     IF (PRNDOT.EQ.1) WRITE(6,3020)
C     IF (PRNSTS.EQ.1) WRITE(6,3030)
C     IF (EXIT.EQ.1) WRITE(6,3040)
C
C     IERR = 0
C     IF (PROC.EQ. 3) GO TO 600
C     IF (STATKY.NE. 1) IERR = 1
C     IF (DOTKEY.NE. 1) IERR = 1
C     IF (COND.EQ. 1 OR MIX.EQ. 1) GO TO 594
C 593 IF (PROC.EQ. 1) GO TO 595
C 594 IF (MAPKEY.NE. 1) IERR = 1
C 595 IF (IERR.EQ. 1) WRITE(6,3010)
C 600 CONTINUE
C
C 1000 FORMAT(/' USER HAS REQUESTED THE FOLLOWING OPTIONS :'/)
C 1010 FORMAT(' STAT CHANNELS ARE',30(12,1X))
C 1020 FORMAT(' DOTDATA CHANNELS ARE',30(12,1X))
C 1030 FORMAT(' CLUSTER/CLASSIFICATION TAPE IS BEING INPUT')
C 1040 FORMAT(' DOTFIL WILL BE RELABELED')
C 1050 FORMAT(' STAT FILE WILL BE RELABELED')
C 1055 FORMAT(' MAPTAP FILE WILL BE OUTPUT')
C 1060 FORMAT(' K-NEAREST PROCEDURE WILL BE USED')
C 1070 FORMAT(' ALL-OF-A-KIND PROCEDURE WILL BE USED')
C 1080 FORMAT(' L1 DISTANCE WILL BE USED')
C 1090 FORMAT(' L2 DISTANCE WILL BE USED')
C 2000 FORMAT(' THRESHOLD DISTANCE = ',F10,3)
C 2010 FORMAT(' 1X,13, *-NEAREST DOTS WILL BE USED')
C 2020 FORMAT(' SUN ANGLES WILL BE EXTRACTED FROM DOTFIL')
C 2030 FORMAT(' SUN ANGLES WILL BE READ IN FROM CARDS')
C 2040 FORMAT(' NO SUN ANGLE CORRECTION WILL BE APPLIED')
C 2050 FORMAT(/' NO. OF STAT CHANNEL AND DOT DATA CHANNELS MUST BE EQUAL')
C 2060 FORMAT(/' A LABELING PROCEDURE MAY NOT BE CHOSEN WHEN UPDATING THE
C * DOTFIL OR SAVTAP FILES')
C 2070 FORMAT(' DOTFIL FILE IS BEING INPUT')
C 2080 FORMAT(' SAVTAP FILE IS BEING INPUT')
C 2090 FORMAT(' CONDITIONAL CLUSTER MAP WILL BE OUTPUT')

```

ORIGINAL PAGE IS OF POOR QUALITY

```

SET04570
SET04580
SET04590
SET04600
SET04610
SET04620
SET04630
SET04640
SET04650
SET04660
SET04670
SET04680
SET04690
SET04700
SET04710
SET04720
SET04730
SET04740
SET04750
SET04760
SET04770
SET04780
SET04790
SET04800
SET04810
SET04820
SET04830
SET04840
SET04850
SET04860
SET04870
SET04880
SET04890
SET04900
SET04910
SET04920
SET04930
SET04940
SET04950
SET04960
SET04970
SET04980
SET04990
SET05000
SET05010
SET05020
SET05030
SET05040
SET05050
SET05060
SET05070
SET05080
SET05090
SET05100
SET05110
SET05120
SET05130
SET05140
SET05150
SET05160
SET05170
SET05180
SET05190
SET05200
SET05210
SET05220
SET05230
SET05240
SET05250
SET05260
SET05270
SET05280
SET05290
SET05300
SET05310
SET05320

```

FILE SET14

```
3000 FORMAT(' MIXED CLUSTER MAP WILL BE OUTPUT')
3010 FORMAT('/' USER HAS NOT INPUT ONE OF THE REQUIRED FILES: '/T20,
      *SAVTAP      MAPFIL      OR      DOTFIL')
3020 FORMAT(' PRINT UPDATED DOTFILE')
3030 FORMAT(' PRINT MEANS AND COVARIANCES')
3040  FORMAT(' EXIT IF INPUT LABEL NOT USED')
C
      RETURN
C
      END
```

```
SET05330
SET05340
SET05350
SET05360
SET05370
SET05380
SET05390
SET05400
SET05410
SET05420
```

FILE: STOMAP

```

C* SURROUTINE STOMAP(ILINE,NSAMP,HIST,LIMIT,BEGIN)
C* STODAT READS AND STORES THE CLASSIFICATION/CLUSTER MAP ON DRUM
C*
C* IMPLICIT INTEGER (A-Z)
C INCLUDE COMRAK.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* .NHSTUN,NHSTFI,SCTRUN,MAPFIL
* .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
CSEND
DIMENSION HIST(LIMIT),FETVEC(1),FLD(6),NLINE(4)
TOTWRD = ILINE*NSAMP
IF (TOTWRD .LE. (DRMWDS-(DRUMAD-BEGIN1))) GO TO 120
WRITE(6,110)
110 FORMAT(' NOT ENOUGH DRUM SPACE TO STORE DAS TAPE DATA')
CALL CMERR
C
120 CALL TAPHDR(MAPUNT,MAPFIL)
FETVEC(1) = 1
NOFEAT = 1
FLD(1) = 1
FLD(2) = ILINE
FLD(3) = 1
FLD(4) = 1
FLD(5) = NSAMP
FLD(6) = 1
REGIN = BEGIN
CALL FLDINI(FLD ,FETVEC,NOFEAT)
DUMPS = TOTWRD / LIMIT
IF (MOD(TOTWRD,LIMIT) .NE. 0) DUMPS = DUMPS + 1
TOTLNS = LIMIT / NSAMP
IF (TOTLNS .GE. ILINE) GO TO 140
DMP = DUMPS - 1
DO 130 I=1,DMP
130 NLINE(I) = TOTLNS
NLINE(DUMPS) = ILINE - TOTLNS*DMP
GO TO 150
140 NLINE(1) = ILINE
C
150 DO 200 J=1,DUMPS
NUMLIN = NLINE(J)
DO 160 K=1,NUMLIN
WORDS = NSAMP*(K-1)
160 CALL LINERD(HIST(WORDS+1),ENDTAP)
C
C STORE ON HIGH SPEED DRUM
C
NWORDS = WORDS + NSAMP
CALL RWRITE(BEGIN,HIST(1),NWORDS,(STAT)
200 REGIN = BEGIN + NLINE(J) * NSAMP
C
MAPFIL = MAPFIL + 1
C
RETURN
END
```

ST000010  
ST000020  
ST000030  
ST000040  
ST000050  
ST000060  
ST000070  
ST000080  
ST000090  
ST000100  
ST000110  
ST000120  
ST000130  
ST000140  
ST000150  
ST000160  
ST000170  
ST000180  
ST000190  
ST000200  
ST000210  
ST000220  
ST000230  
ST000240  
ST000250  
ST000260  
ST000270  
ST000280  
ST000290  
ST000300  
ST000310  
ST000320  
ST000330  
ST000340  
ST000350  
ST000360  
ST000370  
ST000380  
ST000390  
ST000400  
ST000410  
ST000420  
ST000430  
ST000440  
ST000450  
ST000460  
ST000470  
ST000480  
ST000490  
ST000500  
ST000510  
ST000520  
ST000530  
ST000540  
ST000550  
ST000560  
ST000570  
ST000580



FILE: RNI4A1

```
C      SUBROUTINE RNI4A1(IFLD,INCHR,IBN)
C      DAVID LEE SMITH 17 OCTOBER 1977.
C      A SUBROUTINE TO CONVERT INTERNAL BINARY NUMBERS FROM
C      THE INTEGER*4 FORM TO A STRING OF EBCDIC CHARACTERS.
C
C      THE PARAMETERS ARE:
C      IFLD IS THE FIRST WORD OF A FIELD IN AN ARRAY IN WHICH
C      TO STORE THE OUTPUT EBCDIC CHARACTERS, ONE PER
C      WORD, IN A1 FORM, I.E., ONE CHARACTER PER WORD, WITH
C      BLANK FILL TO THE RIGHT.
C      INCHR IS THE NUMBER OF EBCDIC CHARACTERS = THE NUMBER OF
C      WORDS = THE WIDTH OF THE FIELD TO BE FILLED.
C      INCHR SHOULD ALSO BE OF FORM INTEGER*4.
C      IBN IS THE INPUT TO THIS ROUTINE, AN INTEGER IN INTERNAL
C      BINARY FORM, POSITIVE OR NEGATIVE (TWO'S COMPLEMENT)
C      IN THE LEGAL RANGE -2**31 TO (2**31-1).
C
C      INTEGER*4 IFLD(20), INCHR, IBN, IDIG(10), IRL, IHMI
C      DATA IDIG / 1H0, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9 /
C      DATA IHMI / 1H- /
C      DATA IHR / 1HR /
C      LAST = INCHR
C      LBN = IBN
C      NEG = 0
C      IF ( LBN ) 10,30,30
C      NEG = 1
C      LBN = -LBN
C      IF ( LBN ) 20,30,30
C      IFLD(LAST) = IHR
C      LBN = 214 748 364
C      LAST = LAST - 1
C      IF (LAST .LE. 0 ) GO TO 60
C      IPT = LAST
C      DO 50 I = 1, LAST
C      INDEX = MOD( LBN, 10 )
C      IFLD(IPT) = IDIG( INDEX + 1 )
C      LBN = LBN / 10
C      IPT = IPT - 1
C      CONTINUE
C      IF ( NEG .NE. 0 ) IFLD(1) = IHMI
C      RETURN
C      END
```

BNI00010  
BNI00020  
BNI00030  
BNI00040  
BNI00050  
BNI00060  
BNI00070  
BNI00080  
BNI00090  
BNI00100  
BNI00110  
BNI00120  
BNI00130  
BNI00140  
BNI00150  
BNI00160  
BNI00170  
BNI00180  
BNI00190  
BNI00200  
BNI00210  
BNI00220  
BNI00230  
BNI00240  
BNI00250  
BNI00260  
BNI00270  
BNI00280  
BNI00290  
BNI00300  
BNI00310  
BNI00320  
BNI00330  
BNI00340  
BNI00350  
BNI00360  
BNI00370  
BNI00380  
BNI00390  
BNI00400  
BNI00410







FILE: CHAIN

```
100 WRITE (6,120)M CHA00510
M=M+1 CHA00620
IF (M.LT.LNCAT) GO TO 80 CHA00630
IF (KNCAT.EQ.LNCAT)RETURN CHA00640
WRITE (6,130)KNCAT CHA00650
RETURN CHA00660
110 FORMAT(/' THE FOLLOWING CLUSTERS SHOULD BE CHAINED---',20I4) CHA00670
120 FORMAT(/' IN THE FINAL OUTPUT MAP ALL OF THE ABOVE CLUSTERS WILL R CHA00680
*' REPRESENTED BY THE SYMBOL FOR CLUSTER',I4//) CHA00690
130 FORMAT(' THE ABOVE CHAINING REDUCES THE EFFECTIVE NUMBER OF CLUSTE CHA00700
*' PS TO ',I5) CHA00710
140 FORMAT(1H1) CHA00720
END CHA00730
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE CHLDET

```

SUBROUTINE CHLDET( KKK,NV,DUM,DET)
THIS ROUTINE COMPUTES THE MODIFIED CHOLESKY DECOMPOSITION OF
THE COVARIANCE MATRIX. THE DECOMPOSITIONS OVERLAY THE ELEMENTS
OF THE COVARIANCE MATRIX.
KK = L D L*
KK = COVARIANCE MATRIX STORED IN SYMMETRIC STORAGE
NV = NO. OF CHANNELS
DUM = A WORK AREA OF SIZE NV-1
DET = THE DETERMINANT OF THE COVARIANCE MATRIX

REAL KK,KKK
LOGICAL JE1
DIMENSION KKK(1), DUM(1),KK(465)

COPY COVARIANCE MATRIX FROM KKK TO KK TO AVOID OVERSTORING
THE INPUT MATRIX
ISIZE = (NV*(NV+1))/2
DO 5 I=1,ISIZE
KK(I)=KKK(I)
CONTINUE

DOUBLE PRECISION TF, R, R1, DUM, T1
JE1 = .TRUE.
J1 = 0
JD = 0
DET = 1.0

LOOP OVER ALL CHANNELS
DO 10 J=1,NV
KL = J-1
L = J+1
JD = J1
J1 = J1 + J
TF = KK(J1)
IF(JE1) GO TO 12
K1 = 0

COMPUTE THE DIAGONAL ELEMENTS OF D AND STORE IN KK
TEMPORARILY STORE THE PRODUCT KK(I,I)*KK(J,I) IN DUM(I)
DO 15 I=1,KL
R = KK(JD + I)
K1 = K1 + I
R1 = KK(K1) * R
TF = TF - R1 * R
DUM(I) = R1
CONTINUE
KK(J1) = TF
CONTINUE
DET = DET * TF
IF (L .GT. NV) GO TO 10
IRD = J1 - L + 1

COMPUTE THE R, J-TH ELEMENT OF L, USING T1
DO 20 IR=L,NV
IRD = IRD + IR - 1
T1 = KK(IRD + J)
IF(JE1) GO TO 16
DO 25 I=1,KL
T1 = T1 - DUM(I) * KK(IRD + I)
CONTINUE
IF(TF.GT.0.D0)GO TO 17
DET=0

```

MCH00010  
MCH00020  
MCH00030  
MCH00040  
MCH00050  
MCH00060  
MCH00070  
MCH00080  
MCH00090  
MCH00100  
MCH00110  
MCH00120  
MCH00130  
MCH00140  
MCH00150  
MCH00160  
MCH00170  
MCH00180  
MCH00190  
MCH00200  
MCH00210  
MCH00220  
MCH00230  
MCH00240  
MCH00250  
MCH00260  
MCH00270  
MCH00280  
MCH00290  
MCH00300  
MCH00310  
MCH00320  
MCH00330  
MCH00340  
MCH00350  
MCH00360  
MCH00370  
MCH00380  
MCH00390  
MCH00400  
MCH00410  
MCH00420  
MCH00430  
MCH00440  
MCH00450  
MCH00460  
MCH00470  
MCH00480  
MCH00490  
MCH00500  
MCH00510  
MCH00520  
MCH00530  
MCH00540  
MCH00550  
MCH00560  
MCH00570  
MCH00580  
MCH00590  
MCH00600  
MCH00610  
MCH00620  
MCH00630  
MCH00640  
MCH00650  
MCH00660  
MCH00670  
MCH00680  
MCH00690  
MCH00700  
MCH00710  
MCH00720  
MCH00730  
MCH00740  
MCH00750  
MCH00760

FILE CHLDET

```
      RETURN
17  KK(IRO * J) = T1/TF
20  CONTINUE
      JF1 = .FALSE.
10  CONTINUE
C
C
C
C
      KK CONTAINS , IN 'SYMMETRIC' STORAGE, THE MODIFIED CHOLESKY
      FACTORIZATION OF THE INPUT MATRIX. THE LOWER TRIANGULAR MATRIX, L,
      OCCUPIES THE OFF-DIAGONAL ELEMENTS OF KK , AND THE DIAGONAL
      MATRIX, D , IS STORED IN THE DIAGONAL ELEMENTS IN KK.
      RETURN
      END
```

MCH00770  
MCH00780  
MCH00790  
MCH00800  
MCH00810  
MCH00820  
MCH00830  
MCH00840  
MCH00850  
MCH00860  
MCH00870  
MCH00880  
MCH00890

ORIGINAL PAGE IS  
OF POOR QUALITY



ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: CLSCHK

```
      SURROUTINE CLSCHK (CLSDS, SUBDES, FLDSAV, VERTEX, SUBNO,  
      * NOFEAT, FETVEC, NOCLS, NOFLD, BMFLG, NOSUB) CLS00010  
C  
      IMPLICIT INTEGER (A-H, O-Z) CLS00020  
C  
----- CLS00030  
----- CLS00040  
----- CLS00050  
----- CLS00060  
----- CLS00070  
----- CLS00080  
----- CLS00090  
      THIS SURROUTINE IS CALLED FROM REDSAV TO CHECK THE VALIDITY OF  
      USER REQUESTS REGARDING SUBCLASSES, GROUPING AND CHANNELS CLS00100  
----- CLS00110  
----- CLS00120  
----- CLS00130  
----- CLS00140  
----- CLS00150  
----- CLS00160  
----- CLS00170  
----- CLS00180  
      INCLUDE COMR1.LIST CLS00190  
      COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2, COM00010  
      * AVAR2, COVAR2, CLSID2, SURNO2, SURDS2, FLDSV2, VERTX2, COM00020  
      * FETVC2(30), SURVC2(75), SURPTR(75), CLSVC2(60), COM00030  
      * KEPRT5(60), NOGRP, GRPNAM(60), GRPDEX(61), COM00040  
      * GRPCHK(61), GROUPS(124) COM00050  
CSEND  
----- CLS00200  
----- CLS00210  
----- CLS00220  
----- CLS00230  
      DIMENSION INVERT(30) CLS00240  
      DIMENSION SETI(60) CLS00250  
      DATA MAXFET/30/ CLS00260  
      DIMENSION FETVEC(30) CLS00270  
C  
      DIMENSION CLSDS(1), SUBDES(1), FLDSAV(4, NOFLD), VERTEX(1) CLS00280  
C  
      DIMENSION SUBNO(1) CLS00290  
----- CLS00300  
      IF (NOSUB2.LE.0) GO TO 40 CLS00310  
      II = 0 CLS00320  
      IRIG = 0 CLS00330  
      DO 30 I=1, NOSUB2 CLS00340  
      J = SURVC2(I) CLS00350  
      IF (J.LE.IBIG .OR. J.GT.NOSUB) GO TO 10 CLS00360  
      II = II+1 CLS00370  
      SURVC2(II) = J CLS00380  
      IRIG = J CLS00390  
      GO TO 30 CLS00400  
10 WRITE (6, 20) J CLS00410  
20 FORMAT (/5X, '**CLSCHK** - REQUESTED SUBCLASS NO.,', I3, ' IS NOT AVA CLS00420  
*ILABLE IN INPUT STATISTICS -- REQUEST IGNORED*/) CLS00430  
30 CONTINUE CLS00440  
      NOSUR2 = II CLS00450  
      IF (NOSUR2.GT.0) GO TO 60 CLS00460  
40 NOSUR2 = NOSUR CLS00470  
      IF (NOSURH.GT.60) NOSUB2 = 60 CLS00480  
      DO 50 I=1, NOSUR2 CLS00490  
50 SURVC2(I) = I CLS00500  
C  
      CHECK THE GROUPS FOR VALIDITY CLS00510  
----- CLS00520  
----- CLS00530  
----- CLS00540  
----- CLS00550  
60 GRPTR = 0 CLS00560  
      IF (NOGRP.LE.0) GO TO 110 CLS00570  
      II = 0 CLS00580  
      DO 100 I=1, NOGRP CLS00590  
      JR = GRPDEX(I)+1 CLS00600  
      JE = JR+GROUPS(JR-1)-1 CLS00610  
      GRPTR = GRPTR+1 CLS00620  
      II = GRPTR CLS00630  
      DO 90 J=JR, JE CLS00640  
      JJ = GROUPS(J) CLS00650  
      IF (JJ.GT.NOSUR) GO TO 70 CLS00660  
      GRPTR = GRPTR+1 CLS00670  
      GROUPS(GRPTR) = JJ CLS00680  
      GO TO 90 CLS00690  
70 WRITE (6, 80) JJ, I CLS00700  
80 FORMAT (/5X, '**CLSCHK** - REQUESTED SUBCLASS NO.,', I3, ' FOR GROUP CLS00710  
*NO.,', I3, ' IS NOT AVAILABLE IN INPUT STATISTICS FILE*/) CLS00720  
90 CONTINUE CLS00730  
      GRPTR = GRPTR-1 CLS00740  
      IF (GRPTR.LT.II) GO TO 100 CLS00750
```





FILE: CLSCHK

```
GO TO 240
272 WRITE(A,273) (FETVC2(K),K=1,NOFET2)
273 FORMAT(// 5X,10000 B-MATRIX CHANNELS MUST BE EQUAL TO OR A SUB-CLASS
1ET OF AVAILABLE TRAINING DATA CHANNELS --- THE INPUT B-MATRIX CHANNELS
2NEL SET IS ...// (5X,3014//) )
WRITE(A,274)
274 FORMAT(///// 5X,10000 TERMINATING PROGRAM EXECUTION FROM FETCHK
100000//M1)
CALL EXIT
240 CONTINUE
NOFET2 = II
C
C
C-----
SET UP REVISED INVERT TABLE
C-----
DO 290 I=1,MAXFET
290 INVERT(I) = 0
DO 300 I=1,NOFET2
WAT=FETVC2(I)
300 INVERT(WAT) = I
C
C
C SET UP CLSVC2 ARRAY SO THAT IT CONTAINS THE CLASS NO. TO WHICH THE
CORRESPONDING SURCLASS BELONGS.
J=0
DO 305 I=1,NOCLS2
IK=SURNO(I)
DO 305 K=1,IK
J=J+1
305 CLSVC2(J)=I
C
C
C SAVE FIELD DESCRIPTIONS FOR CLASSES AND SUBCLASSES TO BE USED.
NOFLD2=0
IV=0
JV=0
DO 330 I=1,NOFLD
JR=FLOSAV(3,I)
C ARE FIELDS ASSOCIATED WITH SUBCLASSES
IF(JR.NE.0)GO TO 306
IC=FLOSAV(2,I)
C HAS THIS CLASS BEEN ELIMINATED
NC=SETI(IC)
IF(INC.EQ.-1)GO TO 320
NSUR=0
GO TO 307
306 CONTINUE
C HAS THIS SURCLASS BEEN ELIMINATED
NSUR = SURPTR(JR)
IF(NSUR.EQ.0)GO TO 320
NC=CLSVC2(NSUR)
307 CONTINUE
NOFLD2=NOFLD2+1
FLOSAV(1,NOFLD2)=FLOSAV(1,I)
FLOSAV(2,NOFLD2)=NC
FLOSAV(3,NOFLD2)=NSUR
FLOSAV(4,NOFLD2)=FLOSAV(4,I)
C
C
C NOW SAVE VERTICES
NV=FLOSAV(4,I)*2
DO 310 J=1,NV
IV=IV+1
310 VERTEX(IV) = VERTEX(JV+J)
JV=JV+NV
GO TO 330
320 JV = JV + FLOSAV(4,I)*2
330 CONTINUE
TOTVT2=IV/2
RETURN
END
CLS02340
CLS02350
CLS02360
CLS02370
CLS02380
CLS02390
CLS02400
CLS02410
CLS02420
CLS02430
CLS02440
CLS02450
CLS02460
CLS02470
CLS02480
CLS02490
CLS02500
CLS02510
CLS02520
CLS02530
CLS02540
CLS02550
CLS02560
CLS02570
CLS02580
CLS02590
CLS02600
CLS02610
CLS02620
CLS02630
CLS02640
CLS02650
CLS02660
CLS02670
CLS02680
CLS02690
CLS02700
CLS02710
CLS02720
CLS02730
CLS02740
CLS02750
CLS02760
CLS02770
CLS02780
CLS02790
CLS02800
CLS02810
CLS02820
CLS02830
CLS02840
CLS02850
CLS02860
CLS02870
CLS02880
CLS02890
CLS02900
CLS02910
CLS02920
CLS02930
CLS02940
CLS02950
CLS02960
CLS02970
CLS02980
CLS02990
CLS03000
CLS03010
CLS03020
```



FILE: CLSHIS

```

C SURROUTINE CLSHIS (TALLY, HISBUF, TTL, XSIZ, XHGM, XLOW, YSIZ, CLS00010
C *NOMIST, FLOPTS, HISVEC) CLS00020
C IMPLICIT INTEGER (A-H,O-Z) CLS00030
C REAL XSCALE, XSHFT CLS00040
C INCLUDE COMRK6.LIST CLS00050
C COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY, CLS00060
C * HISFIL, HISKEY, THFORM, ERIPTP, ERPKEY, MAPUNT, NOFILE, CLS00070
C * DRUMAD, DRMHDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL, CLS00080
C * NMSTUN, NMSTFI, SCTRUN, MAPFIL, CLS00090
C * DOTUNT, DOTFIL, NCHPAS, TRANSFL, BMTRFL, HISTFL, PCHUNT, CLS00100
C * CRDUNT, PRUNT, RANDIO CLS00110
CSEND CLS00120
C DIMENSION HISVEC(30) CLS00130
C DIMENSION TALLY(NOMIST, XSIZ), HISBUF(XSIZ), XAXIS(1) CLS00140
C LOGICAL *1 DUMM(4), SYMM(4) CLS00150
C EQUIVALENCE (SYM, SYMM(1)), (DUM, DUMM(1)) CLS00160
C DATA STAR/'*'/, BLANK/' '/, ALPHA/'ZC0/' CLS00170
C DATA DOLBCD/'S'/, NUMIC/'ZF0/', ALPHA/'ZC0/' CLS00180
C ----- CLS00190
C ----- CLS00200
C ----- CLS00210
C ----- CLS00220
C ----- CLS00230
C ----- CLS00240
C ----- CLS00250
C ----- CLS00260
C GO = 1 CLS00270
C INIZ CLS00280
C ----- CLS00290
C ----- CLS00300
C ----- CLS00310
C ----- CLS00320
10 JPTCNT = ( PAGESZ-R)/(YSIZ * 10 ) CLS00330
C NSIZ = (XSIZ-R)/10 * 1 CLS00340
C XSCALE = FLOAT(XLOW-XHGM)/(XSIZ-1) CLS00350
C XSHFT = FLOAT(XSIZ-XHGM-XLOW)/(XSIZ-1) CLS00360
C DO 20 I=1,NSIZ CLS00370
C 20 XAXIS(I) = (10*I-9)*XSCALE + XSHFT + 0.50; CLS00380
C JCNT = JPTCNT CLS00390
C DO 160 JFEAT=1, NOMIST CLS00400
C IF (JCNT.LT.JPTCNT) GO TO 40 CLS00410
C WRITE (6, HEAD) CLS00420
C IF (GO.EQ.4) WRITE (6, 11) CLS00430
C 11 FORMAT (' DATA TR') CLS00440
C IF (GO.EQ.1) WRITE (6, 200) TTL CLS00450
C IF (GO.EQ.0) WRITE (6, 201) TTL CLS00460
C 200 FORMAT (T63, 'HISTOGRAM' / T59, '-----' // T55, 'TRAINING SUBC' CLS00470
C * LASS 'A4' / T53, '-----' //) CLS00480
C 201 FORMAT (T63, 'HISTOGRAM' / T59, '-----' // T55, 'TRAINING FIEL' CLS00490
C * D 'A4' / T53, '-----' //) CLS00500
C IF (GO.EQ.2) WRITE (6, 145) TTL, FLOPTS CLS00510
C IF (GO.EQ.3) WRITE (6, 147) TTL CLS00520
C IF (GO.EQ.0) WRITE (6, 30) FLOPTS, TITLE CLS00530
C 30 FORMAT (T48, '(NO. SAMPLES = ', I7, ', SUBCLASS = ', A4, ')') CLS00540
C JCNT = 0 CLS00550
C ----- CLS00560
C SCALE AND PRINT THE HISTOGRAM CLS00570
C ----- CLS00580
C ----- CLS00590
C ----- CLS00600
40 MAX = 0 CLS00610
C YSCALE = 1 CLS00620
C JCNT = JCNT + 1 CLS00630
C DO 50 J=1, XSIZ CLS00640
C IF (TALLY(JFEAT, J) .GT. MAX) MAX = TALLY(JFEAT, J) CLS00650
C 50 CONTINUE CLS00660
C IF (MAX .GT. YSIZ) YSCALE = (MAX+(YSIZ-1))/YSIZ CLS00670
C K = HISVEC(JFEAT) CLS00680
C IF (GO.EQ.4) GO TO 62 CLS00690
C WRITE (6, 60) K CLS00700
C 60 FORMAT (' CHANNEL ', I2 / 'X, '-----') CLS00710
C 62 CONTINUE CLS00720
C WRITE (6, 70) YSCALE CLS00730
C 70 FORMAT ('0', T5, 'EACH REPRESENTS ', I8, ' POINT(S).') CLS00740
C DO 120 JY=1, YSIZ CLS00750
C JH = (YSIZ-(JY-1))*YSCALE CLS00760
C JK = JH - YSCALE CLS00770
C JEMP = XSIZ CLS00780
C ----- CLS00790

```



FILE: CMERR

```
SUBROUTINE CMERR
WRITE(6,100)
100 FORMAT(' ERROR HAS OCCURRED')
CALL EXIT
RETURN
END
```

```
CME00010
CME00020
CME00030
CME00040
CME00050
CME00060
```



FILE: DESCEN

	SUBROUTINE DESCEN(SCN,LNCAT,PTR1,PTR2)	DES00010
	IMPLICIT INTEGER(A-X)	DES00020
	DIMENSION PTR1(LNCAT),PTR2(LNCAT)	DES00030
	REAL SCN(LNCAT),SAVE	DES00040
	J=0	DES00050
60	J=J+1	DES00060
	IF(J.GT.LNCAT)GO TO 90	DES00070
	IF(J.EQ.LNCAT)GO TO 75	DES00080
	IF(SCN(J).LT.SCN(J+1))GO TO 70	DES00090
	GO TO 60	DES00100
C		DES00110
70	SAVE=SCN(J)	DES00120
	SCN(J)=SCN(J+1)	DES00130
	SCN(J+1)=SAVE	DES00140
C		DES00150
	SAVE1=PTR1(J)	DES00160
	PTR1(J)=PTR1(J+1)	DES00170
	PTR1(J+1)=SAVE1	DES00180
C		DES00190
	SAVE2=PTR2(J)	DES00200
	PTR2(J)=PTR2(J+1)	DES00210
	PTR2(J+1)=SAVE2	DES00220
75	K=J	DES00230
80	IF(K.EQ.1)GO TO 60	DES00240
	IF(SCN(K).LT.SCN(K-1))GO TO 60	DES00250
C		DES00260
	SAVE=SCN(K-1)	DES00270
	SCN(K-1)=SCN(K)	DES00280
	SCN(K)=SAVE	DES00290
C		DES00300
	SAVE1=PTR1(K-1)	DES00310
	PTR1(K-1)=PTR1(K)	DES00320
	PTR1(K)=SAVE1	DES00330
C		DES00340
	SAVE2=PTR2(K-1)	DES00350
	PTR2(K-1)=PTR2(K)	DES00360
	PTR2(K)=SAVE2	DES00370
	K=K-1	DES00380
	GO TO 80	DES00390
90	CONTINUE	DES00400
	RETURN	DES00410
	END	DES00420

FILE: DSTAPE

```
C.....DST00010
C*
C*      ISOCLS SUBROUTINE                                DST00020
C*
C*      THIS SUBROUTINE GENERATES A CLUSTER TAPE IN EITHER DST00030
C*      UNIVERSAL OR LARSYS II FORMAT. THE IMAGE MAY OPTIONALLY BE A DST00040
C*      ONE CHANNEL TAPE REFLECTING THE CLUSTER NUMBER OF EACH PIXEL. DST00050
C*      OR A 'NOFEAT' CHANNEL TAPE REFLECTING THE MEAN VECTOR OF THE DST00060
C*      CLUSTER TO WHICH THE PIXEL WAS ASSIGNED. DST00070
C*
C*
C*      SUBROUTINE DSTAPE(IPLACE,IBUF,MEANS,/FLDINF/) DST00080
C*      IMPLICIT INTEGER(A-X) DST00090
C*      INCLUDE COMPK5.LIST DST00100
C*      INCLUDE COMPK6.LIST DST00110
C*      COMMON/PASS/STOP,LNCAT,NMIN,KRN,STDMAX,OLMIN,SEP, DST00120
C*      MAP,SPTRIG,IRD,KPTS,NOPTS,PUNCH, DST00130
C*      ICHN,CHNTHS,ICHAIN(62),NWDS,IBEGIN,REGIN1, DST00140
C*      REGIN2,BEGIN3,CLSNAM,NOFLD,IPT,TOTWRD,TOTPTS, DST00150
C*      NCLAS,NOCLS,TOTSUB,TOTFLD,TOTVRT,NOCL,NVRT DST00160
C*      ,NXTCLS,NOFEAT,MAXCLS,FETVEC(30),SYMMTX(62) DST00170
C*      ,VAPSI7,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP DST00180
C*      ,IORDER,INUNIT,INFILE,INITM,PMIN,SURVEC(62),NOSUB2,CHNVC(30) DST00190
C*      ,NOCHAN,ERCOMP,NOSEQ,MEAND0,MEANDU, DST00200
C*      SYMDO,SYMDU,ITRIG0,ITRIGU,DOFLAG, DST00210
C*      DUFLAG,DODU,STDOTS(60),NSDOTS,SUNCOR(30),LLNCAT, DST00220
C*      DVERT(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2) DST00230
C*      ,MXFFT1,MAXPOP DST00240
C*      REAL SUNCOR DST00250
C*      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY, DST00260
C*      ,HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE, DST00270
C*      ,DRUMAD,DRMWD5,PAGSI7,DATFIL,STAFIL,ASAV,ASAVFL DST00280
C*      ,NHSTUN,NHSTFI,SCTPUN,MAPFIL DST00290
C*      ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, DST00300
C*      CRDUNT,PRTUNT,NANDIO DST00310
C*
CSEND DIMENSION IRUF(1),FETVC2(30) DST00320
REAL MEANS(1) DST00330
NC=1 DST00340
IF(MAPKEY.EQ.1)NC=NOFEAT DST00350
WRITE(6,HEAD) DST00360
C*
C* DIMENSION FINF(6),FL(22) DST00370
DIMENSION IPTT(62) DST00380
EQUIVALENCE (FINF(1),LINSTR),(FINF(4),SAMSTR) DST00390
EQUIVALENCE (FINF(2),LINEND),(FINF(5),SAMEND) DST00400
EQUIVALENCE (FINF(3),LININC),(FINF(6),SAMINC) DST00410
DIMENSION IPLACE(NOPTS) DST00420
DIMENSION FLDINF(1) DST00430
DIMENSION FORM(3,1) DST00440
DATA FORM/UNIV,ERSA,IL ,LARS,YS I,,I // DST00450
ICOUNT=0 DST00460
DO 3 I=1,LNCAT DST00470
IPTT(I)=1 DST00480
IF(ICHN.GT.0)GO TO 5 DST00490
DO 1 I=1,LNCAT DST00500
1 ICHAIN(I)=1 DST00510
5 CONTINUE DST00520
ADRES=REGIN2 DST00530
K=0 DST00540
IRFC=IRD DST00550
IPTS=NOPTS DST00560
10 IF(IRFC.LE.1)IPTS=KPTS DST00570
IF (IRD.EQ.0) GO TO 20 DST00580
CALL PREAD(ADRES,IPLACE,IPTS,ISTAT) DST00590
15 IF(ISTAT.EQ.1)GO TO 15 DST00600
ADRES=ADRES+IPTS DST00610
20 CONTINUE DST00620
IV=5 DST00630
DO 50 IFLD=1,NOFLD DST00640
HOREC = 1 DST00650
NV=FLDINF(IV+1) DST00660
IR=IV+2*NV+2 DST00670
DO 25 I=1,6 DST00680
25 FINE(I)=FLDINF(IR+I) DST00690
LINES=(LINEND-LINSTR)/LININC +1 DST00700
NPTS=(SAMEND-SAMSTR)/SAMINC +1 DST00710
NPTS=NC*NPTS DST00720
IF(NPTS.GT.11500)GO TO 80 DST00730
DST00740
DST00750
DST00760
DST00770
DST00780
DST00790
```

FILE: DSTAPE

```

LINE = LINSTR-LININC.
DO 40 I=1,LINES
C* ZERO IRUF
C*
DO 26 J=1,NPTS
26 IRUF(J)=0
LINE = LINE + LININC
CALL FDLINT(FLDINF(IV+2),NV,FL,LINE,SAMPS,NFL)
DO 31 L=1,NFL,2
IB=(FL(L)-SAMSTR)/SAMINC+1
IE=(FL(L+1)-SAMSTR)/SAMINC+1
IF(MOD(SAMSTR,SAMINC).NE.MOD(FL(L),SAMINC))IB=IR+1
IF(IR.GT.IE)GO TO 31
DO 30 J=IB,IE
K=K+1
KP=IPLACE(K)
IF(MAPKEY.EQ.1)GO TO 70
IRUF(J)=ICHAIN(KP)
GO TO 75
70 DO 71 L7=1,NC
JJ=(L7-1)*LPTS+J
KK=(KP-1)*NOFEAT+LZ
71 IRUF(JJ) = MEANS(KK) + 0.5
75 IF(K.LT.IPTS)GO TO 30
IPFC=IREC-1
IF (IREC.EQ.0) GO TO 30
IF(IREC.EQ.1)IPTS=KPTS
CALL WREAD(ADRES,IPLACE,IPTS,ISTAT)
ADRES=ADRES+IPTS
27 IF(ISTAT.EQ.1)GO TO 27
K=0
30 CONTINUE
31 CONTINUE
GO TO (35,37),HDREC
C
C
C
35 WRITE HEADER RECORD
CONTINUE
LSTLIN = 0
LNFS = 0
IF(MAPKEY.EQ.1)GO TO 72
FETVC2(1)=1
GO TO 73
72 DO 74 KK=1,NOFEAT
74 FETVC2(KK)=KK
73 HDREC=2
NOFILE = NOFILE + 1
CALL WRTHED(NC,FETVC2,LPTS,MAPFMT,MAPUNT)
C
C
C
WRITE DATA RECORD
37 LNFS = LNFS + 1
IF(LNFS.EQ.LINES.AND.MAPKEY.NE.1)LSTLIN=-1
CALL WRTLN(IRUF,LSTLIN)
40 CONTINUE
IF(MAPKEY.NE.1)GO TO 48
IF(ORDER.EQ.1)CALL RANK(NOFEAT,FETVC2,LNCAT,MEANS,IPTT)
DO 41 I=1,NPTS
41 IRUF(I)=0
CALL WRTLN(IRUF,LSTLIN)
NTFN=10
JTFN=NTFN
NCLUS=LNCAT
DIV=11
II=1
NRL=LPTS/DIV
IF(NRL.LE.0)NRL=1
IF(LPTS.LT.11)JTFN=LPTS
IF(LPTS.LT.11)DIV=LPTS
51 CNT=NRL
IF((NCLUS-NRL).LT.0)CNT=NCLUS
NCLUS=NCLUS-NRL
IST=NRL*(II-1)+1
IEND=IST+CNT-1
II=II+1
III=0
DO 42 I=IST,IEND
III=III+1
DO 42 K=1,NC

```

DST00800  
DST00810  
DST00820  
DST00830  
DST00840  
DST00850  
DST00860  
DST00870  
DST00880  
DST00890  
DST00900  
DST00910  
DST00920  
DST00930  
DST00940  
DST00950  
DST00960  
DST00970  
DST00980  
DST00990  
DST01000  
DST01010  
DST01020  
DST01030  
DST01040  
DST01050  
DST01060  
DST01070  
DST01080  
DST01090  
DST01100  
DST01110  
DST01120  
DST01130  
DST01140  
DST01150  
DST01160  
DST01170  
DST01180  
DST01190  
DST01200  
DST01210  
DST01220  
DST01230  
DST01240  
DST01250  
DST01260  
DST01270  
DST01280  
DST01290  
DST01300  
DST01310  
DST01320  
DST01330  
DST01340  
DST01350  
DST01360  
DST01370  
DST01380  
DST01390  
DST01400  
DST01410  
DST01420  
DST01430  
DST01440  
DST01450  
DST01460  
DST01470  
DST01480  
DST01490  
DST01500  
DST01510  
DST01520  
DST01530  
DST01540  
DST01550  
DST01560  
DST01570  
DST01580

FILE: DSTAPE

```
      KK=(IPTT(I)-1)*NC*K
      DO 42 J=1,JTEN
      L=(III-1)*DIV*(K-1)*LPTS*J
42     IBUF(L) = MEANS(KK) * 0.5
      DO 43 I=1,NTEN
      ICOUNT=ICOUNT+1
43     CALL WRTLN(IBUF,LSTLIN)
      DO 44 I=1,NPTS
44     IBUF(I)=0
      IF(INCLUS.LE.0) LSTLIN=-1
      CALL WRTLN(IBUF,LSTLIN)
      IF(INCLUS.GT.0) GO TO 51
4R     CONTINUE
      WRITE(6,60) NOFILE,FLOINF(IV),(FORM(I,MAPFMT),I=1,3),LNES
      *ICOUNT
60     FORMAT(///T55,'FILE NO.      - ',16,/T55,'FIELD NAME  - ',A4,/
      * T55,'FORMAT      - ',3A4,/T55,'NO. OF SCAN LINES - ',16,/T55,
      *'NO. OF COLOR KEY SCAN LINES - ',16)
      IV = IV * NV*2 * 9
50     CONTINUE
      RETURN
R0     WRITE(4,81)
R1     FORMAT('THE NUMRER OF CHANNELS TIMES THE NUMBER OF SAMPLES HAS
1EXCFEDFD 11500.DECREASE THE NUMBER OF CHANNELS OR THE NUMBER OF
2SAMPLES.TERMINATING RUN FROM DSTAPE')
      CALL CMERR
      END
```

DST01590  
DST01600  
DST01610  
DST01620  
DST01630  
DST01640  
DST01650  
DST01660  
DST01670  
DST01680  
DST01690  
DST01700  
DST01710  
DST01720  
DST01730  
DST01740  
DST01750  
DST01760  
DST01770  
DST01780  
DST01790  
DST01800  
DST01810  
DST01820  
DST01830  
DST01840  
DST01850





FILE: FDLINT

```

IF ( L.NE. NPTSE ) GO TO 3000
IF ( L.NE. Y2 ) GO TO 3000
XNM1=X1
YNM1=Y1
X1=X2
Y1=Y2
X2=FIELD(1,2)
Y2=FIELD(2,2)
GO TO 3001
3000 IF ( L.NE. Y1 ) GO TO 2000
3001 IF ((Y1.LT. YNM1) .AND. (Y1.GT. Y2 )) GO TO 4000
IF ((Y1.GT. YNM1) .AND. (Y1.LT. Y2 )) GO TO 4000
GO TO 2000
4000 FL(JJ) = 0
JJ = JJ-1
GO TO 2000
1000 IF (L.NE. Y1) GO TO 2000
IF (X1.GT. X2) GO TO 5000
IF (YNM1.LT. Y1) GO TO 6000
IF ( YNP2 .GT. Y2 ) GO TO 7000
JJ = JJ+1
FL(JJ) = X1
GO TO 2000
7000 JJ = JJ+1
FL(JJ) = X1
MM = JJ+1
FL(MM) = X2
JJ = MM
GO TO 2000
6000 IF ( YNP2 .LT. Y2 ) GO TO 2000
JJ = JJ+1
FL(JJ) = X2
GO TO 2000
5000 IF ( YNM1 .LT. Y1 ) GO TO 9000
IF ( YNP2 .GT. Y2 ) GO TO 2000
JJ = JJ+1
FL(JJ) = X2
IF (NPTSE.EQ.2) FL(JJ)=X1
GO TO 2000
9000 IF ( YNP2 .GT. Y2 ) GO TO 8000
JJ = JJ+1
FL(JJ) = X1
MM = JJ+1
FL(MM) = X2
JJ = MM
GO TO 2000
8000 JJ = JJ+1
FL(JJ) = X1
GO TO 2000
5 NPTS1 = JJ-1
DO 29 NI = 1,NPTS1
NP1 = NI+1
DO 29 NJ = NP1,JJ
IF ( FL(NI) - FL(NJ) ) 29,29,28
28 NTEMP = FL(NI)
FL(NI) = FL(NJ)
FL(NJ) = NTEMP
20 CONTINUE
NSAMP = 0
DO 30 N = 1,JJ,2
NN = N+1
NSAMP = NSAMP+(FL(NN) -FL(N)+1)
30 CONTINUE
RETURN
35 IF (YLINE.NE.FIELD(2,1)) RETURN
FL(1)=FIELD(1,1)
FL(2)=FIELD(1,1)
NSAMP=1
JJ=2
RETURN
END

```

FDL00800  
 FDL00810  
 FDL00820  
 FDL00830  
 FDL00840  
 FDL00850  
 FDL00860  
 FDL00870  
 FDL00880  
 FDL00890  
 FDL00900  
 FDL00910  
 FDL00920  
 FDL00930  
 FDL00940  
 FDL00950  
 FDL00960  
 FDL00970  
 FDL00980  
 FDL00990  
 FDL01000  
 FDL01010  
 FDL01020  
 FDL01030  
 FDL01040  
 FDL01050  
 FDL01060  
 FDL01070  
 FDL01080  
 FDL01090  
 FDL01100  
 FDL01110  
 FDL01120  
 FDL01130  
 FDL01140  
 FDL01150  
 FDL01160  
 FDL01170  
 FDL01180  
 FDL01190  
 FDL01200  
 FDL01210  
 FDL01220  
 FDL01230  
 FDL01240  
 FDL01250  
 FDL01260  
 FDL01270  
 FDL01280  
 FDL01290  
 FDL01300  
 FDL01310  
 FDL01320  
 FDL01330  
 FDL01340  
 FDL01350  
 FDL01360  
 FDL01370  
 FDL01380  
 FDL01390  
 FDL01400  
 FDL01410  
 FDL01420  
 FDL01430  
 FDL01440  
 FDL01450  
 FDL01460  
 FDL01470  
 FDL01480  
 FDL01490  
 FDL01500

FILE: FIND12

```
C      FUNCTION FIND12(CARD,COL,VECTOR)
C      IMPLICIT INTEGER (A-H,O-Z)
-----
C      CALL..      J = FIND12(CARD,COL,VECTOR)
C      ARGS..      CARD   - BCD BUFFER
C                   COL   - PTR TO POSTION IN 'CARD'
C                   VECTOR - VEC CONTAINING N SYMBOLS
C                       TO BE LOCATED IN CARD
C                       (N IS GIVEN IN VECTOR(1))
C                       EG,  /2.'S', 'E'/
C
C      REQUIRES..  NOR
C      PURPOSE..  USED TO LOCATE SPECIAL SYMBOLS IN 'CARD'
C      RETURNS..  COL   - PTS AT SYMROL ( IF LOCATED)
C                   J   - PTS AT SYMBOL LOCATED IN 'VECTOR'
-----
C      DIMENSION CARD(1),VECTOR(1)
C      DATA CRDSIZ/62/
C      K = VECTOR(1)+1
C      L = COL+1
C      DO 10 COL=L,CRDSIZ
C      DO 10 I=2,K
C      II = I
C      IF (CARD(COL).EQ.VECTOR(II)) GO TO 20
10  CONTINUE
C      I = I - 1
C      COL = L - 1
C      20 FIND12 = I
C      WRITE( 6,102) (CARD(K),K=1,62),COL,I,VECTOR(I)
C 102  FORMAT(' FIND12 ENTERED'/ ' ',62A1,110/' ',15,A4)
C      RETURN
C      END
```

F N00010  
F N00020  
F N00030  
F N00040  
F N00050  
F N00060  
F N00070  
F N00080  
F N00090  
F N00100  
F N00110  
F N00120  
F N00130  
F N00140  
F N00150  
F N00160  
F N00170  
F N00180  
F N00190  
F N00200  
F N00210  
F N00220  
F N00230  
F N00240  
F N00250  
F N00260  
F N00270  
F N00280  
F N00290  
F N00300  
F N00310  
F N00320  
F N00330  
F N00340  
F N00350  
F N00360  
F N00370  
F N00380  
F N00390  
F N00400  
F N00410  
F N00420  
F N00430  
F N00440  
F N00450  
F N00460  
F N00470  
F N00480

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE FLOINT

```

SUBROUTINE FLOINT (/BLOCK/, /FETVEC/, /NOFEAT) - FLD00010
IMPLICIT INTEGER (A-Z) FLD00020
C* FLD00030
C* ENTRY FOR POSITIONING TAPE TO CORRECT SCAN LINE FOR A SPECIFIC FIELD FLD00040
C* FLD00050
COMMON /TAPERD/ IUNIT, IFRST, FSCAN, SAMEND, SAMINC, READY, NSCAN, FLD00060
* LINC, IU(200), DSL, LHUF(30), JREC(30), IAYTE(30), NRUFS, FILENO, LINEN FLD00070
* D, LINC, NSAMP, NOCHAN, FORMT FLD00080
DIMENSION BLOCK(6) FLD00090
DIMENSION FETVEC(NOFEAT) FLD00100
EQUIVALENCE ((ID(1), NRPOS), (ID(2), NCPR), FLD00110
, (ID(3), NPWC), (ID(4), ANCLNG), FLD00120
, (ID(5), NC), (ID(6), NS), FLD00130
, (ID(7), NRITS), (ID(8), DOI), FLD00140
, (ID(9), NDSPR), (ID(10), NCAR), FLD00150
, (ID(11), SVD), (ID(16), PRSZ) FLD00160
READY = 1 FLD00170
NOCHAN = NOFEAT FLD00180
C* CHECK FETVEC FLD00190
DO 125 I=1, NOFEAT FLD00200
IF (FETVEC(I) .GT. NC) GO TO 126 FLD00210
125 CONTINUE FLD00220
GO TO 127 FLD00230
124 WRITE(6,470)NC FLD00240
CALL CMERR FLD00250
127 CONTINUE FLD00260
LINSTR=BLOCK(1) FLD00270
IF (LINSTR.GE. IFRST) GO TO 130 FLD00280
WRITE(6,430)IFRST FLD00290
CALL CMERR FLD00300
130 CONTINUE FLD00310
IF (BLOCK(2).GE. IFRST)GO TO 132 FLD00320
WRITE(6,430)IFRST FLD00330
CALL CMERR FLD00340
132 CONTINUE FLD00350
IF (FORMT.EQ.3)GO TO 200 FLD00360
IF (FORMT.EQ.4)GO TO 220 FLD00370
C* FLINE=FIRST SCAN ON RECORD CONTAINING LINSTR FLD00380
FLINE=LINSTR-MOD((LINSTR-IFRST),NDSPR) FLD00390
LSKIP=((FLINE-FSCAN)/NDSPR-1)*NRPS FLD00400
IF (LSKIP) 135,134,138 FLD00410
135 FSKIP = ((BLOCK(1) - IFRST) / NDSPR) * NRPS * 1 FLD00420
C* FLD00430
C* ON MULTI-FILE TAPES FOR FILES OTHER THAN FILE 1, DO THE FOLLOWING- FLD00440
C* 1. HACK SPACE 1 FILE FLD00450
C* 2. HEAD FORWARD 1 E=0-F FLD00460
C* 3. HEAD FORWARD NO. OF DESIRED RECORDS FLD00470
IF (FILENO.EQ. 0 .AND. FSKIP .LE. IABS(LSKIP)) GO TO 136 FLD00480
FSKIP = IABS(LSKIP) FLD00490
DO 134 II=1,FSKIP FLD00500
134 BACKSPACE IUNIT FLD00510
GO TO 139 FLD00520
C* FLD00530
C* FOR FILE 1 DO A REWIND AND SKIP FORWARD THE DESIRED NO. OF RECORDS FLD00540
C* FLD00550
136 REWIND IUNIT FLD00560
DO 137 II=1,FSKIP FLD00570
137 READ(IUNIT,460)DUMMY FLD00580
GO TO 139 FLD00590
C* FLD00600
C* SKIP DOWN THE TAPE TO BEGINNING LINE OF THIS FIELD. FLD00610
C* FLD00620
138 IF (FSCAN.EQ.FLINE) GO TO 140 FLD00630
IF ( LSKIP .EQ. 0) GO TO 139 FLD00640
DO 141 II=1,LSKIP FLD00650
141 READ(IUNIT,480) DUMMY FLD00660
139 CONTINUE FLD00670
FSCAN=FLINE FLD00680
140 CONTINUE FLD00690
NSCAN=LINSTR FLD00700
IF (BLOCK(5).LE.NS)GO TO 145 FLD00710
WRITE(6,440)NS FLD00720
CALL CMERR FLD00730
145 IF (BLOCK(4).LE.NS)GO TO 146 FLD00740
WRITE(6,440)NS FLD00750
FLD00760

```

FILE FLDINT

```

C      CALL CMERR
C      SKIP FORWARD NECESSARY RECORDS FOR LANDSAT 1 OR 2
200    DO 210 I=1,LINSTR
      READ(I,UNIT,480)DUMMY
210    CONTINUE
      GO TO 146
C      SKIP RECORDS FOR LANDSAT III
220    SKIP=NRPDS*(LINSTR-1)
      IF(SKIP.EQ.0)GO TO 146
      DO 230 I=1,SKIP
      READ(I,UNIT,480)DUMMY
230    CONTINUE
C      146 CONTINUE
      LINC=BLOCK(2)
      LINC=BLOCK(3)
      SAMSTR=BLOCK(4)
      SAMEND=BLOCK(5)
      SAMINC=BLOCK(6)
C      LINC=NO. OF RECORDS TO SKIP AFTER EACH SCAN LINE
      LINC=(LINC/NDSPR - 1)*NRPDS
      IF(LINC.LT.0)LINC=0
C      ESTABLISH AREAS ON EACH SCAN LINE TO UNPACK
      IF(FORMAT.EQ.3)GO TO 1000
      IF(FORMAT.EQ.4)GO TO 2000
      ANCLNG = SAMSTR * SVD - 1
      IF (FORMAT.EQ. 1) ANCLNG = ANCLNG * 2
      NBUFS=NRPDS/10
      IF(MOD(NRPDS,10).NE.0)NBUFS=NBUFS+1
      FC=1
      LC=NCAR
      K=1
      DO 190 I=1,NOFEAT
      TRY = 0
185    CONTINUE
      DO 170 JEC=K,NRPDS
      IF (JEC.GT.1)ANCLNG = ANCLNG * 2 + SAMSTR * SVD - 1
      IF (FETVEC(I).GE.FC.AND.FETVEC(I).LE.LC) GO TO 150
      IF (FETVEC(I).GT.LC.AND.JEC.LT.NRPDS) GO TO 160
      FC = 1
      LC = NCAR
      K = 1
      ANCLNG = (ANCLNG + 2) * SAMSTR * SVD - 1
      TRY = TRY + 1
      IF (TRY.LE.2) GO TO 185
      WRITE (4,380)FETVEC(I)
      CALL CMERR
150    IBYTE(I)=(FETVEC(I)-FC)*NS + ANCLNG
      JREC(I)=MOD(IJEC,10)
      IF (JREC(I).EQ.0) JREC(I)=10
      LBUF(I)=IJEC/10 + 1
      IF(MOD(IJEC,10).EQ.0)LBUF(I)=LBUF(I)-1
      GO TO 140
160    FC=LC+1
      LC=LC+NCPR
170    CONTINUE
180    K=JEC
190    CONTINUE
C      NSAMP - NO. OF SAMPLES TO UNPACK FOR EACH FEATURE IN FETVEC
C      NSAMP = (SAMEND - SAMSTR) / SAMINC + 1
900    RETURN
C      SET UP IRYTE FOR LANDSAT 1 OR 2
C      1000
      JREC(1)=SAMSTR
      NSCAN=LINSTR
      FSCAN=LINSTR
      DO 1100 I=1,NOFEAT

```

LANDSAT IS  
FOR QUALITY

FLD00770  
FLD00780  
FLD00790  
FLD00800  
FLD00810  
FLD00820  
FLD00830  
FLD00840  
FLD00850  
FLD00860  
FLD00870  
FLD00880  
FLD00890  
FLD00900  
FLD00910  
FLD00920  
FLD00930  
FLD00940  
FLD00950  
FLD00960  
FLD00970  
FLD00980  
FLD00990  
FLD01000  
FLD01010  
FLD01020  
FLD01030  
FLD01040  
FLD01050  
FLD01060  
FLD01070  
FLD01080  
FLD01090  
FLD01100  
FLD01110  
FLD01120  
FLD01130  
FLD01140  
FLD01150  
FLD01160  
FLD01170  
FLD01180  
FLD01190  
FLD01200  
FLD01210  
FLD01220  
FLD01230  
FLD01240  
FLD01250  
FLD01260  
FLD01270  
FLD01280  
FLD01290  
FLD01300  
FLD01310  
FLD01320  
FLD01330  
FLD01340  
FLD01350  
FLD01360  
FLD01370  
FLD01380  
FLD01390  
FLD01400  
FLD01410  
FLD01420  
FLD01430  
FLD01440  
FLD01450  
FLD01460  
FLD01470  
FLD01480  
FLD01490  
FLD01500  
FLD01510  
FLD01520

FILE FLDINT

1100	IHYTE(I)=1+(FETVEC(I)-1)*2	FL001	530
	CONTINUE	FL001	540
	GO TO 900	FL001	550
C		FL001	560
C	SET UP IHYTE AND NSCAN FOR LANDSAT III	FL001	570
C	IHYTE IS FETVEC FOR LANDSAT III	FL001	580
2000	DO 2100 I=1,NOFEAT	FL001	590
	IHYTE(I)=FETVEC(I)	FL001	600
2100	CONTINUE	FL001	610
	NSCAN=LINSTR	FL001	620
	FSCAN=LINSTR	FL001	630
	LHUF(I)=SAMSTR	FL001	640
	GO TO 900	FL001	650
380	FORMAT(' FEATURE NUMBERS',I5,' AND ABOVE ARE NOT ON DATA TAPE',/	FL001	660
•)		FL001	670
430	FORMAT(' FIRST SCAN ON THIS TAPE IS NUMBERED',I6,' FIELD DEFINITIO	FL001	680
•N IN ERROR')		FL001	690
440	FORMAT(' NUMBER OF SAMPLES UP PER SCAN ON THIS TAPE IS',I6,' FIELD	FL001	700
• DEFINITION IN ERROR')		FL001	710
470	FORMAT(' THIS TAPE CONTAINS ONLY',I6,' CHANNELS')	FL001	720
480	FORMAT(I4)	FL001	730
	END	FL001	740
		FL001	750



FILE: FLTNUM

```
      GO TO 60
50  WNUM = 10*WNUM+MORNUM
60  CONTINUE
C
      COL = CRDSIZ+1
      GO TO 90
70  VECFIN = VECPOS+ITER-1
      IF ( VECFIN .GT. VECMAX ) VECFIN = VECMAX-1
      DO 80 I=VECPOS,VECFIN
80  NUMVEC(I) = SIGN*(WNUM+PNUM)
      L = COL+1
      VECPOS = VECFIN+1
      IF (VECPOS.LE.VECMAX) GO TO 10
      GO TO 110
90  COL = COL-1
      VECFIN = VECPOS+ITER-1
      IF ( VECFIN .GT. VECMAX ) VECFIN = VECMAX
      DO 100 I=VECPOS,VECFIN
100 NUMVEC(I) = SIGN*(WNUM+PNUM)
110 FLTNUM = VECFIN
C  WRITE(6,706) (CARD(K),K=1,62), COL,FLTNUM,(NUMVEC(K),K=1,FLTNUM)
C 706 FORMAT(' FLTNUM ENTERED'/ ' ', 62A1,I10/' ', 15,20F8.2,/' ', 10F8.2)
      RETURN
      END
```

FLT00800  
FLT00810  
FLT00820  
FLT00830  
FLT00840  
FLT00850  
FLT00860  
FLT00870  
FLT00880  
FLT00890  
FLT00900  
FLT00910  
FLT00920  
FLT00930  
FLT00940  
FLT00950  
FLT00960  
FLT00970  
FLT00980  
FLT00990  
FLT01000  
FLT01010  
FLT01020  
FLT01030



FILE: FSRFSL

```
SUBROUTINE FSRFSL(UNIT,FILE,ISTAT)
IMPLICIT INTEGER (A-Z)
N=0
C
  ISTAT = 0
  IF (FILE .EQ. 0) RETURN
  IF (FILE .LT. 0) GO TO 100
C
C
  MOVE UNIT FORWARD N E-O-F'S
40 READ(UNIT,END=50)
   GO TO 40
50 N = N + 1
   IF (N .EQ. FILE) RETURN
   GO TO 40
C
100 WRITE(6,110)
110 FORMAT(' FSRFSL ONLY SKIPS FORWARD*')
   ISTAT = 2
   RETURN
C
END
```

```
FSR00010
FSR00020
FSR00030
FSR00040
FSR00050
FSR00060
FSR00070
FSR00080
FSR00090
FSR00100
FSR00110
FSR00120
FSR00130
FSR00140
FSR00150
FSR00160
FSR00170
FSR00180
FSR00190
FSR00200
FSR00210
FSR00220
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: FFSMFL

```
SUBROUTINE FFSMFL(UNIT,FILE,ISTAT)
IMPLICIT INTEGER (A-Z)
N=0
C
  ISTAT = 0
  IF (FILE .EQ. 0) RETURN
  IF (FILE .LT. 0) GO TO 100
C
  MOVE UNIT FORWARD N E-O-F'S
C
  40 READ(UNIT,30,END=50) DUMMY
  30 FORMAT(1A4)
  GO TO 40
  50 N = N + 1
  IF (N .EQ. FILE) RETURN
  GO TO 40
C
  100 WRITE(6,110)
  110 FORMAT(' FFSMFL ONLY SKIPS FORWARD!')
  ISTAT = 2
  RETURN
C
  END
```

FSB00010  
FSB00020  
FSB00030  
FSB00040  
FSB00050  
FSB00060  
FSB00070  
FSB00080  
FSB00090  
FSB00100  
FSB00110  
FSB00120  
FSB00130  
FSB00140  
FSB00150  
FSB00160  
FSB00170  
FSB00180  
FSB00190  
FSB00200  
FSB00210  
FSB00220  
FSB00230

FILE: GETINF

```
      SURROUTINE GETINF (ARRAY, FLDSAV, VERTEX, CLSNMS, NOSUBS, SUBNM, NOCLS,  
      * TOTSUB)  
C  
      IMPLICIT INTEGER (A-Z)  
      DIMENSION ARRAY (1), FLDSAV (4, 1), VERTEX (1), CLSNMS (1), NOSUBS (1),  
      * SUBNM (1), CLSVEC (60)  
C  
      JJ = 0  
      KPT = 1  
      NFS = 0  
      L = 0  
C  
      SEE SURROUTINE RDATA FOR STORAGE ARRANGEMENT OF 'ARRAY'  
C  
      DO 80 CLS=1, NOCLS  
      L = L + NFS  
      CLSNMS (CLS) = ARRAY (KPT)  
      NOSUBS (CLS) = ARRAY (KPT+2)  
      NFS = ARRAY (KPT+3)  
      IKP = KPT + 4  
C  
      DO 100 I=1, NFS  
      FLDSAV (1, I+L) = ARRAY (IKP)  
      FLDSAV (2, I+L) = CLS  
      FLDSAV (3, I+L) = 0  
      FLDSAV (4, I+L) = ARRAY (IKP+1)  
C  
      NV = FLDSAV (4, I+L) * 2  
C  
      DO 90 J=1, NV  
90  VERTEX (JJ+J) = ARRAY (IKP+1+J)  
      JJ = NV + JJ  
C  
100  CONTINUE  
C  
      IKP = IKP + NV + 9  
C  
      KPT = ARRAY (KPT+1)  
C  
80  CONTINUE  
C  
      K = 0  
      DO 120 I=1, NOCLS  
      NSUB = NOSUBS (I)  
C  
      DO 120 J=1, NSUB  
      K = K + 1  
120  CLSVEC (K) = I  
C  
      CALL NAMSTA (SUBNM, CLSVEC, NOSUBS, TOTSUB, CLSNMS, NOCLS)  
C  
      RETURN  
C  
      END
```

FILE: GETST

```

SUBROUTINE GETST(UNIT,FILE,MENS,STDEV,NOSUB2,SUBVEC,NOCHAN,
* CHNVEC,MEANS,COVAR,ITRIG)
.....
SUBROUTINE GETSTA RETRIEVES THE MEANS AND STANDARD DEVIATIONS
FROM A STATISTICS FILE IN THE LARSYS 'SAVTAP' FORMAT.
INPUT ARGUMENTS:
UNIT - FORTPAN UNIT NUMBER FROM WHICH THE STATS ARE TO BE
RETRIEVED.
FILE - FILE NO. ON 'UNIT' FROM WHICH THE STATS ARE TO BE
RETRIEVED.
NOCHAN - NO. OF CHANNELS REQUESTED FROM TRAINING SEGMENT,
NOCHAN LESS THAN OR = NO. OF CHANNELS ON STAT FILE
CHNVEC - ARRAY CONTAINING ACTUAL CHANNELS REQUESTED FROM
TRAINING SEGMENT. MUST BE A SUBSET OF CHANNELS
ON STAT FILE
ITRIG - IF ITRIG=1, ST. DEV. WILL BE RETURNED ALONG WITH MEANS.
OUTPUT ARGUMENTS:
MENS - ARRAY CONTAINING THE MEAN VECTORS FOR EACH SUBCLASS
(A SUBSET OF THE CHANNELS MAY BE SELECTED, BUT NOT
A SUBSET OF THE SUBCLASSES).
STDEV - ARRAY CONTAINING THE SUBSET OF ST. DEV. FOR REQUES-
TED CHANNELS IN EACH SUBCLASS
IN EACH SUBCLASS.
NOSUB - NUMBER OF SUBCLASSES ON THE STAT FILE
CONTINUE
CHNVEC - ARRAY CONTAINING ACTUAL CHANNELS REQUESTED FROM
TRAINING SEGMENT
.....
**NOTE** THE STORAGE ARRAYS PASSED TO THIS SUBROUTINE FOR THE
MEANS AND STANDARD DEVIATIONS SHOULD BE SINGLY DIMENSIONED
IN THE CALLING ROUTINE. ON OUTPUT THE ITEMS ARE STORED
AS FOLLOWS: (SAME FOR STDEV)
MEANS(1) - CHANNEL 1, SUBCLASS 1
(2) - CHANNEL 2, SUBCLASS 1
3 - CHANNEL 3, SUBCLASS 1
.
.
(NOCHAN) - CHANNEL NOCHAN, SUBCLASS 1
(NOCHAN+1) - CHANNEL 1, SUBCLASS 2
(NOCHAN+2) - CHANNEL 2, SUBCLASS 2
(NOCHAN+3) - CHANNEL 3, SUBCLASS 2
.
.
(2*NOCHAN) - CHANNEL NOCHAN OF SUBCLASS 2
.
.
ETC.
THRU
CONTINUE (NOCHAN*NOSUB)
.....
IMPLICIT INTEGER (A-Z)
DIMENSION CHNVEC(30),DUMVEC(30),CHNVEC(1)
DATA BLANK/' '/
DIMENSION SUBVEC(1)
DIMENSION MEANS(1),STDEV(1)
REAL MEANS,STDEV,COVAR(465)
REAL MENS(1)
REWIND UNIT
NF=FILE-1
CALL F99SFL(UNIT,NF,ISTAT1)
IF(ISTAT1.EQ.0)GO TO 5
WRITE(6,100)UNIT,FILE
100 FORMAT(1X,'ERROR IN POSITIONING UNIT',I3,' TO FILE',I3)
CALL CMERR

```

GET00010  
GET00020  
GET00030  
GET00040  
GET00050  
GET00060  
GET00070  
GET00080  
GET00090  
GET00100  
GET00110  
GET00120  
GET00130  
GET00140  
GET00150  
GET00160  
GET00170  
GET00180  
GET00190  
GET00200  
GET00210  
GET00220  
GET00230  
GET00240  
GET00250  
GET00260  
GET00270  
GET00280  
GET00290  
GET00300  
GET00310  
GET00320  
GET00330  
GET00340  
GET00350  
GET00360  
GET00370  
GET00380  
GET00390  
GET00400  
GET00410  
GET00420  
GET00430  
GET00440  
GET00450  
GET00460  
GET00470  
GET00480  
GET00490  
GET00500  
GET00510  
GET00520  
GET00530  
GET00540  
GET00550  
GET00560  
GET00570  
GET00580  
GET00590  
GET00600  
GET00610  
GET00620  
GET00630  
GET00640  
GET00650  
GET00660  
GET00670  
GET00680  
GET00690  
GET00700  
GET00710  
GET00720  
GET00730  
GET00740  
GET00750  
GET00760  
GET00770  
GET00780  
GET00790

FILE: GETST

```
5 CONTINUE
  READ(UNIT)NOCLS,NOSUB,NCHAN,NOFLD,TOTVRT,(CHNVC1(I),I=1,NCHAN)
C
C**  DEFAULT ALL SUBCLASSES FROM STATISTICS FILE
C**
  IF(NOSUB2.NE.0)GO TO 7
  DO 6 I=1,NOSUB
6    SURVEC(I)=I
  NOSUB2=NOSUB
7    CONTINUE
  DO 77 I=1,NOSUB2
  IF (SURVEC(I).LE. NOSUB) GO TO 77
  WRITE(6,70)NOSUB
70  FORMAT(/' REQUESTED SUBCLASS IS NOT ON STAT FILE. STAT FILE CONTAINS
  *NS ' ,I3,' SUBCLASSES')
  CALL CMERR
77 CONTINUE
C**
C**  DEFAULT -- ALL CHANNELS FROM STAT FILE
C**
  IF (NOCHAN .NE. 0) GO TO 9
  DO 8 I=1,NCHAN
8    CHNVEC(I) = CHNVC1(I)
  NOCHAN = NCHAN
9    CONTINUE
C**
C**  READ PAST THE TRAINING FIELD INFORMATION
C**
  DO 10 I=1,NOFLD
  READ(UNIT)DUM
  READ(UNIT)DUM
10  CONTINUE
  READ(UNIT)DUM
  VARSIZ = NCHAN*(NCHAN+1)/2
  MB=1
  ME = NCHAN
  IC=1
  DO 30 J=1,NOSUB
  READ(UNIT) N,(COVAR(J),J=1,VARSIZ),(MEANS(J),J=MB,ME)
  IF(SURVEC(IC).NE.I)GO TO 30
  IF(ITRIG.NE.1)GO TO 25
C**
C**  GET STANDARD DEVIATIONS
C**
  JK=0
  DO 20 JA=1,NOCHAN
  JK = JK + JA
  STDEV(MB+JA-1) = SQRT(COVAR(JK))
20  CONTINUE
25  CONTINUE
  IC = IC + 1
  MB = MB + NCHAN
  ME = ME + NCHAN
30  CONTINUE
C**
C**  GET SURSET OF MEANS
  AND GET SUBSET OF ST. DEV.
C**
  DO 50 J=1,NOCHAN
  DO 40 K=1,NCHAN
  IF (CHNVEC(J).EQ. CHNVC1(K)) GO TO 50
40  CONTINUE
  WRITE(6,110)CHNVEC(J),(CHNVC1(L),L=1,NCHAN)
110 FORMAT(' CHANNEL NO. ',I2,' IS NOT ON TRAINING STAT FILE. CHANNELS
  1 ARE %10X.30(I2,1X))
  CALL EXIT
50  DUMVEC(J) = K
C
  JJJ = 0
  DO 60 K=1,NOSUB2
  DO 60 J=1,NOCHAN
  JJ = DUMVEC(J) + (K-1)*NCHAN
  JJJ = JJJ + 1
  MFNS(JJJ) = MEANS(JJ)
60  IF ( ITRIG.NE. 0) STDEV(JJJ) = STDEV(JJ)
  WRITE(6,200)
200 FORMAT(/'I57,'MEANS')
  ISTART = 1
  IEND = 12
```

GET00800  
GET00810  
GET00820  
GET00830  
GET00840  
GET00850  
GET00860  
GET00870  
GET00880  
GET00890  
GET00900  
GET00910  
GET00920  
GET00930  
GET00940  
GET00950  
GET00960  
GET00970  
GET00980  
GET00990  
GET01000  
GET01010  
GET01020  
GET01030  
GET01040  
GET01050  
GET01060  
GET01070  
GET01080  
GET01090  
GET01100  
GET01110  
GET01120  
GET01130  
GET01140  
GET01150  
GET01160  
GET01170  
GET01180  
GET01190  
GET01200  
GET01210  
GET01220  
GET01230  
GET01240  
GET01250  
GET01260  
GET01270  
GET01280  
GET01290  
GET01300  
GET01310  
GET01320  
GET01330  
GET01340  
GET01350  
GET01360  
GET01370  
GET01380  
GET01390  
GET01400  
GET01410  
GET01420  
GET01430  
GET01440  
GET01450  
GET01460  
GET01470  
GET01480  
GET01490  
GET01500  
GET01510  
GET01520  
GET01530  
GET01540  
GET01550  
GET01560  
GET01570  
GET01580

FILE: GETST

```
LOOPCT = NOCHAN/12
LOOPC1 = MOD(NOCHAN,12)
IF (LOOPC1 .GT. 0) LOOPCT = LOOPCT + 1
IF (LOOPCT .EQ. 1) IEND = NOCHAN
DO 240 I=1,LOOPCT
  START = ISTART
  IEND = IEND
  WRITE (6,210) (BLANK,CHNVEC(I),I=ISTART,IEND)
210  FORMAT(/2X,'CLUSTER',5X,12(A1,'CH(',12,')'),2X)
  DO 230 J=1,NOSUB2
  WRITE (6,220) SURVEC(J), (MENS(K),K=START,END)
220  FORMAT(5X,12,7X,12(F7.2,2X))
  START = ISTART + NOCHAN*J
  IEND = IEND + NOCHAN*J
230  CONTINUE
  WRITE (6,235)
235  FORMAT(/)
  ISTART = IEND + 1
  IEND = IEND + ISTART - 1
  IF (IEND .GT. NOCHAN) IEND = NOCHAN
240  CONTINUE
  RETURN
END
```

```
GET 01590
GET 01600
GET 01610
GET 01620
GET 01630
GET 01640
GET 01650
GET 01660
GET 01670
GET 01680
GET 01690
GET 01700
GET 01710
GET 01720
GET 01730
GET 01740
GET 01750
GET 01760
GET 01770
GET 01780
GET 01790
GET 01800
GET 01810
```

FILE: GRPSCN

```
C      FUNCTION GRPSCN(CARD,NNCLAS,GRPTR)                                GRP00010
C      IMPLICIT INTEGER (A-H,O-Z)                                       GRP00020
C      -----                                                            GRP00030
C      -----                                                            GRP00040
C      -----                                                            GRP00050
C      -----                                                            GRP00060
C      -----                                                            GRP00070
C      -----                                                            GRP00080
C      CALL..    CALL GRPSCN(CARD,NNCLAS,GRPTR)                          GRP00090
C      ARGV..    CAPD   = 62 COL CARD BUFFER                               GRP00100
C      NNCLAS = MAX NO OF CLASSES TO ALLOW                               GRP00110
C      GRPTR  = PTR TO 'GROUPS'                                          GRP00120
C      -----                                                            GRP00130
C      REQUIRES. COMMONS /INFORM/ /INFORS/ /DISCOM/                       GRP00140
C      ROUTINES  N1TCHR  FIXUP    NUMBER                                GRP00150
C      -----                                                            GRP00160
C      PURPOSE.. SCANS ALL 'GROUP' (TRAIN/TEST) CARDS                    GRP00170
C      AND SET UP 'GRPDEX','GRPNAM','GROUPS'                             GRP00180
C      -----                                                            GRP00190
C      RETURNS.. GRPTR  - SEE ARGV                                        GRP00200
C      -----                                                            GRP00210
C      -----                                                            GRP00220
C      -----                                                            GRP00230
C      -----                                                            GRP00240
C      -----                                                            GRP00250
C      -----                                                            GRP00260
C      -----                                                            GRP00270
C      -----                                                            GRP00280
C      -----                                                            GRP00290
C      INCLUDE COMRK1,LIST                                               GRP00300
C      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARZ2,TOTVT2,NOFLD2,          GRP00310
C      *      AVAR2,COVAR2,CLS1D2,SUBNO2,SUBDS2,FLOSV2,VERTX2,          GRP00320
C      *      FETVC2(30),SUBVC2(75),SUBPTR(75),CLSVC2(60),            GRP00330
C      *      KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),                  GRP00340
C      *      GRPCHK(61),GROUPS(124)                                    GRP00350
C      -----                                                            GRP00360
C      SEND                                                                GRP00370
C      -----                                                            GRP00380
C      -----                                                            GRP00390
C      DIMENSION RUF(4), CARD(62), COMVEC(2), NUMVEC(30)                GRP00400
C      LOGICAL*1 LCHAR(4)                                                GRP00410
C      DIMENSION ICHAR(1)                                                GRP00420
C      EQUIVALENCE (LCHAR(1),ICHA(1))                                    GRP00430
C      LOGICAL*1 LLCHAR(4)                                               GRP00440
C      DIMENSION IICHAR(1)                                               GRP00450
C      EQUIVALENCE (LLCHAR(1),IICHA(1))                                  GRP00460
C      -----                                                            GRP00470
C      DATA BLANK/' ', COMMA/',', COMVEC/1,0,0'                        GRP00480
C      -----                                                            GRP00490
C      -----                                                            GRP00500
C      -----                                                            GRP00510
C      -----                                                            GRP00520
C      -----                                                            GRP00530
C      COL = 0                                                            GRP00540
C      GRPSCN = 1                                                         GRP00550
C      J = NXTCHR(CARD,COL)                                               GRP00560
C      IF (J.EQ.BLANK) GO TO 110                                          GRP00570
C      DO 10 I=1,4                                                         GRP00580
C      J2 = CARD(COL-1+I)                                                 GRP00590
C      IF (J2.EQ.COMMA) GO TO 20                                          GRP00600
C      10 RUF(I) = J2                                                      GRP00610
C      GO TO 40                                                            GRP00620
C      -----                                                            GRP00630
C      20 DO 30 J=I,4                                                      GRP00640
C      30 RUF(J) = BLANK                                                  GRP00650
C      -----                                                            GRP00660
C      40 N = ABS(0.0)                                                     GRP00670
C      DO 50 I=1,4                                                         GRP00680
C      IICHAR(I)=RUF(I)                                                  GRP00690
C      50 LCHAR(I)=LLCHAR(I)                                              GRP00700
C      WRD1=ICHA(1)                                                       GRP00710
C      GRPNAM(NOGRP+1) = WRD1                                             GRP00720
C      J = FIND12(CARD,COL,COMVEC)                                        GRP00730
C      IF (J.LE.0) GO TO 110                                              GRP00740
C      J = NUMBER(CARD,COL,NUMVEC,0)                                       GRP00750
C      II = 0                                                                GRP00760
C      LAST = 0                                                            GRP00770
C      DO 90 I=1,J                                                         GRP00780
C      JJ = NUMVEC(I)                                                      GRP00790
C      IF (JJ.GT.LAST.AND.JJ.LE.NNCLAS.AND.GRPCHK(JJ).EQ.0) GO TO 80   GRP00800
C      WRITE(6,70) JJ, JJ, CARD                                           GRP00810
C      70 FORMAT(// 5X, '///// FROM SUBR. GRPSCN --- CLASS ',I5,      GRP00820
```

FILE: GRPSCN

```
1  INCORRECT --- CLASS '15' IGNORED // 11X CARD BEING SCANNED GRP00800
2 ED IS ... /9X,2H... 62A1, 2H... // GRP00810
GO TO 90 GRP00820
90  I = I + 1 GRP00830
    NUMVEC(I) = JJ GRP00840
    GRPCHK(JJ) = I GRP00850
    LAST = JJ GRP00860
90  CONTINUE GRP00870
    IF (II.LE.0) GO TO 110 GRP00880
C GRP00890
    NOGRP = NOGRP + 1 GRP00900
    GRPTR = GRPTR + 1 GRP00910
    GRPDEX(NOGRP) = GRPTR GRP00920
    GROUPS(GRPTR) = II GRP00930
    DO 100 I=1,II GRP00940
100 GROUPS(GRPTR+I) = NUMVEC(I) GRP00950
    GRPTR = GRPTR + II GRP00960
    GRPSCN = 0 GRP00970
    RETURN GRP00980
C GRP00990
110 RETURN GRP01000
    END GRP01010
    GRP01020
```



FILE: HISTGM

```

SURROUTINE HISTGM(FILHIS,FLDTAL,TOTTAL)
.....
PURPOSE  -- CALCULATES HISTOGRAMS AND WRITES TOTAL HISTOGRAMMED
           STATISTICS ON UNIT 13 TO BE READ BY GRAYMP PROCESSOR
.....
IMPLICIT INTEGER(A-T)
INCLUDE COMMK3.LIST
INCLUDE COMMK4.LIST
INCLUDE COMMK6.LIST
COMMON /GHCBLK/MAXFET,NOFEAT,NOFET2,FETVEC(30),
       FETVC2(30),FLOINF(6),INFMT,FILESV,NOHIST,
       HISVEC(30),NOFLD,        FLOPTS
*
* .XSIZ,XLOW,XHGH,YSIZ
* DIMENSION MED1(15),MED2(15),DATE(3),COMENT(15)
* EQUIVALENCE (MED1(1),HEAD(4)),(DATE(1),HEAD(22)),
*              (MED2(1),HEAD(30)),(COMENT(1),HEAD(48))
2  COMMON/GLOBAL/HEAD(63),MAPTAP,DATEPE,SAVTAP,BMFILE,HMKEY,
*              HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
*              DRUMAD,DHMUS,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVFL
*              .NHSTUN,NHSTFI,SCTRUN,MAPFIL
*              .DOTUNT,DOTFIL,NCMPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
*              CRDUNT,PRUNT,RANDIO
CSEND
COMMON /HISTOR/HF
C
INTEGER XSIZ,YSIZ,XHGH,XLOW
INTEGER VERTCS
DIMENSION DATA(12000),FILHIS(NOFEAT,256),
* FLDTAL(NOHIST,XSIZ),TOTTAL(NOHIST,XSIZ),IFLD(50,24),
* VERTCS(2,11),FL(A)
DATA HLANK/' ',DIM/12000/
DATA TOTAL,'TOTA'
EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND),
*           (FLDINF(3),LININC),(FLDINF(4),SAMSTR),
*           (FLDINF(5),SAMEND),(FLDINF(6),SAMINC)
CALL TAPHDR(DATEPE,DATAFIL)
HISKEY=1
FILESV = 0
ISWTH = 0
DO 7   J=1,XSIZ
DO 7   I=1,NOHIST
7  FLDTAL(I,J) = 0
TOTTAL(I,J) = 0
NOFLD=0
IF(HF.NE.1) GO TO 10
NC=5
VERTCS(1,1)=1
VERTCS(2,1)=1
VERTCS(1,2)=200
VERTCS(2,2)=1
VERTCS(1,3)=200
VERTCS(2,3)=500
VERTCS(1,4)=1
VERTCS(2,4)=500
VERTCS(1,5)=1
VERTCS(2,5)=1
FLDNAM=BLANK
FLDINF(1)=1
FLDINF(2)=500
FLDINF(3)=10
FLDINF(4)=1
FLDINF(5)=200
FLDINF(6)=10
GO TO 15
READ IN FIELD CARDS
10 ICK=LAREAD(FLDNAM,VERTCS,FLDINF,NC)
IF(ICK.EQ.1) GO TO 15
IF(ICK.EQ.0) GO TO 60
IF(ICK.LE.-1) GO TO 10
15 NSAMP = ( FLDINF(5) - FLDINF(4) ) / FLDINF(6) + 1
CHECK DATA DIMENSIONS. IF TOO MUCH DATA REQUESTED, RESET SAMPLE END
TOTPTS = NSAMP*NOFEAT

```



FILE: HISTOM

```
33 CONTINUE
   IP = 0
   CALL HISTIC(FILMIS,IP,IFLD,      VERTCS,NC)
   WRITE(HISFIL) ((FILMIS(I,J),J=1,256),I=1,NOFEAT)
   IF (MF.FO.1) GO TO 60
   IF (NOMIST.EQ.0) GO TO 10
C
   CALL PLOTTING ROUTINE TO PLOT HISTOGRAM FOR THE FIELDS
C
   CALL HISTGRM(FLOTAL,IDATA,FLDNAM,2,XSIZ,XHGH,XLOW,YSIZ,
•NOMIST,FLOPTS,HISVEC)
   GO TO 10
C
   WRITE TOTAL HIST ON TAPE --UNIT 13
C
60 CONTINUE
   REWIND HISFIL
   DO 104 I=1,NOFEAT
   DO 104 J=1,256
104 FILMIS(I,J)=0
   DO 107 K=1,NOFLD
   JK=NOFEAT+256
   READ(HISFIL) (IDATA(I),I=1,JK)
   M=0
   DO 108 I=1,NOFEAT
   DO 108 J=1,256
   M=M+1
108 FILMIS(I,J)=IDATA(M)+FILMIS(I,J)
107 CONTINUE
   REWIND HISFIL
   WRITE(HISFIL) NOFEAT, (FETVEC(I),I=1,NOFEAT)
   WRITE(HISFIL) ((FILMIS(I,J),J=1,256),I=1,NOFEAT)
   REWIND HISFIL
C
   PRINT TOTAL STATS
C
   IP = -1
   CALL HISTIC(FILMIS,IP,IFLD,      VERTCS,NC)
   IF (NOMIST.EQ.0) RETURN
   IF (NOFLD.EQ.1) RETURN
C
   CALL PLOTTING ROUTINE TO PLOT TOTAL HISTOGRAM
C
   CALL HISTGRM(TOTAL,IDATA,TOTAL,3,XSIZ,XHGH,XLOW,YSIZ,
•NOMIST,FLOPTS,HISVEC)
   RETURN
C 200 FORMAT(13A6,A2)
   END
```

HIS01590  
HIS01600  
HIS01610  
HIS01620  
HIS01630  
HIS01640  
HIS01650  
HIS01660  
HIS01670  
HIS01680  
HIS01690  
HIS01700  
HIS01710  
HIS01720  
HIS01730  
HIS01740  
HIS01750  
HIS01760  
HIS01770  
HIS01780  
HIS01790  
HIS01800  
HIS01810  
HIS01820  
HIS01830  
HIS01840  
HIS01850  
HIS01860  
HIS01870  
HIS01880  
HIS01890  
HIS01900  
HIS01910  
HIS01920  
HIS01930  
HIS01940  
HIS01950  
HIS01960  
HIS01970  
HIS01980  
HIS01990  
HIS02000  
HIS02010  
HIS02020  
HIS02030  
HIS02040  
HIS02050  
HIS02060

FILE: HISTIC

```

SURROUTINE HISTIC (IMG,NI,IFLD, VERTCS,NC)
PURPOSE -- COMPUTES AND DISPLAYS STATS FOR HISTOGRAM ROUTINE
INCLUDE COMRK3.LIST
INCLUDE COMRK4.LIST
INCLUDE COMRK5.LIST
COMMON /GRCHLR/MAXFET,NOFEAT,NOFET2,FETVEC(30),
      FETVC2(30),FLDINF(6),INFMT,FILESV,NOHIST,
      HISVEC(30),NOFLD, FLOPTS
      *XSIZ,XLOW,XHIGH,YSIZ
      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
      EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
      (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
COMMON /GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
      HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
      DRUMAD,DRMUDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
      *NHSTUN,NHSTFI,SCTHUN,MAPFIL
      *DOTUNT,DOTFIL,NCPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
      CRDUNT,PRUNT,RANDIO
CSEND
DIMENSION RANGE(30,2),ZMEAN(30),STDDV(30),
      *NRANGE(30,2),IMG(NOFEAT,256),IFLD(50,24)
      INTEGER VERTCS(1),OP,CP,COMMA
      DATA OP / ( /,CP / ) /,COMMA / , /
      INTEGER FLDINF,FETVEC,HEAD,FILESV
      REAL NRANGE
      COMPUTES THE DATA RANGE
      DO 10 J = 1,NOFEAT
      DO 20 J = 1,256
      ISTR = J
      IF (IMG(I,J) .NE. 0) GO TO 30
      CONTINUE
      20 RANGE(I,1) = J
      K = 257
      40 K = K - 1
      IF (IMG(I,K) .NE. 0) GO TO 50
      IF (K.GT.ISTR) GO TO 40
      50 RANGE(I,2) = K
      COMPUTES THE MEAN -- (1/N) * ( 1*IMG(1) + 2*IMG(2) + ... N*IMG(N) )
      STDDV1 = 0
      RMEAN = 0
      N = 0
      DO 60 L = 1,256
      N = IMG(I,L) * N
      AMEAN = L * IMG(I,L)
      STDDV = AMEAN * L
      RMEAN = AMEAN + RMEAN
      60 STDDV1 = STDDV + STDDV1
      ZMEAN(I) = RMEAN / N
      STDDV1 = STDDV1 / N
      COMPUTES THE STD. DEV. -- SORT.( (1/N)*(1**2 * IMG(1) + 2**2 * IMG(
      2) + ... + N**2 * IMG(N)) - MEAN**
      2 )
      A = STDDV1 - ZMEAN(I)**2
      STDDV(I) = SQRT(A)
      COMPUTES NORMALIZED RANGE
      NRANGE(I,1) = ZMEAN(I) - 3*STDDV(I)
      10 NRANGE(I,2) = ZMEAN(I) + 3*STDDV(I)
      IN=NC-1
      NNC=2*(IN)
      WRITE(6,HEAD)
      IF (N).EQ.-1) GO TO 45
      WRITE(6,S10)
      DO 70 I1 = 1,NOFEAT
      70 WRITE(6,S20) FETVEC(I1),IFLD(NOFLD,21),IN,FLDINF(6),FLDINF(3),
      * (OP,VERTCS(I1),COMMA,VERTCS(I1+1),CP),J=1,NNC,2)
      WRITE(6,S30)
      75 CONTINUE

```

FILE: HISTIC

```
DO 80 I3 = 1,NOFEAT
80 WRITE(6,540) FETVEC(I3),RANGE(I3,1),RANGE(I3,2),ZMEAN(I3),
*STODEV(I3),NRANGE(I3,1),NRANGE(I3,2)
RETURN
85 WRITE(6,550)
IF(NOFLO.GT.50) NOFLD=50
DO 1 I=1,NOFLD
IN=IFLD(I,24)-1
NNC=IN*2
KJ=10
IF(NNC.LE.10) KJ=NNC
DO 2 J=1,NNC
2 VERTCS(J)=IFLD(I,J)
1 WRITE(6,560) IFLD(I,21),IN,IFLD(I,22),IFLD(I,23),
*((OP,VERTCS(K),COMMA,VERTCS(K+1),CP),K=1,KJ,2)
IF(NNC.LE.10) GO TO 2017
WRITE(6,561) ((OP,VERTCS(K),COMMA,VERTCS(K+1),CP),K=1,NNC,2)
2017 CONTINUE
IF (FILESV.EQ. -2) WRITE(6,565)
WRITE(6,570)
GO TO 75
530 FORMAT(//////// T53,
* HISTOGRAM STATISTICS:// 16X,'CHANNEL',9X,'DATA',
* RANGE',9X,'MEAN', 7X,'STANDARD DEVIATION', 9X,'NORMALIZED',
* RANGE'/ 86X,'(MEAN + AND - 3 STD DEV)')
540 FORMAT(18X,I2,11X,F5.1,2X,F5.1,7X,F5.1,12X,F5.1,17X,F6.1,2X,F6.1)
510 FORMAT(////T53,'DATA BLOCK(S) HISTOGRAMMED'//,
* T24,'NO. OF SAMPLE LINE'//
* T3,'CHANNEL FIELDNAME VERTICES INC INC VERTICES(SAMPLE,LI
*NE))
550 FORMAT(////T53,'DATA BLOCK(S) HISTOGRAMMED'//,
* T24,'NO. OF SAMPLE LINE'//
* T12,'FIELDNAME VERTICES INC INC VERTICES(SAMPLE,LI
*NE))
520 FORMAT(4X,I2,7X,A4,8X,I2,7X,I4,2X,I4,1X,
* 5(A1,I4,A1,I4,A1,2X)/T46.5(A1,I4,A1,I4,A1,2X))
540 FORMAT(13X,A4,8X,I2,7X,I4,2X,I4,1X,
* 5(A1,I4,A1,I4,A1,2X))
541 FORMAT(T46.5(A1,I4,A1,I4,A1,2X))
545 FORMAT(T2,'ONLY THE FIRST 50 FIELD DESCRIPTIONS WERE PRINTED,
* BUT ALL THE FIELDS WERE INCLUDED IN THE TOTAL HISTOGRAMMED STATS',
*)
570 FORMAT(//////// T60,'TOTAL'// T53,
* HISTOGRAM STATISTICS:// 16X,'CHANNEL',9X,'DATA',
* RANGE',9X,'MEAN', 7X,'STANDARD DEVIATION', 9X,'NORMALIZED',
* RANGE'/ 86X,'(MEAN + AND - 3 STD DEV)')
END
```

PRECEDING PAGE NOT BLANK - MISHNUMBER.

FILE: I4A1BN

```

C          SURROUTINE I4A1RN(IFLD,NCHFLD,NCVTE)
C          DAVID LFF SMITH 9 SEPTEMBER 1977.
C          THIS SURROUTINE ACCEPTS AN ARRAY OF EBCDIC CHARACTERS AND CONVERTS
C          EBCDIC DIGITS TO A BINARY INTEGER.
C          CALLING SEQUENCE:
C          CALL I4A1RN( FIELD, LENGTH, OUTPUT )
C          WHERE FIELD IS THE FIRST WORD OF AN ARRAY OF EBCDIC CHARACTERS
C          TO BE CONVERTED TO BINARY. CHARACTERS STORED ONE PER
C          WORD, LEFT JUSTIFIED, AS BY AN A1 FORMAT.
C          LENGTH IS THE NUMBER OF CHARACTERS IN THE FIELD.
C          AND OUTPUT IS THE ONE WORD RESULT.
C          INTEGER * 4 IDUM(2), IFLD(20)
C          LOGICAL * 1 L(8)
C          EQUIVALENCE (L(1),IDUM(1)),(ILCH,IDUM(1)),(ICHR,IDUM(2))
C          DATA ICHAR / 0 /
C          DATA IR0 / 240 /
C          DATA IR9 / 249 /
C          DATA IRBL / 64 /
C          DATA IRPL / 78 /
C          DATA IRMI / 96 /
C          NCVTE = 0
C          IERFLG = 0
C          MINUS = 1
C          DO 30 I = 1, NCHFLD
C             ILCH = IFLD( I )
C             L(A) = L(1)
C             IF ( ICHAR .LT. IR0 ) GO TO 10
C             IF ( ICHAR .GT. IR9 ) GO TO 10
C             JDIG = I
C             GO TO 200
10          NEXT = I + 1
C             IF ( ICHAR .EQ. IRBL ) GO TO 30
C             IF ( ICHAR .EQ. IRPL ) GO TO 100
C             IF ( ICHAR .NE. IRMI ) GO TO 20
C             MINUS = - MINUS
C             GO TO 100
20          IERFLG = I
C          CONTINUE
30          IERFLG = NCHFLD + 1
C             GO TO 240
100         IF ( NEXT .GT. NCHFLD ) GO TO 130
C             DO 120 I = NEXT, NCHFLD
C                ILCH = IFLD( I )
C                L(A) = L(1)
C                IF ( ICHAR .LT. IR0 ) GO TO 110
C                IF ( ICHAR .GT. IR9 ) GO TO 110
C                JDIG = I
C                GO TO 200
110         IF ( ICHAR .EQ. IRBL ) GO TO 120
C             IERFLG = I
C          CONTINUE
120         IERFLG = NCHFLD + 1
C             GO TO 240
130         DO 230 I = JDIG, NCHFLD
C             ILCH = IFLD( I )
C             L(A) = L(1)
C             IF ( ICHAR .LT. IR0 ) GO TO 210
C             IF ( ICHAR .E. IR9 ) GO TO 220
C             IF ( ICHAR .NE. IRBL ) IERFLG = I
C             ICHAR = IR0
C             IVAL = ICHAR - IR0
C             NCVTE = NCVTE * 10 - IVAL
230         CONTINUE
C             IF ( MINUS .EQ. 1 ) NCVTE = - NCVTE
240         IF ( IERFLG .EQ. 0 ) GO TO 250
C             NCH = NCHFLD
C             IF ( NCH .GT. 80 ) NCH = 80
C             WRITE ( 6, 1000 ) IERFLG, NCHFLD, (IFLD(K), K = 1, NCH )
1000        FORMAT ( ' EBCDIC TO BINARY INTEGER CONVERSION ERROR'// ' AT CHARACTER
1          1, IS, OF '15,' CHARACTER FIELD: '//1X,80A1)
250         RETURN
C          END
I4A00010
I4A00020
I4A00030
I4A00040
I4A00050
I4A00060
I4A00070
I4A00080
I4A00090
I4A00100
I4A00110
I4A00120
I4A00130
I4A00140
I4A00150
I4A00160
I4A00170
I4A00180
I4A00190
I4A00200
I4A00210
I4A00220
I4A00230
I4A00240
I4A00250
I4A00260
I4A00270
I4A00280
I4A00290
I4A00300
I4A00310
I4A00320
I4A00330
I4A00340
I4A00350
I4A00360
I4A00370
I4A00380
I4A00390
I4A00400
I4A00410
I4A00420
I4A00430
I4A00440
I4A00450
I4A00460
I4A00470
I4A00480
I4A00490
I4A00500
I4A00510
I4A00520
I4A00530
I4A00540
I4A00550
I4A00560
I4A00570
I4A00580
I4A00590
I4A00600
I4A00610
I4A00620
I4A00630
I4A00640
I4A00650
I4A00660
I4A00670
I4A00680
I4A00690
I4A00700
I4A00710
I4A00720
I4A00730
```

FILE: LARMAN

```
SUBROUTINE LARMAN(UNIT,FILE,NOCLS,TOTSUB,NOFEAT,TOTFLD,TOTVRT,
  FETVEC,FLDSAV,VERTEX,CLSNMS,NOSUBS,SUBNM,N,
  STADRS,VARSI2,PUNCH,SUBVEC,PRNSTS,SWICH)
.....
SUBROUTINE STATFL WILL WRITE A STATISTICS FILE IN THE FORMAT
EXPECTED BY THE LARSYS PROGRAM. IT WILL ALSO PUNCH THE MODULE
DECK IF REQUESTED.
ARGUMENTS ARE ALL INPUT.
UNIT - FORTRAN UNIT NUMBER WHERE THE FILE IS TO BE WRITTEN.
FILE - FILE NUMBER ON 'UNIT' WHERE THE FILE IS TO BE WRITTEN.
NOCLS - NUMBER OF CLASSES.
TOTSUR - TOTAL NUMBER OF SUBCLASSES FOR ALL CLASSES.
NOFEAT - NUMBER OF CHANNELS.
TOTFLD - TOTAL NUMBER OF TRAINING FIELDS.
TOTVRT - TOTAL NUMBER OF VERTICES FOR ALL TRAINING FIELDS.
FETVEC - VECTOR CONTAINING THE CHANNEL NUMBERS FOR WHICH
  THE STATISTICS WERE COMPUTED. (DIMENSIONED - NOFEAT)
FLDSAV - ARRAY DIMENSIONED - 4 BY TOTFLD AND CONTAINING THE
  FOLLOWING INFORMATION FOR EACH TRAINING FIELD.
  FLDSAV(1,I) - NAME OF FIELD I.
  FLDSAV(2,I) - CLASS NO. THAT FIELD I BELONGS TO.
  FLDSAV(3,I) - SUBCLASS NO. THAT FIELD I BELONGS TO.
  ZERO IF THE FIELD IS NOT ASSOCIATED
  WITH A SUBCLASS.
  FLDSAV(4,I) - NO. OF VERTICES FOR THIS FIELD,
  INCLUDING THE CLOSURE POINT.
VERTEX - ARRAY CONTAINING VERTICES FROM ALL TRAINING FIELDS.
  DIMENSIONED - 2*TOTVRT, THE VERTICES SHOULD BE STORED
  SAMPLE NO. FIRST THEN LINE NUMBER FOR EACH VERTEX.
  CLOSURE POINTS MUST BE INCLUDED FOR EACH VERTEX.
  (I.F. THE FIRST VERTEX IS REPEATED AS THE LAST VERTEX)
CLSNMS - ARRAY CONTAINING ALPHANUMERIC CLASS NAMES.
NOSUBS - ARRAY CONTAINING THE NO. OF SUBCLASSES IN EACH CLASS.
SUBNM - ARRAY CONTAINING ALPHANUMERIC SUBCLASS NAMES.
N - ARRAY CONTAINING THE NO. OF PIXELS IN EACH SUBCLASS.
PUNCH - IF PUNCH=1 THE MODULE DECK WILL BE PUNCHED.
SWICH = 1 CALLED BY ISOCLS
SWICH = 2 CALLED BY LABEL
SWICH = 3 CALLED BY STAT
SWICH = 4 CALLED BY TRSTAT
.....
IMPLICIT INTEGER(A-Z)
DIMENSION FETVEC(NOFEAT),FLDSAV(4,TOTFLD),VERTEX(TOTVRT)
DIMENSION CLSNMS(NOCLS),NOSUBS(NOCLS),SUBNM(TOTSUB),N(TOTSUR)
DIMENSION SURVEC(1)
DATA PCHUNT /7/
REAL COVAR(465),MEANS(30)
POSITION 'UNIT' TO CORRECT FILE NO.
SAVTAP=UNIT
REWIND SAVTAP
NF = FILE
CALL FSHSFL(SAVTAP,NF,ISTAT1)
IF(ISTAT1.EQ.0)GO TO 1
WRITE(6,220)FILE
270 FORMAT(/15,'ERROR IN POSITIONING SIG. EXTENSION TAPE TO FILE',I3/
  * 15,'OUTPUT FILE NOT WRITTEN')
1 CONTINUE
IR=1
DATA PCHUNT/7/
IF(PUNCH.NE.1)GO TO 6
WRITE(PCHUNT,110)
WRITE(PCHUNT,120)NOCLS,TOTSUR,NOFEAT,TOTFLD,TOTVRT
WRITE(PCHUNT,130)(FETVEC(I),I=1,NOFEAT)
6 CONTINUE
IF(ISTAT1.NE.0) GO TO 11
WRITE(SAVTAP)NOCLS,TOTSUR,NOFEAT,TOTFLD,TOTVRT,
  (FETVEC(J),J=1,NOFEAT)
11 CONTINUE
DO 2 I=1,TOTFLD
  NV = FLDSAV(4,I)
```

FILE: LARMAN

```
IF=IR *NV*2 -1
IF (PUNCH.NE.1) GO TO 7
WRITE (PCHUNT,135) (FLDSAV(J,I),J=1,4)
WRITE (PCHUNT,140) (VERTEX(J),J=18,1E)
7 CONTINUE
IF (ISTAT1.NE.0) GO TO 3
WRITE (SAVTAP) (FLDSAV(J,I),J=1,4)
WRITE (SAVTAP) (VERTEX(J),J=18,1E)
3 IR=IF+1
7 CONTINUE
IF (PUNCH.NE.1) GO TO A
WRITE (PCHUNT,145) (CLSNMS(I),I=1,NOCLS)
WRITE (PCHUNT,150) (NOSUBS(I),I=1,NOCLS)
WRITE (PCHUNT,155) (SUBNM(I),I=1,TOTSUB)
A CONTINUE
IF (ISTAT1.NE.0) GO TO 18
WRITE (SAVTAP) (CLSNMS(I),I=1,NOCLS), (NOSUBS(I),I=1,NOCLS),
* (SUBNM(I),I=1,TOTSUB)
1A CONTINUE
III = 0
WRITE (6,225)
225 FORMAT(1H1)
NUMSUR = 0
TOTSTA = 0
DO 20 ICLAS=1,NOCLS
TOTSTA = NUMSUR + TOTSTA
NUMSUR=NOSURS(ICLAS)
DO 20 J=1,NUMSUB
III=III+1
C STATS ARE COMING FROM ISOCLS
IF (SWTCH.NE.1) GO TO 50
MEAN1 = STADRS + (VARSIZ*NOFEAT)*TOTSTA
MEAN2 = MEAN1 + (J-1)*NOFEAT
COVAR1 = MEAN1 + NUMSUB*NOFEAT + VARSIZ*(J-1)
KK=III
GO TO A0
C STAT ARE COMING LABEL
50 IF (SWTCH.NE.2) GO TO 60
KK = SUBVEC(III)
MEAN1 = STADRS + VARSIZ*TOTSUB
MEAN2 = MEAN1 + NOFEAT*(KK-1)
COVAR1 = STADRS + VARSIZ*(KK-1)
GO TO A0
C STATS ARE COMING FROM STAT
60 CONTINUE
GO TO A0
C STATS ARE COMING TRSTAT
A0 CONTINUE
C
C READ MEANS AND COVARIANCES INTO CORE FROM DRUM
C
CALL RREAD(COVAR1,COVAR2,VARSIZ,ISTAT)
30 IF (ISTAT.EQ.1) GO TO 30
CALL RREAD(MEAN2,MEANS,NOFEAT,ISTAT1)
40 IF (ISTAT1.EQ.1) GO TO 40
C
IF (PUNCH.NE.1) GO TO 9
WRITE (PCHUNT,170) N(KK)
WRITE (PCHUNT,90) (MEANS(K),K=1,NOFEAT)
WRITE (PCHUNT,100) (COVAR(K),K=1,VARSIZ)
0 CONTINUE
IF (ISTAT1.NE.0) GO TO 20
WRITE (SAVTAP) N(KK), (COVAR(K),K=1,VARSIZ), (MEANS(K),K=1,NOFEAT)
IF (PRNSTS.NE.1) GOTO 20
C
C PRINTS THE STATS ON THE LINE PRINTER
C
DATA RCDTWO/'2'/
WRITE (6,65)
65 FORMAT(/)
WRITE (6,310) CLSNMS(ICLAS),SURNM(III)
310 FORMAT(/' CLASS :',A6/' SURCLASS: ',A6)
DO 340 LOC=1,NOFEAT,12
STOP=LOC+11
IF (STOP.GT.NOFEAT) STOP=NOFEAT
340 WRITE (6,350) (MEANS(I),I=LOC,STOP)
350 FORMAT(10 MEAN:'.3X,12F9.2)
WRITE (6,360)
360 FORMAT(/' COVARIANCE MATRIX:')
```

LAH00800  
LAH00810  
LAH00820  
LAH00830  
LAH00840  
LAH00850  
LAH00860  
LAH00870  
LAH00880  
LAH00890  
LAH00900  
LAH00910  
LAH00920  
LAH00930  
LAH00940  
LAH00950  
LAH00960  
LAH00970  
LAH00980  
LAH00990  
LAH01000  
LAH01010  
LAH01020  
LAH01030  
LAH01040  
LAH01050  
LAH01060  
LAH01070  
LAH01080  
LAH01090  
LAH01100  
LAH01110  
LAH01120  
LAH01130  
LAH01140  
LAH01150  
LAH01160  
LAH01170  
LAH01180  
LAH01190  
LAH01200  
LAH01210  
LAH01220  
LAH01230  
LAH01240  
LAH01250  
LAH01260  
LAH01270  
LAH01280  
LAH01290  
LAH01300  
LAH01310  
LAH01320  
LAH01330  
LAH01340  
LAH01350  
LAH01360  
LAH01370  
LAH01380  
LAH01390  
LAH01400  
LAH01410  
LAH01420  
LAH01430  
LAH01440  
LAH01450  
LAH01460  
LAH01470  
LAH01480  
LAH01490  
LAH01500  
LAH01510  
LAH01520  
LAH01530  
LAH01540  
LAH01550  
LAH01560  
LAH01570  
LAH01580



FILE: LABMAN

```
CALL WPTMTX(COVAR(1),NOFEAT,BCDTWO)
20 CONTINUE
IF (ISTAT1.NE. 0) GO TO 19
ENDFILE SAVTAP
19 CONTINUE
WRITE(6,180)NOCLS,TOTSUB
WRITE(6,190)
K=0
J=1
21 NSURS=NOSUHS(I)
WRITE(6,205)
WRITE(6,200)I,CLSNMS(I),NSUBS
WRITE(6,205)
DO 25 J=1,NSUBS
K=K+1
WRITE(6,210)K,SUBNM(K)
25 CONTINUE
I=I+1
IF(I.LE.NOCLS)GO TO 21
RETURN
90 FORMAT('MEAN ',5F15.8)
100 FORMAT('COVAR',5F15.8)
110 FORMAT('MODULE DECK FROM SLABEL')
120 FORMAT('NOCLS ',I4,' NOSUB ',I4,' NOFEAT ',I3,' NOFLD ',I4,' TOTVRT ',
* I5)
130 FORMAT('VECTR',5X,30I2)
135 FORMAT(A6,4X,3(I2,8X))
140 FORMAT('VERTICFS ',14I5)
145 FORMAT(('CLSDDESC ',9(2X,A6)))
150 FORMAT(('NOSUBS ',24(1X,I2)))
155 FORMAT(('SUBDESC ',10(A6,1X)))
160 FORMAT('FREQ ',12F6.2)
170 FORMAT('NOPTS',7X,18)
180 FORMAT('/// THE STATISTICS FILE FOR ',I4,' CLASSES AND ',I4,
* ' SUBCLASSES HAS BEEN WRITTEN'//)
185 FORMAT('IM ', 'THE STATS WERE WRITTEN ON FILE ',I3)
190 FORMAT(' THE STATS FOR A PARTICULAR CLASS OR SUBCLASS SHOULD BE RE
* FERRED TO IN LATER RUNS BY '/' THE FOLLOWING NAMES AND NUMBERS (W
* HICHEVER APPLICABLE)')
200 FORMAT(5X,'CLASS',I3,2X,A6,5X,'SUBCLASSES (TOTAL=',I3,')')
205 FORMAT('/')
210 FORMAT(25X,I3,2X,A6)
END
```

LAB01590  
LAB01600  
LAB01610  
LAB01620  
LAB01630  
LAB01640  
LAB01650  
LAB01660  
LAB01670  
LAB01680  
LAB01690  
LAB01700  
LAB01710  
LAB01720  
LAB01730  
LAB01740  
LAB01750  
LAB01760  
LAB01770  
LAB01780  
LAB01790  
LAB01800  
LAB01810  
LAB01820  
LAB01830  
LAB01840  
LAB01850  
LAB01860  
LAB01870  
LAB01880  
LAB01890  
LAB01900  
LAB01910  
LAB01920  
LAB01930  
LAB01940  
LAB01950  
LAB01960  
LAB01970  
LAB01980  
LAB01990  
LAB02000  
LAB02010

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: LAREAD

```
FUNCTION LAREAD(FLDNAM,VERTCS,FLDINF,NC)
  IMPLICIT INTEGER(A-Z)
  DIMENSION CARD(62), FLDINF(6),VERTCS(2,11),VER(2,10)
  DIMENSION ACARD(20)
  DATA BLANK/' ',COMMA',',OP/('(',CP/'),',
  *AST/'*',ENORCD/'$END',
  DATA C/'CLAS',S/'SUBC',
  DATA D/'DEST',
  DATA T/'TYPE',
  DO 50 I=1,2
  DO 50 J=1,10
C 50 VEP(I,J)=0
C SET UP REREAD BUFFER
  PRUNIT=30
  CALL RFREAD(RRUNIT,80)
C NOW PUT THE CARD INTO THE BUFFER
  1 READ(21,100)(ACARD(I),I=1,20)
  100 FORMAT(20A4)
  WRITE(RRUNIT,100)(ACARD(I),I=1,20)
  REWIND RRUNIT
C READ IN FIRST CARD
  READ(RRUNIT,20) FLDNAM
  20 FORMAT(A4)
  REWIND RRUNIT
  IF (FLDNAM.NE.T) GO TO 55
  LAPEAD = -4
  RETURN
  55 CONTINUE
  IF(FLDNAM.NE.D)GO TO 19
  LAPEAD=-3
  RETURN
  19 CONTINUE
  IF(FLDNAM .NE.C) GO TO 24
  LAPEAD=-1
  RETURN
  24 IF(FLDNAM .NE.S) GO TO 21
  LAPEAD=-2
  RETURN
  21 IF(FLDNAM .NE.ENDBCD) GO TO 22
  LAPEAD=0
  RETURN
C REREAD FIRST CARD
  22 READ(RRUNIT,23) CARD
  23 FORMAT(10X,62A1)
  REWIND RRUNIT
  COL=0
  II=0
  NC=0
C FIND (
  11 J=NXTCHR(CARD,COL)
  IF(J.EQ.BLANK) GO TO 1
  IF(J.EQ.AST) GO TO 10
  IF(J.NE.OP) GO TO 3
  9 I=0
  K=0
  KK=0
  NUM=0
  4 COL=COL+1
  IF(COL.GT.62) GO TO 35
  IF(CARD(COL).EQ.BLANK) GO TO 4
  IF((I.EQ.0).AND.(CARD(COL).EQ.COMMA)) GO TO 3
  IF((I.EQ.1).AND.(CARD(COL).EQ.COMMA)) GO TO 7
  IF((K.EQ.0).AND.(CARD(COL).EQ.CP)) GO TO 3
  IF((K.EQ.1).AND.(CARD(COL).EQ.CP)) GO TO 8
  CALL I4ABN(CARD(COL),I,NW)
  NUM=I0*NUM+NW
  IF((I4.LT.0).OR.(I4.GT.9)) GO TO 3
  I=I+1
  IF(K.EQ.1) KK=1
  GO TO 4
  7 IF(II.EQ.0) GO TO 30
C VERTEX SAMPLE NUMBER
  NC=NC+1
  VER(I,NC)=NUM
  K=1
  NIJM=0
  GO TO 4
  8 IF(II.EQ.0) GO TO 31
C VERTEX LINE NUMBER
```

LAR00010  
LAR00020  
LAR00030  
LAR00040  
LAR00050  
LAR00060  
LAR00070  
LAR00080  
LAR00090  
LAR00100  
LAR00110  
LAR00120  
LAR00130  
LAR00140  
LAR00150  
LAR00160  
LAR00170  
LAR00180  
LAR00190  
LAR00200  
LAR00210  
LAR00220  
LAR00230  
LAR00240  
LAR00250  
LAR00260  
LAR00270  
LAR00280  
LAR00290  
LAR00300  
LAR00310  
LAR00320  
LAR00330  
LAR00340  
LAR00350  
LAR00360  
LAR00370  
LAR00380  
LAR00390  
LAR00400  
LAR00410  
LAR00420  
LAR00430  
LAR00440  
LAR00450  
LAR00460  
LAR00470  
LAR00480  
LAR00490  
LAR00500  
LAR00510  
LAR00520  
LAR00530  
LAR00540  
LAR00550  
LAR00560  
LAR00570  
LAR00580  
LAR00590  
LAR00600  
LAR00610  
LAR00620  
LAR00630  
LAR00640  
LAR00650  
LAR00660  
LAR00670  
LAR00680  
LAR00690  
LAR00700  
LAR00710  
LAR00720  
LAR00730  
LAR00740  
LAR00750  
LAR00760  
LAR00770  
LAR00780  
LAR00790

FILE: LAREAD

```
VER(2,NC)=NUM
C CHECK FOR COMMA OR ASTERISK
6 J=NXTCHR(CARD,COL)
IF(J.EQ.BLANK) GO TO 2
IF(J.EQ.AST) GO TO 10
IF(J.NE.COMMA) GO TO 3
GO TO 11
3 WRITE(6,13) CARD
13 FORMAT(' ERROR IN FIELD CARD TERMINATING RUN'/10X,62A1)
CALL CMERR
5 WRITE(6,15) CARD
15 FORMAT(10X,62A1/' INCORRECT FIELD CARD,TERMINATING RUN')
CALL CMERR
C DETERMINE RECTANGULAR FIELD COORDINATES
2 IF((NC.LT.1).OR.(NC.GT.10)) GO TO 3
NT1=0
NT3=0
NT2=1000000
NT4=1000000
DO 14 N=1,NC
IF((VER(1,N).EQ.0).OR.(VER(2,N).EQ.0)) GO TO 5
IF(VER(1,N).GT.NT1) NT1=VER(1,N)
IF(VER(2,N).LT.NT4) NT4=VER(2,N)
IF(VER(2,N).GT.NT3) NT3=VER(2,N)
IF(VER(1,N).GE.NT2) GO TO 14
NT2=VER(1,N)
CNT=N
14 CONTINUE
FLDINF(1)=NT4
FLDINF(2)=NT3
FLDINF(4)=NT2
FLDINF(5)=NT1
C SFT UP VERTICES IN CLOCKWISE ORDER WITH SMALLEST SAMPLE FIRST
DO 32 I=1,NC
IF(CNT.GT.NC) CNT=1
VERTCS(1,I)=VER(1,CNT)
VERTCS(2,I)=VER(2,CNT)
32 CNT=CNT+1
VERTCS(1,NC+1)=VERTCS(1,1)
VERTCS(2,NC+1)=VERTCS(2,1)
35 IAREAD=1
NC=NC+1
RETURN
30 FLDINF(6)=NUM
K=1
NUM=0
GO TO 4
31 FLDINF(3)=NUM
II=1
GO TO 6
C READ CONTINUATION CARD
10 READ(21,23) CARD
COL=0
GO TO 11
END
```

LAR00800  
LAR00810  
LAR00820  
LAR00830  
LAR00840  
LAR00850  
LAR00860  
LAR00870  
LAR00880  
LAR00890  
LAR00900  
LAR00910  
LAR00920  
LAR00930  
LAR00940  
LAR00950  
LAR00960  
LAR00970  
LAR00980  
LAR00990  
LAR01000  
LAR01010  
LAR01020  
LAR01030  
LAR01040  
LAR01050  
LAR01060  
LAR01070  
LAR01080  
LAR01090  
LAR01100  
LAR01110  
LAR01120  
LAR01130  
LAR01140  
LAR01150  
LAR01160  
LAR01170  
LAR01180  
LAR01190  
LAR01200  
LAR01210  
LAR01220  
LAR01230  
LAR01240  
LAR01250  
LAR01260  
LAR01270  
LAR01280  
LAR01290  
LAR01300  
LAR01310  
LAR01320  
LAR01330  
LAR01340

FILE LINERD

```

SUBROUTINE LINERD(/IDATA/,ENDTAP)
IMPLICIT INTEGER (A-Z)
C*
C*
C* ENTRY FOR READING AND UNPACKING ONE SCAN LINE OF DATA
C*
LOGICAL*1 ISCAN(4),BYTE(4),IDATA(1),IBUF(13500),IZERO(4)
C INCLUDE CMHK17
C*START
COMMON /HUFF/IBUFF(3375)
COMMON /TAPERD/ IUNIT,(FRST,FSCAN,SAMEND,SAMINC,READY,NSCAN,
* LINC,ID(200),DSL,LBUF(30),JREC(30),IHYTE(30),NHUFS,FILENO,LINEND
* LININC,NSAMP,NOCHAN,FORMT
C*END
EQUIVALENCE (SCAN,ISCAN(1)),(IZERO(1),ZERO)
EQUIVALENCE (IHUFF,IBUF(1))
EQUIVALENCE (ID(1),NRPDS),(ID(2),NCPR),
* (ID(3),NPRC),(ID(4),ANCLNG),
* (ID(5),NC),(ID(6),NS),
* (ID(7),NHITS),(ID(8),DOI),
* (ID(9),NDSPR),(ID(10),NCAR),
* (ID(11),SVD),(ID(16),PRSZ)
C ZERO OUT IDATA
C
SCAN = 0
TOTPIX=NSAMP*NOCHAN*4
ZERO=0
DO 180 J=1,TOTPIX
180 IDATA(J)=IZERO(4)
IF (READY)190,190,200
190 WRITE (6,410)
C*
200 IADR = 4
MAXREC = PRSZ
RUF = 1
REC = 0
IF (FORMT.EQ.4)GO TO 2000
IF (NDSPR.EQ.1)GO TO 195
IF (FSCAN.EQ.NSCAN)GO TO 195
IF ((FSCAN+NDSPR-1).LE.NSCAN)GO TO 196
195 CALL HUFFILL(REC,IUNIT,MAXREC,IBUF,NRPDS,ENDTAP,IERR)
IF (FORMT.EQ.3)GO TO 1000
IF (FORMT.EQ.1) ISCAN(3) = IHUF(71)
IF (FORMT.EQ.1) ISCAN(4) = IHUF(72)
IF (FORMT.EQ.2) ISCAN(3) = IHUF(1)
IF (FORMT.EQ.2) ISCAN(4) = IHUF(2)
C
IF (SCAN.EQ.FSCAN) GO TO 196
C
CALL SEARCH(&250,&235,ENDTAP,IBUF,NRPDS,NDSPR)
MAXREC=PRSZ
RUF = 1
REC = 0
CALL HUFFILL(REC,IUNIT,MAXREC,IBUF,NRPDS,ENDTAP,IERR)
196 CONTINUE
ADD = (NSCAN-FSCAN)*DSL
DO 230 IFT=1,NOCHAN
201 IF (LBUF(IFT).EQ.HUF)GO TO 205
CALL HUFFILL(REC,IUNIT,MAXREC,IBUF,NRPDS,ENDTAP,IERR)
RUF=RUF+1
GO TO 201
205 CONTINUE
J=JREC(IFT)
JJ=(J-1)*MAXREC
C*
C* CHECK STATUS OF THIS RECORD BEFORE UNPACKING
C*
IF (ENDTAP.EQ.-1) GO TO 250
C*
C* UNPACK DATA FOR THIS FEATURE
C*
220 IP = ADD + IHYTE(IFT)+JJ
DO 225 II=1,NSAMP
IDATA(IADR+4*(II-1)) = IBUF(IP+SAMINC*(II-1))
225 CONTINUE
IADR = IADR + NSAMP*4

```

LIN00010  
LIN00020  
LIN00030  
LIN00040  
LIN00050  
LIN00060  
LIN00070  
LIN00080  
LIN00090  
LIN00100  
LIN00110  
LIN00120  
LIN00130  
LIN00140  
LIN00150  
LIN00160  
LIN00170  
LIN00180  
LIN00190  
LIN00200  
LIN00210  
LIN00220  
LIN00230  
LIN00240  
LIN00250  
LIN00260  
LIN00270  
LIN00280  
LIN00290  
LIN00300  
LIN00310  
LIN00320  
LIN00330  
LIN00340  
LIN00350  
LIN00360  
LIN00370  
LIN00380  
LIN00390  
LIN00400  
LIN00410  
LIN00420  
LIN00430  
LIN00440  
LIN00450  
LIN00460  
LIN00470  
LIN00480  
LIN00490  
LIN00500  
LIN00510  
LIN00520  
LIN00530  
LIN00540  
LIN00550  
LIN00560  
LIN00570  
LIN00580  
LIN00590  
LIN00600  
LIN00610  
LIN00620  
LIN00630  
LIN00640  
LIN00650  
LIN00660  
LIN00670  
LIN00680  
LIN00690  
LIN00700  
LIN00710  
LIN00720  
LIN00730  
LIN00740  
LIN00750  
LIN00760

FILE LINERD.

C*	230	CONTINUE	LIN00770
C*		FINISHED UNPACKING ONE SCAN LINE OF DATA	LIN00780
C*		IF ((NSCAN+LININC).GT.LINEND) GO TO 260	LIN00790
C*		MAKE SURE ALL BUFFERS FOR THIS DATA SET HAVE BEEN READ	LIN00800
C*			LIN00810
	231	IF (RUF.EQ.NRUF) GO TO 235	LIN00820
		CALL HUFILL(REC,IUNIT,MAXREC,IBUF,NRPOS,ENDTAP,IERR)	LIN00830
		IF (ENDTAP.EQ.-1) GO TO 250	LIN00840
		RUF=RUF+1	LIN00850
		GO TO 231	LIN00860
	235	CONTINUE	LIN00870
		NSCAN=NSCAN+LININC	LIN00880
		IF (NSCAN.LT.(FSCAN+NDSPR)) RETURN	LIN00890
		IF (NSCAN.GT.(FSCAN+NDSPR)) GO TO 236	LIN00900
		FSCAN = FSCAN + NDSPR*(1 + LINC/NRPOS)	LIN00910
		RETURN	LIN00920
	236	FSCAN=FSCAN + NDSPR*(1 + LINC/NRPOS)	LIN00930
		DO 237 II=1,LINC	LIN00940
	237	READ(IUNIT,420)DUMMY	LIN00950
		IF (NSCAN.LT.(FSCAN+NDSPR)) GO TO 240	LIN00960
		READ(IUNIT,420)DUMMY	LIN00970
		FSCAN=FSCAN+NDSPR	LIN00980
	240	CONTINUE	LIN00990
		RETURN	LIN01000
	250	IF (NSCAN.GT.LINSTR) NSCAN=NSCAN-LININC	LIN01010
		WRITE (6,320) NSCAN	LIN01020
		IF (FILENO.EQ.0) GO TO 255	LIN01030
C*			LIN01040
C*		RACK SPACE 1 FILE AND POSITION AT FIRST SCAN LINE	LIN01050
C*			LIN01060
		RSKIP = (NSCAN-IFRST) * NDSPR + 1	LIN01070
		WRITE(6,560)RSKIP	LIN01080
	560	FORMAT(' RACKSPACE',I3)	LIN01090
		DO 253 II=1,RSKIP	LIN01100
	253	RACKSPACE IUNIT	LIN01110
		GO TO 257	LIN01120
C*			LIN01130
C*		REWIND TAPE AND POSITION AT FIRST SCAN LINE	LIN01140
C*			LIN01150
	255	REWIND IUNIT	LIN01160
		READ(IUNIT,420)DUMMY	LIN01170
	257	FSCAN = IFRST	LIN01180
	260	READY = -1	LIN01190
		RETURN	LIN01200
C			LIN01210
C		UNPACK SCAN LINE OF DATA FOR LANDSAT 1 OR 2	LIN01220
C			LIN01230
1000		SAMSTR=JREC(1)	LIN01240
		DO 1100 I=1,NOCHAN	LIN01250
		IJ=-1	LIN01260
		DO 1200 II=SAMSTR,SAMEND,SAMINC	LIN01270
		IJ=IJ+1	LIN01280
		IJ=II	LIN01290
		KK=1	LIN01300
		IF (MOD (II,2).EQ.0) JJ=JJ-1	LIN01310
		IF (MOD (II,2).EQ.0) KK=1	LIN01320
		IADD=IBYTE(I)+(JJ-1)*4+KK	LIN01330
		DATA(IADR+4*IJ)=IBUF(IADD)	LIN01340
1200		CONTINUE	LIN01350
		IADR=IADR+NSAMP*4	LIN01360
1100		CONTINUE	LIN01370
		GO TO 235	LIN01380
C			LIN01390
C		UNPACK SCAN LINE FOR LANDSAT III	LIN01400
C			LIN01410
2000		J=1	LIN01420
		SAMSTR=IRUF(1)	LIN01430
		DO 2500 I=1,NOCHAN	LIN01440
		IF (IBYTE(I).EQ.J) GO TO 2200	LIN01450
		LIM=IBYTE(I)-J	LIN01460
		DO 2100 II=1,LIM	LIN01470
		READ(IUNIT,420)DUMMY	LIN01480
2100		CONTINUE	LIN01490
2200		CALL HUFILL(REC,IUNIT,MAXREC,IBUF,1,ENDTAP,IERR)	LIN01500
		DO 2300 II=1,NSAMP	LIN01510
			LIN01520

FILE LINERO

2300	DATA(IADR+4*(II-1))=IHUF(13*SAMINC*(II-1)*SAMSTR)	L	N	530
	CONTINUE	L	N	540
	IADR=IADR+4*NSAMP	L	N	550
	J=IAYE(I)+1	L	N	560
2400	CONTINUE	L	N	570
	IF(J.GT.NRPOS)GO TO 235	L	N	580
	LIM=NRPOS-J+1	L	N	590
	DO 2600 II=1,LIM	L	N	600
	READ(IUNIT,420)DUMMY	L	N	610
2500	CONTINUE	L	N	620
	GO TO 235	L	N	630
320	FORMAT(' FIELD BOUNDARY FOR THIS FIELD DEFINED BEYOND SCOPE OF DATA')	L	N	640
	*TA// THIS FLIGHT LINE CONTAINS '.16.' SCAN LINES')	L	N	650
410	FORMAT(' FLDINT MUST BE CALLED TO INITIALIZE PARAMETERS FOR A NEW FIELD')	L	N	660
420	FORMAT(1A4)	L	N	670
	END	L	N	680
		L	N	690

FILE: LISTLC

```

C          FIELDS - CATEGORY NAME AND DOT TYPE FOR DOT 1 STORED IN          L|S00010
C          FIELD(1,1) AND FIELD(4,1)                                       L|S00020
C          STAMNT - INITIALLY SET TO 1, SWITCHED TO INDICATE DOTS BEING     L|S00030
C          TAKEN FROM CURRENTLY READ CARD.                                  L|S00040
C          IPT - INITIALLY SET TO 1, INDEX NUMBER FOR FIELD VERTEX INFORMATION L|S00050
C          VERTEX - VERTEX INFORMATION FOR EACH DOT.                         L|S00060
C          SUBROUTINE LISTLC(FIELDS,STAMNT,*,*,*,SWCHG,INIT,IUNIT,IFILE,IPT,  L|S00070
C          *VERTEX)                                                         L|S00080
C          IMPLICIT INTEGER (A-Z)                                          L|S00090
C          LOGICAL*1 LCARD(300),LCATNM(4)                                  L|S00100
C          REAL DUM                                                         L|S00110
C          DIMENSION FIELDS(4,1),VERTEX(1),CARD(75),NDOTS(30)             L|S00120
C          DIMENSION ACARD(80)                                             L|S00130
C          LOGICAL SWITCH                                                  L|S00140
C          DATA SWITCH/,TRUE./,ENDRCD/'SEN'/,                             L|S00150
C          *CATNM)/' ' /                                                  L|S00160
C          INCLUDE CMHK14                                                  L|S00170
C          INCLUDE COMAK1                                                  L|S00180
C          COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,        L|S00190
C          *AVAR2,COVAR2,CLS1D2,SUHND2,SUHDS2,FLDSV2,VERTX2,              L|S00200
C          *FFTVC2(30),SU=VC2(75),SUMPTR(75),CLSV2(60),                   L|S00210
C          *KFPPTS(60),NOGRP,GHPNAM(60),GRPDIX(61),                        L|S00220
C          *GRPCHK(61),GROUPS(124)                                         L|S00230
C          COMMON /DOTVEC/TYPE,CATNM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY     L|S00240
C          *SIZE,LACIE                                                     L|S00250
C          DIMENSION IBUF(80)                                              L|S00260
CSEND  EQUIVALENCE (LCATNM(1),CATNM),(CARD(1),LCARD(1))                   L|S00270
C          IF (INIT.NE.0) GO TO 5                                           L|S00280
C          REWIND IUNIT                                                    L|S00290
C          CALL FCFMFL(IUNIT,IFILE,ISTAT)                                   L|S00300
C          READ (IUNIT,1010) (IBUF(I), I=1,80)                             L|S00310
C          FORMAT(M0A1)                                                    L|S00320
C          WRITE(6,1020) (IBUF(I), I=1,80)                                  L|S00330
C          FORMAT(1H0,80A1)                                                L|S00340
C          INIT = 1                                                         L|S00350
C          IF (STAMNT.EQ.2) GO TO 30                                         L|S00360
C          IF (.NOT.SWITCH) GO TO 20                                        L|S00370
C          CALL REPEAT(30,80)                                              L|S00380
C          READ (IUNIT,103) (ACARD(I), I=1,80)                              L|S00390
C          FORMAT(M0A1)                                                    L|S00400
C          WRITE (30,103) (ACARD(I), I=1,80)                                L|S00410
C          REWIND 30                                                         L|S00420
C          READ (30,1000) ID,TYPES,CARD                                     L|S00430
C          REWIND 30                                                         L|S00440
C          FORMAT(A3,1X,I1,75A1)                                           L|S00450
C          IF (TYPE.EQ.TYPES) GO TO 20                                       L|S00460
C          IF (SWCHG.NE.0) GO TO 40                                       L|S00470
C          TYPE = TYPES                                                    L|S00480
C          READ CARD                                                         L|S00490
C          COL = 0                                                           L|S00500
C          CATNM = NXCHP(CARD,COL)                                          L|S00510
C          IF NEXT CHAR IS NOT A CAT. NAME, CORRECT COL COUNT TO READ NUM  L|S00520
C          IF (CATNM.GT.0) GO TO 21                                         L|S00530
C          LINDEX=4*COL+1                                                  L|S00540
C          LCATNM(2)=LCARD(LINDEX)                                          L|S00550
C          COL=COL+1                                                        L|S00560
C          IF (CATNM.EQ.CATNM) GO TO 23                                       L|S00570
C          NOCAT=NOCAT + 1                                                 L|S00580
C          CATNAM(NOCAT)=CATNM                                             L|S00590
C          CATNM = CATNM                                                    L|S00600
C          GO TO 23                                                         L|S00610
C          COL=COL - 1                                                       L|S00620
C          NDCARD=0                                                         L|S00630
C          CALL NUMBP(NDOTS,NDCARD,CARD,COL)                                L|S00640
C          IF (NDCARD.EQ.0) GO TO 10                                         L|S00650
C          ICNT = 0                                                         L|S00660
C          STAMNT = 2                                                       L|S00670
C          SWITCH = .TRUE.                                                 L|S00680
C          GO TO 100                                                         L|S00690
C          TEST FOR END OF DOTS TO BE PROCESSED ON CARD                    L|S00700
C          IF (ICNT.LT.NDCARD) GO TO 100                                     L|S00710
C          READ NEXT CARD                                                  L|S00720
C          L|S00730
C          L|S00740
C          L|S00750
C          L|S00760
C          L|S00770
C          L|S00780
C          L|S00790

```

FILE: LISTLC

C

```
STAMNT = 1
ICNT = 0
READ(IUNIT,103)(ACARD(I),I=1,40)
WRITE(30,103)(ACARD(I),I=1,80)
REWIND 30
RFAD(30,1000)ID,TYPES,CARD
REWIND 30
IF(ID.FO.FNDHCD)RETURN 3
IF(TYPE.EQ.TYPES)GO TO 20
SWITCH = .FALSE.
SWCHG = SWCHG + 1
IF(SWCHG.GT.1)GO TO 40
TYPE = TYPES
IPT = 0
C***** CHANGED JUNE 28 1978
RETURN 2
```

C

100

```
ICNT = ICNT + 1
NOFLD2 = NOFLD2 + 1
```

C

```
COMPUTE LINE INCREMENT
```

C

```
NN = NDOTS(ICNT)
NI = IARS(NN) / 100000000
LI = IARS(NN) - NI * 100000000
IF(LI.GE.100000000)NI = NI + 1
```

C

```
COMPUTE SAMPLE INCREMENT.
```

C

```
KK=1
IF(NN.LT.0)KK=-1
LI = NI * KK
N2 = NN - LI * 100000000
N3 = IARS(N2)/10000
SI = IARS(N2) - N3 * 10000
IF(SI.GE.10000)N3 = N3 + 1
KK=1
IF(N2.LT.0)KK=-1
SI = N3 * KK
LACI = N2 - SI * 10000
LR = (LACI-1)/19
LS = LR - 1
LS = LS / 10
LS = 10 * (LACI - (LS*19))
L = LR - LI
S = LS + SI
```

C

```
STORE DOT INFO
```

```
FIELDS(1,NOFLD2) = CATNM
FIELDS(4,NOFLD2) = 2
FLDINF(1) = L
FLDINF(2) = L
FLDINF(3) = I
FLDINF(4) = S
FLDINF(5) = S
FLDINF(6) = I
```

```
IF(IPT.NE.0)GO TO 35
```

35

```
IPT = IPT + 4
VERTEX(IPT) = S
VERTEX(IPT+1) = L
VERTEX(IPT+2) = S
VERTEX(IPT+3) = L
RETURN 1
```

40

```
WRITE(5,2000)
2000 FORMAT(//EX. ERROR HAS OCCURRED IN READING LACIE FORMATTED DOT CAR
DS - SUBROUTINE FLULAC - EXIT TAKEN)
RETURN 3
END
```

```
L|S00800
L|S00810
L|S00820
L|S00830
L|S00840
L|S00850
L|S00860
L|S00870
L|S00880
L|S00890
L|S00900
L|S00910
L|S00920
L|S00930
L|S00940
L|S00950
L|S00960
L|S00970
L|S00980
L|S00990
L|S01000
L|S01010
L|S01020
L|S01030
L|S01040
L|S01050
L|S01060
L|S01070
L|S01080
L|S01090
L|S01100
L|S01110
L|S01120
L|S01130
L|S01140
L|S01150
L|S01160
L|S01170
L|S01180
L|S01190
L|S01200
L|S01210
L|S01220
L|S01230
L|S01240
L|S01250
L|S01260
L|S01270
L|S01280
L|S01290
L|S01300
L|S01310
L|S01320
L|S01330
L|S01340
L|S01350
L|S01360
L|S01370
L|S01380
L|S01390
L|S01400
L|S01410
L|S01420
L|S01430
L|S01440
L|S01450
L|S01460
L|S01470
L|S01480
L|S01490
L|S01500
L|S01510
L|S01520
```

19-52  
425



FILE: MATVEC

```
SUBROUTINE MATVEC(A,H,C,L,M)
C MULTIPLY MATRIX A BY VECTOR B AND STORE IN VECTOR C
DIMENSION A(L,H),B(H),C(L)
DO 10 I=1,L
SUM=0.
DO 5 K=1,M
5 SUM=SUM+A(I,K)*B(K)
10 C(I)=SUM
RETURN
END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: MTMDAT

```
C
C
C
SUBROUTINE MTMDAT(A,B,C,L,M,N,D,DD)
MULTIPLY MATRIX A BY THE TRANSPOSE OF B AND STORE IN DD
A LOWER TRIANGULAR MATRIX
DIMENSION A(L,M),B(N,M),C(L,N),D(L),DD(I)
DO 60 I=1,L
DO 60 J=1,N
SUM=0.0
DO 55 K=1,M
55 SUM=SUM+A(I,K)*B(J,K)
C(I,J)=SUM
IF(I.EQ.J) D(I)=SORT(SUM)
60 CONTINUE
MM=0
KK=0
DO 1 II=1,L
KK=KK+1
DO 1 LL=1,KK
MM=MM+1
DD(MM)=C(II,LL)
1 CONTINUE
RETURN
END
```

FILE: MTMLS6

	SUBROUTINE MTMLS6(A,B,C,M,N)	MTM00010
C	MULTIPLY MATRIX A BY B AND STORE IN C. B IS STORED IN SYMMETRIC NOTAT	MTM00020
C	DIMENSION A(M,N),H(1),C(M,N)	MTM00030
	DO 50 J=1,M	MTM00040
	LE=0	MTM00050
	DO 40 I=1,N	MTM00060
	LB=LE+1	MTM00070
	LE=LE+I	MTM00080
	SUM=0.	MTM00090
	K=0	MTM00100
	DO 35 L=LB,LE	MTM00110
	K=K+1	MTM00120
35	SUM=SUM+A(J,K)*B(L)	MTM00130
	IF(I.FQ.N) GO TO 40	MTM00140
	KS=K+1	MTM00150
	L=LE	MTM00160
	DO 36 K=KS,N	MTM00170
	L=L+I+K-KS	MTM00180
36	SUM=SUM+A(J,K)*B(L)	MTM00190
40	C(J,I)=SUM	MTM00200
50	CONTINUE	MTM00210
	RETURN	MTM00220
	END	MTM00230
		MTM00240
		MTM00250

FILE: NAMSTA

```
C      NAMSTA ASSIGNS NAMES TO CLUSTERS AND UPDATES STAT INFO      NAM00010
C      SUBROUTINE NAMSTA(SURNAM,CATVEC,SUBNO,NOSUB2,CATNAM,NOCAT)    NAM00020
C      IMPLICIT INTEGER (A-Z)                                       NAM00030
C      DIMENSION SURNAM(60),CATNAM(60)                               NAM00040
C      DIMENSION CATVEC(60),SUBNO(1)                                NAM00050
C      INTEGER*4 I4(3)                                              NAM00060
C      LOGICAL*1 L1(12)                                             NAM00070
C      EQUIVALENCE(L1(1),I4(1))                                     NAM00080
C      K = 0                                                         NAM00090
C      ASSIGN NAMES TO CLUSTERS                                     NAM00100
C      DO 20 I=1,NOCAT                                             NAM00110
C      L = 0                                                         NAM00120
C      DO 20 J=1,NOSUB2                                           NAM00130
C      IF (CATVEC(J) .NE. 1) GO TO 20                               NAM00140
C      K = K + 1                                                    NAM00150
C      L = L + 1                                                    NAM00160
C      USE FIRST 2 CHAR OF CATEGORY NAME + 2 DIGITS              NAM00170
C      I4(1) = CATNAM(I)                                           NAM00180
C      CALL RNI421(I4(2),2,L)                                       NAM00190
C      L1(3)=L1(5)                                                  NAM00200
C      L1(4)=L1(9)                                                  NAM00210
C      SURNAM(K) = I4(1)                                           NAM00220
C      20 CONTINUE                                                 NAM00230
C      CHECK FOR NULL CATEGORY                                     NAM00240
C      I=NOCAT                                                      NAM00250
C      30 IF (SURNAM(I) .NE. 0) GO TO 60                            NAM00260
C      IF (I .EQ. NOCAT) GO TO 55                                   NAM00270
C      DO 50 J=I,NOCAT                                             NAM00280
C      CATNAM(J) = CATNAM(J+1)                                       NAM00290
C      50 SURNAM(J) = SURNAM(J+1)                                   NAM00300
C      55 NOCAT = NOCAT - 1                                         NAM00310
C      60 I=I-1                                                     NAM00320
C      IF (I.GT.0) GOTO 30                                         NAM00330
C      RETURN                                                       NAM00340
C      END                                                           NAM00350
C      60000                                                         NAM00360
C      60001                                                         NAM00370
C      60002                                                         NAM00380
C      60003                                                         NAM00390
C      60004                                                         NAM00400
C      60005                                                         NAM00410
C      60006                                                         NAM00420
C      60007                                                         NAM00430
C      60008                                                         NAM00440
C      60009                                                         NAM00450
C      60010                                                         NAM00460
C      60011                                                         NAM00470
C      60012                                                         NAM00480
```

FILE: NUMBER

```
FUNCTION NUMBER(CARD,COL,NUMVEC,NOW)
IMPLICIT INTEGER (A-Z)
DIMENSION CARD(1),NUMVEC(1)
DATA CRDSIZ/62/,VECSIZ/100/,BLANK/' ',COMMA/',',/
DATA ZERO/'0'/,NINE/'9'/
NEXT = NOW + 1
IF (NEXT .LE. 0 .OR. NEXT .GT. VECSIZ) NEXT = 1
J = 0
L = COL + 1
IF (L .GT. CRDSIZ) GO TO 92
VK=VECSIZ
DO 80 J=NEXT,VK+1
JJ = J
NUM = 0
ITRIG=0
DO 60 COL=L,CRDSIZ,1
IF (CARD(COL).EQ.BLANK) GO TO 60
IF (CARD(COL).EQ.COMMA) GO TO 70
IF (CARD(COL).LT.ZERO.OR.CARD(COL).GT.NINE) GO TO 90
CALL I4A1HN(CARD(COL),1,NWORD)
NUM = 10 * NUM + NWORD
ITRIG=1
60 CONTINUE
COL = CRDSIZ
GO TO 90
70 NUMVEC(J) = NUM
L = COL + 1
IF (L .GT. CRDSIZ) GO TO 92
90 CONTINUE
J = VECSIZ
IF (ITRIG.EQ.1) GO TO 91
J = J - 1
GO TO 92
91 NUMVEC(J) = NUM
92 NUMBER = J
C 106 WRITE(6,106) (CARD(K),K=1,62),COL,NUMBER,(NUMVEC(K),K=1,J)
C 106 FORMAT(' NUMBER ENTERED'/' ',62A1,I10/' ',15,18I3)
RETURN
C*****
C***** FUNCTION ENTRIES MUST RETURN VALUE IN ORIGINAL FUNCTION NAME
C*****
END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

~~19-57~~  
430

FILE: NUMBR

```
C* SURROUTINE NUMBR WILL PROCESS ONE CARD AT A TIME. NUM00010
C* IT READS AND STORES ALL NUMBERS IN ARRAY NDOTS WITH NUM00020
C* NDCARD AS AN INDEX. BLANKS ARE THE ONLY RECONIZED NUM00030
C* DELIMITERS. NUM00040
SURROUTINE NUMBR (NDOTS, NDCARD, CARD, COL) NUM00050
IMPLICIT INTEGER (A-Z) NUM00060
DIMENSION NDOTS(1), CARD(1) NUM00070
DATA BLANK/' ', CRDSIZ/75/ NUM00080
NUM=0 NUM00090
NC = COL + 1 NUM00100
5 IF (NC.GT.CRDSIZ)GO TO 50 NUM00110
DO 10 I=NC,CRDSIZ NUM00120
IF (CARD(I).EQ.BLANK)GO TO 7 NUM00130
CALL I4A1BY(CARD(I),1,NWORD) NUM00140
NUM = NUM*10 + NWORD NUM00150
GO TO 30 NUM00160
7 IF (NUM.LT.1)GO TO 30 NUM00170
IF (NUM.GT.209)WRITE(6,500)NUM NUM00180
NDCARD=NDCARD + 1 NUM00190
NDOTS(NDCARD)=NUM NUM00200
NUM = 0 NUM00210
30 CONTINUE NUM00220
10 CONTINUE NUM00230
500 FORMAT(//5X,'LACIE DOT READ THAT IS GREATER THAN SIZE LIMIT NUM00240
:OF 209 - EXECUTION CONTINUED WITH VALUE READ OF ',I4) NUM00250
50 CONTINUE NUM00260
RETURN NUM00270
END NUM00280
```



FILE: ORDER

```
SUBROUTINE ORDER(VEC,N)
  IMPLICIT INTEGER(A-Z)
  DIMENSION VEC(1)
  LOGICAL SWITCH
  IF(N.LE.1)RETURN
  M=N-1
  5 SWITCH=.FALSE.
  DO 10 I=1,M
  IF(VEC(I).LE.VEC(I+1))GO TO 10
  TEMP=VEC(I)
  VEC(I)=VEC(I+1)
  VEC(I+1)=TEMP
  SWITCH=.TRUE.
  10 CONTINUE
  IF(SWITCH)GO TO 5
  RETURN
  END
```

~~19-60~~

433



FILE: PRINT

```
SUBROUTINE PRINT(KKT,IPLACE,MEANS,STDEV,CLD,FLDINF,N)
IMPLICIT INTEGER(A-X)
INCLUDE COMPK5.LIST
C INCLUDE COMPK6.LIST
COMMON/PASS/STOP,LNCAT,NMIN,KRN,STDMAX,DLMIN,SEP,
MAP,SPTRIG,IRD,KPTS,NOPTS,PUNCH,
* ICHN,CHNTHS,ICHAIN(42),NWDS,IREGIN,BEGIN1,
* HEGIN2,HEGIN3,CLSNAM,NOFLD,IPT,TOTWRD,TOTPTS,
* NCLASS,NOCLS,TOTSUR,TOTFLD,TOTVRT,NOCL,NVRT
* ,NXTCLS,NOFEAT,MAXCLS,FETVEC(30),SYMMTX(62)
* ,VARSIZ,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
* ,JORDER,INUNIT,INFILE,INITM,PMIN,SURVEC(62),NOSUR2,CHNVC(30)
* ,NOCHAN,FRCOMP,NOSED,MEANDU,MEANDU,
* SYMDO,SYMDOU,ITRIGO,ITRIGU,DOFLAG,
* DUFLAG,DODU,SDOTS(60),NSDOTS,SUNCOR(30),LLNCAT,
* DVERT(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2)
* ,MXFET1,MAXPOP
REAL SUNCOR
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWDS,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,PANDIO
CSEND
DIMENSION IPLACE(NOPTS)
DIMENSION MEANS(NOFEAT,MAXCLS),STDEV(NOFEAT,MAXCLS)
DIMENSION SYMBLS(1),FL(12)
DIMENSION FLDINF(1)
DIMENSION CLD(MAXCLS,1),N(MAXCLS),NBLK(62),FINF(6)
REAL MEANS,STDEV,CLD
DIMENSION COL(3,110),OUT(110)
EQUIVALENCE (FINF(1),LINSTR), (FINF(4),SAMSTR),
* (FINF(2),LINENR), (FINF(5),SAMENR),
* (FINF(3),LININC), (FINF(6),SAMINC)
EQUIVALENCE (SYMMTX,SYMBLS)
EQUIVALENCE (COL(1,1),OUT(1)), (COL(1,111),NBLK)
DATA BLANK/' '/
IF(DODU.EQ.0) GO TO 20
SAVEP=SYMMTX(LNCAT-DODU+DOFLAG)
SAVEF=SYMMTX(LNCAT-DODU+DODU)
IF (DOFLAG.NE.0) SYMMTX(LNCAT-DODU + 1) = SYMDO
IF (DOFLAG.NE.0) SYMMTX(LNCAT) = SYMDO
20 CONTINUE
WRITE (6,HEAD)
IF (KKT.GT.0) WRITE (6,240) KKT
IF (KKT.LT.0) WRITE (6,245) CLSNAM
WRITE (6,250) LNCAT
TOTPTS=TOTWRD/NOFEAT
WRITE (6,260) TOTPTS
WRITE (6,270)
DO 30 J=1,LNCAT
WRITE (6,280) J,SYMBLS(J),N(J)
30 CONTINUE
WRITE (6,290)
ISTART = 1
IEND = 12
LOOPCT = NOFEAT / 12
LOOPC1 = MOD(NOFEAT,12)
IF (LOOPC1.GT.0) LOOPCT = LOOPCT + 1
IF (LOOPCT.EQ.1) IEND = NOFEAT
DO 45 M=1,LOOPCT
WRITE (6,300) (BLANK,FETVEC(J),J=ISTART,IEND)
DO 40 J=1,LNCAT
WRITE (6,310) J,(MEANS(I,J),I=ISTART,IEND)
40 CONTINUE
WRITE (6,315)
315 FORMAT( )
ISTART = IEND + 1
IEND = ISTART + IEND - 1
IF (IEND.GT. NOFEAT) IEND = NOFEAT
45 CONTINUE
ISTART = 1
IEND = 12
IF (LOOPCT.EQ.1) IFND = NOFEAT
WRITE (6,320)
DO 55 M=1,LOOPCT
WRITE (6,300) (BLANK,FETVEC(J),J=ISTART,IEND)
DO 50 J=1,LNCAT
PRI00010
PRI00020
PRI00030
PRI00040
PRI00050
PRI00060
PRI00070
PRI00080
PRI00090
PRI00100
PRI00110
PRI00120
PRI00130
PRI00140
PRI00150
PRI00160
PRI00170
PRI00180
PRI00190
PRI00200
PRI00210
PRI00220
PRI00230
PRI00240
PRI00250
PRI00260
PRI00270
PRI00280
PRI00290
PRI00300
PRI00310
PRI00320
PRI00330
PRI00340
PRI00350
PRI00360
PRI00370
PRI00380
PRI00390
PRI00400
PRI00410
PRI00420
PRI00430
PRI00440
PRI00450
PRI00460
PRI00470
PRI00480
PRI00490
PRI00500
PRI00510
PRI00520
PRI00530
PRI00540
PRI00550
PRI00560
PRI00570
PRI00580
PRI00590
PRI00600
PRI00610
PRI00620
PRI00630
PRI00640
PRI00650
PRI00660
PRI00670
PRI00680
PRI00690
PRI00700
PRI00710
PRI00720
PRI00730
PRI00740
PRI00750
PRI00760
PRI00770
PRI00780
PRI00790
```

19-61  
434

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: PRINT

```
WRITE (6,310) J, (STDEV(I,J), I=ISTART, IEND)
50 CONTINUE
WRITE (6,315)
  ISTART = IEND + 1
  IEND = ISTART + IEND - 1
  IF (IEND .GT. NOFEAT) IEND = NOFEAT
55 CONTINUE
  L=1
  J=LNCAT
  IF (J.GT.15) J=15
60 WRITE (6,340) (K,K=L,J)
  DO 70 I=1,LNCAT
70 WRITE (6,350) I, (CLD(I,K), K=L,J)
  IF (J.EQ.LNCAT) GO TO 80
  L=L+15
  J=J+15
  IF (J.GE.LNCAT) J=LNCAT
  GO TO 60
80 CONTINUE
  IF (KKT.EQ.-1) GO TO 90
  IF (MOD(KKT, MAP).NE.0) RETURN
90 CONTINUE
  IRC=IRD
  ICCT=NOPTS
  IF (IRD.EQ.0) ICCT=KPTS
  IF (IRD.EQ.0) GO TO 110
  ADRES2=HEGIN2
105 CALL RREAD(ADRES2, IPLACE, ICCT, ISTAT)
  IF (ISTAT.EQ.1) GO TO 105
  ADRES2=ADRES2+ICCT
  IRFC=1
110 JPTS=0
  IRC=IPC-1
  CALL SFTMPG(66,0,66)
  IV=5
  DO 200 IFLD=1,NOFLD
C* 200 IFLD=1,NOFLD
  ZFRO NMLK
  DO 115 I=1,LNCAT
115 NMLK(I)=0
  FLDNAM = FLDINF(IV)
  NV=FLDINF(IV+1)
  IR=IV*2+NV*2
  DO 116 I=1,6
116 FINF(I)=FLDINF(IR+I)
  J=0
120 DO 130 I=SAMSTR,SAMEND,SAMINC
  J=J+1
  COL(1,J)=I/100
  COL(2,J)=MOD(I,100)/10
  COL(3,J)=MOD(I,10)
  IF (J.EQ.110) GO TO 140
130 CONTINUE
140 LPTS=J
  WRITE (6,220)
  WRITE (6,HEAD)
  TPTS=FLDINF(IV+NV*2+2)
  WRITE (6,330) FLDNAM, TPTS
  DO 150 I=1,3
150 WRITE (6,210) (COL(I,J), J=1,LPTS)
  WRITE (6,220)
  DO 180 I=LINSTR,LINEND,LININC
C* FIND FIELD INTERSECTIONS FOR THIS LINE
  CALL FDLINT(FLDINF(IV+2), NV, FL, I, PTS, NFL)
  DO 155 J=1,110
155 OUT(J)=HLANK
  DO 175 IJ=1,NFL,2
  IR=(FL(IJ)-SAMSTR)/SAMINC + 1
  IF=(FL(IJ+1)-SAMSTR)/SAMINC + 1
  IF (MOD(SAMSTR,SAMINC).NE.MOD(FL(IJ),SAMINC)) IB=IB+1
  IF (IR.GT.IE) GO TO 175
  DO 170 J=IR,IE
  JPTS=JPTS+1
  IF (JPTS.LE.ICCT) GO TO 160
  IF (IRC.EQ.1) ICCT=KPTS
  CALL RREAD(ADRES2, IPLACE, ICCT, ISTAT)
  ADRES2=ADRES2+ICCT
156 IF (ISTAT.EQ.1) GO TO 156
  IRC=IPC-1
  JPTS=1
```

PRI00800  
PRI00810  
PRI00820  
PRI00830  
PRI00840  
PRI00850  
PRI00860  
PRI00870  
PRI00880  
PRI00890  
PRI00900  
PRI00910  
PRI00920  
PRI00930  
PRI00940  
PRI00950  
PRI00960  
PRI00970  
PRI00980  
PRI00990  
PRI01000  
PRI01010  
PRI01020  
PRI01030  
PRI01040  
PRI01050  
PRI01060  
PRI01070  
PRI01080  
PRI01090  
PRI01100  
PRI01110  
PRI01120  
PRI01130  
PRI01140  
PRI01150  
PRI01160  
PRI01170  
PRI01180  
PRI01190  
PRI01200  
PRI01210  
PRI01220  
PRI01230  
PRI01240  
PRI01250  
PRI01260  
PRI01270  
PRI01280  
PRI01290  
PRI01300  
PRI01310  
PRI01320  
PRI01330  
PRI01340  
PRI01350  
PRI01360  
PRI01370  
PRI01380  
PRI01390  
PRI01400  
PRI01410  
PRI01420  
PRI01430  
PRI01440  
PRI01450  
PRI01460  
PRI01470  
PRI01480  
PRI01490  
PRI01500  
PRI01510  
PRI01520  
PRI01530  
PRI01540  
PRI01550  
PRI01560  
PRI01570  
PRI01580

~~19-62~~  
435

FILE: PRINT

```
160 CONTINUE
    K=IPLACE(JPTS)
    NBLK(K)=NBLK(K)+1
    LINE=I
    IF (J.GT.110) GO TO 170
    OUT(J)=SYMBLS(K)
170 CONTINUE
175 CONTINUE
    WRITE (6,230)LINE,(OUT(J),J=1,LPTS)
180 CONTINUE
    IV=IV + NV*2 + 9
    WRITE (6,370)
    DO 190 I=1,LNCAT
190 WRITE (6,380)I,SYMBLS(I),NBLK(I)
200 CONTINUE
    IF (DODU.EQ.0) GO TO 205
    SYMPTX(LNCAT-DODU+DOFLAG)=SAVEP
    IF (DUFLAG.NE.0) SYMPTX(LNCAT) = SAVEB
205 CONTINUE
    CALL SETMPG(66,4,62)
    RETURN
210 FORMAT(9X,11011)
220 FORMAT(//)
230 FORMAT(2X,I5,2X,110A1)
240 FORMAT(///' INTERMEDIATE PRINTOUT FOR ITERATION',I5//)
245 FORMAT(///' FINAL CLUSTER SUMMARY FOR CLASS',I1X,A4//)
250 FORMAT(///' TOTAL NUMBER OF CLUSTERS =',I3)
260 FORMAT(///' TOTAL NUMBER OF POINTS =',I7)
270 FORMAT(///' CLUSTER      SYMROL      POINTS IN CLUSTER')
280 FORMAT(4X,I2,9X,A1,10X,I7)
290 FORMAT(///15X,'MEANS'//)
300 FORMAT(//2X,'CLUSTER',5X,I2(A1,'CH(',I2,')',I1X))
310 FORMAT(5X,I2,7X,I2(F7.2,I1X))
320 FORMAT(///10X,' STANDARD DEVIATIONS'//)
330 FORMAT(///2X,A4,///' TOTAL NUMBER OF POINTS IN THIS FIELD',I7//)
    *)
340 FORMAT(///15X,'DISTANCES BETWEEN CLUSTERS'//1X,'CLUSTER',15I8)
350 FORMAT(3X,I2,5X,15F8,2)
370 FORMAT(///2X,'POINTS PER CLUSTER IN THIS FIELD'/3X,'CLUSTER',
    * 5X,'SYMBOL',5X,'POINTS'//)
380 FORMAT(6X,I2,10X,A1,7X,I5)
    END
```

PRI01590  
PRI01600  
PRI01610  
PRI01620  
PRI01630  
PRI01640  
PRI01650  
PRI01660  
PRI01670  
PRI01680  
PRI01690  
PRI01700  
PRI01710  
PRI01720  
PRI01730  
PRI01740  
PRI01750  
PRI01760  
PRI01770  
PRI01780  
PRI01790  
PRI01800  
PRI01810  
PRI01820  
PRI01830  
PRI01840  
PRI01850  
PRI01860  
PRI01870  
PRI01880  
PRI01890  
PRI01900  
PRI01910  
PRI01920  
PRI01930  
PRI01940  
PRI01950  
PRI01960  
PRI01970  
PRI01980  
PRI01990  
PRI02000

FILE: PRTCOV

```
      SURROUTINE PRTCOV(COVMTX,AVEMTX,CV1,AV1,CLSMTX)
C WRITE HEADING FOR TRANSFORMED COVARIANCE MATRIX
      IMPLICIT INTEGER(A-Z)
      REAL COVMTX(CV1,NOSUB2),AVEMTX(AV1,NOSUB2)
C
C
C
      INCLUDE COMRK1.LIST
      INCLUDE COMRK4.LIST
      INCLUDE COMRK6.LIST
      INCLUDE COMRK9.LIST
      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VAR5Z2,TOTVT2,NOFLD2,
      AVAP2,COVAR2,CLS1D2,SURN02,SURDS2,FLDSV2,VERTX2,
      FETVC2(30),SURVC2(75),SURPTR(75),CLSV2(60),
      KFPPTS(40),NOGRP,GRPNAM(60),GRPDEX(61),
      GRPCHK(61),GPOUPS(124)
      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
      EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
      (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,RMKEY,
      HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
      DRUMAD,DRUMDS,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVFL
      .NHSTUN,NHSTFI,SCTRUN,MAPFIL
      .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
      CHDUNT,PRTUNT,RANDIO
C DATA TRANSFORMATION COMMON BLOCK
      COMMON/TRBLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30)
CSEND
C
C
      DIMENSION CLSMTX(NOSUB2)
C
      DATA BCDTWO/'2'/'
C
      CNT=7+(5+3*2*AV1)*((AV1+11)/12)
      INC=PAGSIZ/CNT
      INC=CNT
      DO 1 ICLAS=1,NOSUB2
      IF(INC.LT.CNT) GO TO 2
      WRITE(6,HEAD)
      INC=0
      2 WRITE(6,3) CLSMTX(ICLAS)
      3 FORMAT(/' SUBCLASS ',A4)
      DO 4 LOC=1,AV1,12
      STOP=LOC+11
      IF(STOP.GT.AV1) STOP=AV1
      4 WRITE(6,5) (AVEMTX(I,ICLAS),I=LOC,STOP)
      5 FORMAT(' MEAN',3X,12F9.2)
      WRITE(6,6)
      6 FORMAT(' COVARIANCE MATRIX')
      CALL WRMTX(COVMTX(1,ICLAS),AV1,BCDTWO)
      INC=INC+1
      1 CONTINUE
      RETURN
      END
```

PRT00010  
PRT00020  
PRT00030  
PRT00040  
PRT00050  
PRT00060  
PRT00070  
PRT00080  
PRT00090  
PRT00100  
PRT00110  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00010  
COM00020  
COM00030  
COM00030  
COM00010  
COM00020  
COM00030  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00060  
  
PRT00130  
PRT00140  
PRT00150  
PRT00160  
PRT00170  
PRT00180  
PRT00190  
PRT00200  
PRT00210  
PRT00220  
PRT00230  
PRT00240  
PRT00250  
PRT00260  
PRT00270  
PRT00280  
PRT00290  
PRT00300  
PRT00310  
PRT00320  
PRT00330  
PRT00340  
PRT00350  
PRT00360  
PRT00370  
PRT00380  
PRT00390

FILE: RANK

```

      SUBROUTINE RANK(NOFEAT,FETVC2,LNCAT,MEANS,IPTT)
      IMPLICIT INTEGER(A-X)
      REAL MEANS(NOFEAT,LNCAT),SAVE,G(60)
      DIMENSION FETVC2(26),IPTT(LNCAT)
C
      INCLUDE COMMR4
      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,BMKEY,
      * HISFIL,HISKEY,TRFORM,ERIPTP,EPPKEY,MAPUNT,NOFILE,
      * DRUMAD,DRMWDOS,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVFL
      * ,NHSTIIN,NHSTFI,SCTRUN,MAPFIL
      * ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
      * CRDUNT,PRTUNT,RANDIO
C$END
      IF(MOD(NOFEAT,4))10,20,10
10    WRITE(6,500)
      GO TO 99
20    DO 30 I=1,LNCAT
      IPTT(I)=I
30    CONTINUE
      DO 40 J=1,LNCAT
      G(J)=0
40    CONTINUE
      DO 50 I=1,NOFEAT,NCHPAS
      DO 50 J=1,LNCAT
      G(J)=G(J)+(-0.24*MEANS(I,J)-0.56*MEANS(I+1,J)+0.6*MEANS(I+2,J)
      *0.49*MEANS(I+3,J))
50    CONTINUE
      J=0
60    J=J+1
      IF(J.GT.LNCAT)GO TO 90
      IF(J.EQ.LNCAT)GO TO 75
      IF(G(J).LT.G(J+1))GO TO 70
70    GO TO 60
      SAVE=G(J)
      G(J)=G(J+1)
      G(J+1)=SAVE
      ISAVE=IPTT(J)
      IPTT(J)=IPTT(J+1)
      IPTT(J+1)=ISAVE
75    K=J
80    IF(K.EQ.1)GO TO 60
      IF(G(K).LT.G(K-1))GO TO 60
      SAVE=G(K-1)
      G(K-1)=G(K)
      G(K)=SAVE
      ISAVE=IPTT(K-1)
      IPTT(K-1)=IPTT(K)
      IPTT(K)=ISAVE
      K=K-1
      GO TO 80
90    CONTINUE
99    WRITE(6,510)(I,IPTT(I),G(I),I=1,LNCAT)
      CONTINUE
      RETURN
500  FORMAT(1X,'THE NUMBER OF CHANNELS ARE NOT A MULTIPLE OF 4.
      *THE COLOR KEYS WILL BE ORDERED BY CLUSTER NUMBER.')
```

```

      RAN00010
      RAN00020
      RAN00030
      RAN00040
      RAN00050
      COM00010
      COM00020
      COM00030
      COM00040
      COM00050
      COM00060
      RAN00070
      RAN00080
      RAN00090
      RAN00100
      RAN00110
      RAN00120
      RAN00130
      RAN00140
      RAN00150
      RAN00160
      RAN00170
      RAN00180
      RAN00190
      RAN00200
      RAN00210
      RAN00220
      RAN00230
      RAN00240
      RAN00250
      RAN00260
      RAN00270
      RAN00280
      RAN00290
      RAN00300
      RAN00310
      RAN00320
      RAN00330
      RAN00340
      RAN00350
      RAN00360
      RAN00370
      RAN00380
      RAN00390
      RAN00400
      RAN00410
      RAN00420
      RAN00430
      RAN00440
      RAN00450
      RAN00460
      RAN00470
      RAN00480
      RAN00490
      RAN00500
      RAN00510
      RAN00520
```

FILE: RDDOTS

```
C
C      UTILITY ROUTINE THAT READS THE DOTFIL
C
C      SUBROUTINE RDDOTS (DOTS, DOTVEC, TOTDT3, TYPST, SIZE, TOTDT2,
C      NOCAT, CATNAM, NOFEAT, FETVC2, NOFEAT, FETVEC, NOSUN, ANGLE,
C      NOFLD, TOTVRT, FLDSAV, VERTEX, /KVAR/)
C      IMPLICIT INTEGER (A-Z)
C      DIMENSION KVAR (SIZE, 1)
C      REAL KVAR
C
C      TYPST = 1 - RETURNS SPECTRAL INFO
C      TYPST = 2 - RETURNS SPATIAL INFO
C      TYPST = 3 - RETURNS BOTH SPATIAL AND SPECTRAL INFO
C
C      DATA BLANK/ ' /
C      INCLUDE COMMK6.LIST
C      COMMON /GLOBAL/ HFAI(63), MAPTAP, DATAPE, SAVTAP, RMFILE, RMKEY,
C      HISFIL, HISKEY, TRFORM, ERIPTR, ERPKEY, MAPUNT, NOFILE,
C      DRUMAD, DRMADS, PAGESZ, DATEFIL, STAFIL, ASAV, ASAVL
C      NHSTJN, NHSTFI, SCTRUN, MAPFIL
C      DOTUNT, DOTFIL, RCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
C      CRDUNT, PRUNT, HANDIO
C
C      DIMENSION CATNAM(1), DOTVEC(1), DOTS(SIZE, 1)
C      DIMENSION FETVEC(30), FETVC3(70), FETVC2(1)
C      DIMENSION FLDSAV(4, 1), VERTEX(2, 1), ANGLE(1)
C      DIMENSION TEMDOT(5000)
C
C      READ REC NO. 1 FOR INDICES
C
C      REWIND (DOTUNT)
C      CALL FSRFSL (DOTUNT, DOTFIL, ISTAT)
C
C      READ (DOTUNT) NOCAT, NOFEAT, NOFLD, TOTVRT, TOTDOT, NOSUN, (CATNAM(I),
C      I=1, NOCAT), SIZE
C
C      COMPUTE ADDRESSES FOR ARRAY
C
C      DOTS1 = 1
C
C      READ REC. NO. 2
C
C      IF (TYPST .EQ. 1) READ (DOTUNT) (FETVEC(I), I=1, NOFEAT)
C      IF (TYPST .EQ. 2) READ (DOTUNT) DUMMY
C      IF (TYPST .EQ. 3) READ (DOTUNT) (FETVEC(I), I=1, NOFEAT)
C      ((FLDSAV(I, J), I=1, 4), J=1, NOFLD), ((VERTEX(I, J), I=1, 2),
C      J=1, TOTVRT), (ANGLE(I), I=1, NOSUN)
C
C      IF (TYPST .EQ. 2) GO TO A7
C      IF (NOFEAT .NE. 0) GO TO 60
C
C      SET DEFAULT CHANNELS
C
C      DO 50 I=1, NOFEAT
C      FETVC2(I) = 1
C 50 FETVC3(I) = 1
C      NOFEAT = NOFEAT
C      GO TO A7
C 60 DO 60 J=1, NOFEAT2
C      DO 70 K=1, NOFEAT
C      IF (FETVEC(K) .EQ. FETVC2(J)) GO TO 75
C 70 CONTINUE
C      WRITE (A, 85) FETVC2(J), (FETVEC(I), I=1, NOFEAT)
C 85 FORMAT (// ' CHANNEL ', I2, ' IS NOT ON DOTFIL % CHANNELS ARE ', 30I3)
C      CALL CMFHW
C 75 FETVC3(J) = K
C
C 90 CONTINUE
C  A7 IF (TOTDT3 .EQ. 0) GO TO 96
C
C      CODE ADDED NOV 21, 1978 FOR LIST PROCESSING
C      IF (TOTDT3.GT.TOTDOT) TOTDT3 = TOTDOT
C
C      DO 95 J=1, TOTDT3
C
C RDD00010
C RDD00020
C RDD00030
C RDD00040
C RDD00050
C RDD00060
C RDD00070
C RDD00080
C RDD00090
C RDD00100
C RDD00110
C RDD00120
C RDD00130
C RDD00140
C RDD00150
C RDD00160
C RDD00170
C RDD00180
C RDD00190
C RDD00200
C RDD00210
C RDD00220
C RDD00230
C RDD00240
C RDD00250
C RDD00260
C RDD00270
C RDD00280
C RDD00290
C RDD00300
C RDD00310
C RDD00320
C RDD00330
C RDD00340
C RDD00350
C RDD00360
C RDD00370
C RDD00380
C RDD00390
C RDD00400
C RDD00410
C RDD00420
C RDD00430
C RDD00440
C RDD00450
C RDD00460
C RDD00470
C RDD00480
C RDD00490
C RDD00500
C RDD00510
C RDD00520
C RDD00530
C RDD00540
C RDD00550
C RDD00560
C RDD00570
C RDD00580
C RDD00590
C RDD00600
C RDD00610
C RDD00620
C RDD00630
C RDD00640
C RDD00650
C RDD00660
C RDD00670
C RDD00680
C RDD00690
C RDD00700
C RDD00710
C RDD00720
C RDD00730
C RDD00740
C RDD00750
C RDD00760
C RDD00770
C RDD00780
C RDD00790
```



FILE: R0DOT1

```
      SUBROUTINE R0DOT1(TEMDOT,DOTS,KVAR,SIZES,TOTDT2,DOTVEC,FETVC3,
      SIZE,TOTDOT,TOTDT3,NOFET2,TYPSWT)
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION TEMDOT(1),DOTS(SIZES,1),DOTVEC(1),FETVC3(1)
      REAL KVAR(SIZES,1)
C
      TYPSWT = 1 -- RETRIEVE SPECTRAL INFO
      TYPSWT = 2 -- RETRIEVE SPATIAL INFO
      TYPSWT = 3 -- RETRIEVE SPECTRAL AND SPATIAL INFO
C
      GO TO (130,150,140),TYPSWT
C
      RETRIEVE SPECTRAL INFO
C
      130 CONTINUE
C
      PICK SURSET OF DOTS AND CHANNELS
C
      DO 140 K=1,TOTDT2
      KK = DOTVEC(K)
      DO 140 J=1,SIZES
      JJ = FETVC3(J)
      JJJ = (KK-1)*SIZE + JJ+4
      KVAR(J,K) = FLOAT(TEMDOT(JJJ))
      140 CONTINUE
      RETURN
C
      RETRIEVE SPATIAL INFO
C
      150 CONTINUE
      DO 170 I=1,TOTDT2
      KK = (I-1)*SIZE
      DO 170 K=1,SIZES
      DOTS(K,I) = TEMDOT(KK+K)
C
      RETURN
C
      140 CONTINUE
C
      RETRIEVE SPECTRAL AND SPATIAL INFO
C
      JJ = 0
      KK = 1
      DO 300 K=1,TOTDOT
      IF (KK.GT. TOTDT3) GO TO 190
      IF (K.EQ. DOTVEC(KK)) GO TO 190
      KK = KK+1
      GO TO 300
      190 JJ = JJ + 1
      JJJ = (K-1)*SIZE
      DO 200 J=1,4
      200 DOTS(J,JJ) = TEMDOT(JJJ+J)
C
      DO 210 J=1,NOFET2
      I = FETVC3(J)
      210 DOTS(4+J,JJ) = TEMDOT(JJJ+4+I)
      300 CONTINUE
      RETURN
C
      END
```

R0D00010  
R0D00020  
R0D00030  
R0D00040  
R0D00050  
R0D00060  
R0D00070  
R0D00080  
R0D00090  
R0D00100  
R0D00110  
R0D00120  
R0D00130  
R0D00140  
R0D00150  
R0D00160  
R0D00170  
R0D00180  
R0D00190  
R0D00200  
R0D00210  
R0D00220  
R0D00230  
R0D00240  
R0D00250  
R0D00260  
R0D00270  
R0D00280  
R0D00290  
R0D00300  
R0D00310  
R0D00320  
R0D00330  
R0D00340  
R0D00350  
R0D00360  
R0D00370  
R0D00380  
R0D00390  
R0D00400  
R0D00410  
R0D00420  
R0D00430  
R0D00440  
R0D00450  
R0D00460  
R0D00470  
R0D00480  
R0D00490  
R0D00500  
R0D00510  
R0D00520  
R0D00530  
R0D00540  
R0D00550  
R0D00560  
R0D00570  
R0D00580  
R0D00590  
R0D00600  
R0D00610

19-68  
441



FILE: ROMEAN

```

SURROUTINE RDMEAN(MENS)
IMPLICIT INTEGER(A-X)

THIS SUBROUTINE READS THE 'MEAN' CARD DECK OR FILE FOR ISOCLS.

INCLUDE COMRK4.LIST
INCLUDE COMPK5.LIST
INCLUDE COMRK6.LIST
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
COMMON/PASS/STOP,LNCAT,NMIN,KRN,STDMAX,DLMIN,SEP,
MAP,SPTRIG,IRD,KPTS,NOPTS,PUNCH,
* ICHN,CHNTHS,ICHAIN(62),NWDS,IREGIN,REGIN),
* RFGIN2,REGIN3,CLSNAM,NOFLD,IPT,TOTWRD,TOTPTS,
* NCLASS,NOCLS,TOTSUB,TOTFLD,TOTVRT,NOCL,NVRT
* ,NXTCLS,NOFFAT,MAXCLS,FETVEC(30),SYMMTX(62)
* ,VARSTZ,STATKY,ISCKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
* ,INPDER,INUNIT,INFILE,INITM,PMIN,SURVEC(62),NOSUB2,CHNVC(30)
* ,NOCHAN,ERCOMP,NOSEQ,MEAND0,MEANDU,
* SYMNO,SYMDU,ITRIGO,DOFLAG,
* DUFFLAG,DDDU,STDOTS(60),NSDOTS,SUNCOR(30),LLNCAT,
* DVERT(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2)
* ,MXFFT1,MAXPOP
REAL SUNCOR
COMMON/G...AL/HEAD(63),MARTAP,DATAP,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,EPPKEY,MAPUNT,NOFILF,
* DRUMAU,DRMWDOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVL
* ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,HANDIO

CSEND
COMMON/PASSA/NOFET1,FTVEC1(30)
DIMENSION MENS(30,1)
READ(CRDUNT,500) LNCAT,NOFET1,(FTVEC1(I), I=1,NOFET1)
DO 10 I=1,LNCAT
10 READ(CRDUNT,510) (MENS(J,I), J=1,NOFET1)
500 FORMAT(5X,15,15X,15/5X,30I2)
510 FORMAT(5X,5F15.8)
ADDRESS=IBEGIN
CALL PWRITE(ADDRESS,LNCAT,1,LSTAT)
ADDRESS=ADDRESS+1
CALL RWRITE(ADDRESS,NOFET1,1,LSTAT)
ADDRESS=ADDRESS+1
CALL RWRITE(ADDRESS,FTVEC1,NOFET1,LSTAT)
ADDRESS=ADDRESS+NOFET1
KW=MXFFT1*LNCAT
CALL RWRITE(ADDRESS,MENS,KW,JSTAT)
11 IF(JSTAT.EQ.1)GOTO 11
RETURN

C
ENTRY RDFILE(MFANS,MENS)
DIMENSION MEANS(NOFFAT,MAXCLS)
ADDRESS=IBEGIN
CALL RREAD(ADDRESS,LNCAT,1,LSTAT)
ADDRESS=ADDRESS+1
CALL RREAD(ADDRESS,NOFET1,1,LSTAT)
ADDRESS=ADDRESS+1
CALL RREAD(ADDRESS,FTVEC1,NOFET1,LSTAT)
ADDRESS=ADDRESS+NOFET1
KW=MXFFT1*LNCAT
CALL RREAD(ADDRESS,MENS,KW,JSTAT)
12 IF(JSTAT.EQ.1)GOTO 12

C*
C*
C*
STORE ONLY CHANNELS REQUESTED IN FETVEC

IF(LNCAT.GT.MAXCLS)LNCAT=MAXCLS
DO 40 J=1,NOFFAT
DO 30 K=1,NOFET1
IF(FTVEC(J).NE.FTVEC1(K))GO TO 30
DO 20 I=1,LNCAT
20 MEANS(J,I) = MENS(K,I)
GO TO 40
30 CONTINUE
WRITE(6,100)FETVEC(J)
DO 35 I=1,LNCAT
35 MFANS(J,I) = 50. * I*10.
40 CONTINUE

```

RDM00010  
RDM00020  
RDM00030  
RDM00040  
RDM00050  
RDM00060  
RDM00070  
RDM00080  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00060  
COM00070  
COM00080  
COM00090  
COM00100  
COM00110  
COM00120  
COM00130  
COM00140  
COM00150  
COM00160  
COM00170  
COM00180  
COM00190  
COM00200  
COM00210  
COM00220  
COM00230  
COM00240  
COM00250  
COM00260  
COM00270  
COM00280  
COM00290  
COM00300  
COM00310  
COM00320  
COM00330  
COM00340  
COM00350  
COM00360  
COM00370  
COM00380  
COM00390  
COM00400  
COM00410  
COM00420  
COM00430  
COM00440  
COM00450  
COM00460  
COM00470  
COM00480  
COM00490  
COM00500  
COM00510  
COM00520  
COM00530  
COM00540  
COM00550  
COM00560  
COM00570

FILE: RDMEAN

```
C*
C* PRINT INITIAL CLUSTER CENTERS RDM00580
C* WRITE(6,200) RDM00590
IR=1 RDM00600
IF=12 RDM00610
45 IF(NOFFAT.LT.IE)IE=NOFEAT RDM00620
WRITE(6,300)(FFIVEC(J),J=IB,IE) RDM00630
DO 50 I=1,LNCAT RDM00640
50 WRITE(6,400)I.(MEANS(J,I),J=IB,IE) RDM00650
IF(IF.EQ.NOFEAT)GO TO 60 RDM00660
IR=IR+12 RDM00670
IF=IF+12 RDM00680
GO TO 45 RDM00690
60 RETURN RDM00700
100 FORMAT(' MEANS FOR CHANNEL',I4,' ARE NOT ON FILE--DUMMY VALVES WILL RDM00710
*L RE USED') RDM00720
200 FORMAT(///15X,'INITIAL CLUSTER MEANS') RDM00730
300 FORMAT(2X,'CLUSTER',2X,12(1X,'CH(',12,')',1X)) RDM00740
400 FORMAT(5X,12,5X,12(F6.2,2X)) RDM00750
END RDM00760
RDM00770
RDM00780
```

~~18-70~~  
443



FILE: RDMODK

```

SUBD2 = FLDSV1 - 1
WRITE(SAVTAP)(ARRAY(I),I=CLSID1,SUBD2)
7 CONTINUE
C
  MEAN2 = CLSID1 - 1
  DO 10 I=1,NOSUR
  READ(CRDUNT,240) KEPPTS(I)
  READ(CRDUNT,280) (AVAR(J),J=1,NOFEAT)
  READ(CRDUNT,280) (COVAR(J),J=1,VARSIZ)
C
  IF (ISTAT .GT. 0) GO TO 10
CI
  WRITE(SAVTAP)KEPPTS(I),(ARRAY(J),J=1,MEAN2)
  10 CONTINUE
CI
  IF (ISTAT .GT. 0) STAFIL = -1
  IF (ISTAT .GT. 0) RETURN
CI
  FND FILE SAVTAP
  RETURN
C
215 FORMAT(A4.6X,I2.8X,I2.8X,I2)
220 FORMAT(.10X,14I5)
230 FORMAT((6X,9(2X,A4.2X)))
240 FORMAT((7X,24(1X,I2)))
250 FORMAT((8X,10(A4.3X)))
260 FORMAT(12X,1A)
270 FORMAT(6X,I2F6.2)
280 FORMAT(5X,5E15.8)
END
```

RDM00800  
RDM00810  
RDM00820  
RDM00830  
RDM00840  
RDM00850  
RDM00860  
RDM00870  
RDM00880  
RDM00890  
RDM00900  
RDM00910  
RDM00920  
RDM00930  
RDM00940  
RDM00950  
RDM00960  
CIRDM00970  
RDM00980  
RDM00990  
RDM01000  
RDM01010  
RDM01020  
RDM01030  
RDM01040  
RDM01050  
RDM01060  
RDM01070  
RDM01080  
RDM01090

~~19-72~~  
445

FILE: RENDAT

```

SURROUTINE RENDAT(COVAR,AVAR,CLSDFS,SURNO,SURDES,FLDSAV,VERTEX,
* COV,AVEN,CLSDFS,SURNOS,SURDS,FLDSV,VERTX,
* NOFEAT,VARSI2,NOCLS,NOFLD,NOSUB,FETVEC)
IMPLICIT INTEGER (A-Z)
DIMENSION FETVEC(30)
* * * * *
READS COVARIANCES AND MEANS FROM FILE AND REDUCES STATS
* * * * *

INCLUDE COMRK1.LIST
INCLUDE COMRK6.LIST
COMMON/INFOPM/NOCLS2,NOSUB2,NOFET2,VARSI2,TOTVT2,NOFLD2,
* AVAR2,COVAR2,CLSID2,SURNO2,SURDS2,FLDSV2,VERTX2,
* FETVC2(30),SURVC2(75),SUBPTR(75),CLSV2(60),
* KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWDOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* .NHSTUN,NHSTFI,SCTRUN,MAPFIL
* .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
CSEND
COMMON/RESTKN/KFPPTS(60),JPRIOR,KBEST,NCPASS
REAL COVAR(VARSZ2,NOSUB2),AVAR(NOFEAT,NOSUB),COV(VARSIZ),
* AVEN(NOFEAT,NOSUB),R
DIMENSION CLSDFS(NOCLS),SURNO(NOCLS),SURDES(NOSUB),CLSDFS(NOCLS2),
* SURNOS(NOCLS2),SURDS(NOSUR2),FLDSV(4,NOFLD2),
* VERTX(2,TOTVT2),FLDSAV(4,NOFLD),VERTEX(2,TOTVT2)
* .DUMVEC(30)
REDUCE CLASS DESCRIPTION AND ARRAY CONTAINING NO OF SUBCLASSES
DO 150 I=1,NOCLS2
CLSDFS(I) = CLSDFS(I)
150 SURNOS(I) = SURNO(I)
REDUCE SUBCLASS DESCRIPTIONS
DO 160 I=1,NOSUR2
160 SURDS(I) = SURDES(I)
REDUCE FIELD INFORMATION
DO 170 I=1,NOFLD2
DO 170 J=1,4
170 FLDSV(J,I) = FLDSAV(J,I)
REDUCE VERTICES
DO 180 I=1,TOTVT2
DO 180 J=1,2
180 VERTX(J,I) = VERTEX(J,I)
ZERO OUT JUST PORTION OF COVAR THAT WILL CONTAIN SUBCLASSES
THAT HAVE BEEN GROUPED
DO 200 J=1,NOGRP
KR = GRPDEX(J) + 1
KF = KR + GROUPS(KR-1) - 1
IF (KR .GE. KF) GO TO 200
KK = SUBPTR(KR)
DO 195 LL = 1,VARSI2
195 COVAR(LL,KK) = 0.0
200 CONTINUE
CHECK CLASSIFICATION CHANNELS AGAINST TRAINING CHANNELS
DO 220 J=1,NOFET2
DO 210 L=1,NOFEAT
IF ( FETVC2(J) .EQ. FETVEC(L) ) GO TO 220
210 CONTINUE
WRITE(6,230) FETVC2(J), (FETVEC(K),K=1,NOFEAT)
230 FORMAT(' ** CHANNEL NO. ',I2,' IS NO. A TRAINING CHANNEL **',
* 40X,' TRAINING CHANNELS ARE - ',/ 10X,30(I2,1X))

```

19-73  
446

FILE: REDDAT

```
220 CALL EXIT
DUMVEC(J) = L
DO 100 JJ=1,NOSUR
C IS THIS SUBCLASS A MEMBER OF SELECTED SUBCLASSES
IF (SURPTR(JJ) .LE. 0) HEAD(SAVTAP) DUMMY
IF (SURPT*(JJ) .LE. 0) GO TO 100
READ(SAVTAP)KEPPTS(JJ),COV,(AVAR(I,JJ),I=1,NOFEAT)
C
C
C REDUCE BY CHANNELS
NEWSUR = SURPTR(JJ)
KK = 0
DO 20 J=1,NOFET2
K = DUMVEC(J)
LOC = K*(K-1)/2
DO 40 L=1,1
KK = KK + 1
WAT = LOC + DUMVEC(L)
40 COV(KK) = COV(WAT)
20 AVAR(J,JJ) = AVAR(K,JJ)
C
C
C GROUP SUBCLASSES
IF (GRPCHK(JJ) .LE. 0) GO TO 60
KK = 0
DO 50 J1=1,NOFET2
DO 50 J2=1,J1
KK = KK + 1
R = COVAR(KK,NEWSUR)
COVAR(KK,NEWSUR) = R + (COV(KK)*(KEPPTS(JJ)-1)) +
* (AVAR(J1,JJ)*AVAR(J2,JJ)*KEPPTS(JJ))
50 CONTINUE
GO TO 100
60 DO 70 I=1,VARSZ2
70 COVAR(I,NEWSUR) = COV(I)
100 CONTINUE
C
C
C GROUP MEANS
DO 125 I=1,NOGRP
KH = GRPDEX(I) + 1
KE = KP + GROUPS(KH-1) - 1
KR1 = GROUPS(KH)
IF (KH .GE. KE) GO TO 130
DO 120 J=1,NOFET2
R = 0.0
KPTS = 0
DO 110 K=KR,KE
KR2 = GROUPS(K)
KPTS = KPTS + KEPPTS(KR2)
110 R = R + AVAR(J,KR2) * KEPPTS(KR2)
120 AVAR(J,KH1) = R / KPTS
KEPPTS(KH1)=KPTS
JJ = 0
NEWSUR = SURPTR(KR1)
DO 123 J1=1,NOFET2
DO 123 J2=1,J1
JJ = JJ + 1
123 COVAR(JJ,NEWSUR) = (COVAR(JJ,NEWSUR) - (AVAR(J1,KH1)*
* AVAR(J2,KH1))*KPTS) / (KPTS-1)
125 CONTINUE
C REDUCE MEANS
130 DO 140 K=1,NOSUB2
II = SUBVCP(K)
KEPPTS(K)=KEPPTS(II)
DO 140 J=1,NOFET2
140 AVEN(J,K) = AVAR(J,II)
C
C
C RETURN
END
```

RED00800  
RED00810  
RED00820  
RED00830  
RED00840  
RED00850  
RED00860  
RED00870  
RED00880  
RED00890  
RED00900  
RED00910  
RED00920  
RED00930  
RED00940  
RED00950  
RED00960  
RED00970  
RED00980  
RED00990  
RED01000  
RED01010  
RED01020  
RED01030  
RED01040  
RED01050  
RED01060  
RED01070  
RED01080  
RED01090  
RED01100  
RED01110  
RED01120  
RED01130  
RED01140  
RED01150  
RED01160  
RED01170  
RED01180  
RED01190  
RED01200  
RED01210  
RED01220  
RED01230  
RED01240  
RED01250  
RED01260  
RED01270  
RED01280  
RED01290  
RED01300  
RED01310  
RED01320  
RED01330  
RED01340  
RED01350  
RED01360  
RED01370  
RED01380  
RED01390  
RED01400  
RED01410  
RED01420  
RED01430  
RED01440  
RED01450  
RED01460  
RED01470  
RED01480  
RED01490  
RED01500  
RED01510

FILE: REDSAV

```

C      SURROUTINE REDSAV (ARRAY, TOP, BMFLG)                                RED000010
C                                                                              RED000020
C                                                                              RED000030
C      IMPLICIT INTEGER (A-Z)                                             RED000040
C      DIMENSION ARRAY (1)                                                RED000050
C                                                                              RED000060
C      INCLUDE COMPK1.LIST                                                RED000070
C                                                                              RED000080
C      INCLUDE COMPK4.LIST                                                RED000090
C      COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,      RED000100
C      *      AVAR2, COVAR2, CLSID2, SURNO2, SURDS2, FLDSV2, VERTX2,      RED000110
C      *      FELVC2 (30), SUHVC2 (75), SUHPTR (75), CLSVC2 (60),      RED000120
C      *      KFPPTS (60), NOGRP, GMPNAM (60), GRPDEX (61),      RED000130
C      *      GRPCHK (61), GROUPS (124)                                    RED000140
C      COMMON/GLOBAL/HEAD (63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,    RED000150
C      *      HISFIL, HISKEY, TRFORM, ERITP, ERPKEY, MAPUNT, NOFILE,    RED000160
C      *      DRUMAD, DRUMDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL      RED000170
C      *      NHSTUN, NHSTFI, SCTRIIN, MAPFIL                              RED000180
C      *      DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,    RED000190
C      *      CRDUNT, PRUNT, RANDIO                                       RED000200
CSEND  DIMENSION FETVEC (30)                                             RED000210
      REWIND SAVTAP                                                         RED000220
      IF (STAFIL.EQ.-1) WRITE (6,100)                                     RED000230
100    FORMAT (// T5, 'STAT FILE WAS NOT CREATED. EXITING FROM **REDSAV**') RED000240
      IF (STAFIL.EQ.-1) CALL CMERR                                         RED000250
      IF (STAFIL.EQ.0) GO TO 10                                           RED000260
      CALL FSHSFL (SAVTAP, STAFIL, ISTAT)                                  RED000270
      IF (ISTAT.EQ.0) GO TO 10                                           RED000280
      FILNO = STAFIL + 1                                                 RED000290
      WRITE (6,110) FILNO                                               RED000300
110    FORMAT (// T5, 'ERROR IN POSITIONING STAT FILE TO FILE ', I3,     RED000310
      *      ' EXITING FROM REDSAV')                                       RED000320
10    CONTINUE                                                           RED000330
      HEAD (SAVTAP) NOCLS, NOSUB, NOFEAT, NOFLD, TOTVRT, (FETVEC (I) .I=1, NOFEAT) RED000340
C                                                                              RED000350
C      COMPUTE BASES.                                                    RED000360
C                                                                              RED000370
C      VARSIZ = NOFEAT * (NOFEAT + 1) / 2 .                               RED000380
C      CLSID1 = 1                                                         RED000390
C      SUBNO1 = CLSID1 + NOCLS                                           RED000400
C      SURDS1 = SUBNO1 + NOCLS                                           RED000410
C      FLDSV1 = SURDS1 + NOSUB                                           RED000420
C      VERTX1 = FLDSV1 + NOFLD * 4                                       RED000430
C                                                                              RED000440
C      * CALL SAVFIL (ARRAY (FLDSV1), ARRAY (VERTX1), ARRAY (CLSID1),    RED000450
C      *      ARRAY (SUBNO1), ARRAY (SUBDS1), NOFLD, NOCLS, NOSUB)      RED000460
C                                                                              RED000470
C      * CALL CLSCHK (ARRAY (CLSID1), ARRAY (SUBDS1), ARRAY (FLDSV1),    RED000480
C      *      ARRAY (VERTX1), ARRAY (SUBNO1), NOFEAT, FETVEC,          RED000490
C      *      NOCLS, NOFLD, BMFLG, NOSUB)                                  RED000500
C                                                                              RED000510
C                                                                              RED000520
C                                                                              RED000530
C      COMPUTE REDUCED BASES                                             RED000540
C                                                                              RED000550
C      VARSZ2 = NOFET2 * (NOFET2 + 1) / 2                                RED000560
C      CLSID2 = 1                                                         RED000570
C      SURNO2 = CLSID2 + NOCLS2                                           RED000580
C      SURDS2 = SURNO2 + NOCLS2                                           RED000590
C      FLDSV2 = SURDS2 + NOSUR2                                           RED000600
C      VERTX2 = FLDSV2 + NOFLD2 * 4                                       RED000610
C      COVAR2 = VERTX2 + TOTVT2 * 2                                       RED000620
C      AVAR2 = COVAR2 + VARSZ2 * NOSUR2                                       RED000630
C      COV1 = AVAR2 + (NOFEAT * NOSUB)                                       RED000640
C      TIPTOP = COV1 + VARSIZ                                           RED000650
C      RANCOV = TOP - TIPTOP                                             RED000660
C      IF (RANCOV.LT.0) GO TO 50                                           RED000670
C                                                                              RED000680
C      * CALL REDDAT (ARRAY (COVAR2), ARRAY (AVAR2), ARRAY (CLSID1),      RED000690
C      *      ARRAY (SURDS1), ARRAY (FLDSV1), ARRAY (VERTX1),          RED000700
C      *      ARRAY (COV1), ARRAY (AVAR2), ARRAY (CLSID2),            RED000710
C      *      ARRAY (SURNO2), ARRAY (SURDS2), ARRAY (FLDSV2),          RED000720
C      *      ARRAY (VERTX2), NOFEAT, VARSIZ, NOCLS, NOFLD,          RED000730
C      *      NOSUB, FETVEC)                                             RED000740
C                                                                              RED000750
C                                                                              RED000760
C                                                                              RED000770
C      GO TO 70                                                           RED000780
50    WRITE (6,60) NOFET2, NOSUR2, NOCLS2                                RED000790
60    FORMAT (// 'USER HAS REQUESTED ', I2, ' CHANNELS, ', I2, ' SUBCLASSES, ', RED000780
      *      ' AND ', I2, ' CLASSES. // THIS COMBINATION OF STATS WILL NOT FIT IN' RED000790

```

FILE: REDSAV

\*CONF. PLEASE REDUCE REQUEST.)\*  
CALL CMERR

C  
70 CONTINUE  
RETURN  
END

RED00800  
RED00810  
RED00820  
RED00830  
RED00840  
RED00850

~~19-76~~

449



FILE: PREAD

SUBROUTINE PREAD(BEGADD,/WHERE/,TOTWDS,STATUS)

THIS SUBROUTINE SIMULATES THE RANDOM ACCESS READ OF A WORKFILE USED TO STORE PROGRAM DATA TEMPORARILY DURING A LARSYS PROCESSOR RUN. THE CALLING ARGUMENTS ARE:

BEGADD - THE NUMBER OF WORDS FROM THE BEGINNING OF THE FILE WHERE THE READ IS TO BEGIN.  
WHERE - WHERE THE DATA READ IS TO BE PUT (OUTPUT AREA).  
TOTWDS - THE TOTAL NUMBER OF WORDS TO BE READ.  
STATUS - SET TO ZERO WHEN I/O IS COMPLETE (NO LONGER USED, BUT MUST BE RETURNED AS 0).

IMPLICIT INTEGER (A-Z)

HANSEN / VERSION 0800/R/31/77

DIMENSION BUFFER(200),WHERE(1)  
BUFSIZ=200

BUFFER AND BUFSIZ ARE SET TO THE MOST EFFICIENT SIZE TO MATCH THE PHYSICAL RECORD SIZE OF THE I/O BUFFER. IN CMS IT IS 800 BYTES - 200 WORDS.

STATUS=0  
LUD=22

LUD IS THE LOGICAL UNIT NUMBER WHERE THE FORTRAN DIRECT ACCESS FILE IS STORED.

I1=1  
J1=MOD(BEGADD,BUFSIZ)  
IF(J1.EQ.0) J1=BUFSIZ

J1 IS THE RELATIVE ADDRESS OF THE BEGINNING WORD IN THE FIRST RECORD TO BE READ. IF IT IS 0, IT IS THE LAST WORD IN THE RECORD.

J2=BEGADD  
J3=BEGADD + TOTWDS - 1

J2 AND J3 ARE THE BEGINNING AND ENDING WORDS OF THE DATA TO BE READ.

J4=MOD(J3,BUFSIZ)  
IF(J4.EQ.0) J4=BUFSIZ

J4 IS THE RELATIVE ADDRESS OF THE ENDING WORD IN THE FINAL RECORD TO BE READ. IF IT IS 0, IT IS THE LAST WORD IN THE RECORD.

REGREC=((J2-1)/BUFSIZ) + 1  
ENDREC=((J3-1)/BUFSIZ) + 1  
IF(REGREC.EQ.ENDREC) GOTO 300

REGREC AND ENDREC ARE THE RELATIVE ADDRESSES (RECORD NUMBERS) OF THE FIRST AND LAST RECORDS TO BE READ. IF THEY ARE EQUAL THEN WE ARE TO BEGIN AND END IN THE SAME RECORD.

K1=REGREC

READ THE FIRST RECORD AND MOVE THE REQUIRED PORTION TO THE OUTPUT AREA.

READ(LUD\*K1) BUFFER  
DO 200 K2=J1,BUFSIZ  
WHERE(I1)=BUFFER(K2)  
I1=I1+1  
CONTINUE  
K1=K1+1

READ IN THE NEXT RECORD. IF IT IS THE FINAL RECORD TO BE READ, GO TO THE FINAL RECORD MOVE CODE ELSE MOVE THE ENTIRE RECORD TO THE OUTPUT AREA.

IF(K1.EQ.ENDREC) GOTO 230

RRE00010  
RRE00020  
RRE00030  
RRE00040  
RRE00050  
RRE00060  
RRE00070  
RRE00080  
RRE00090  
RRE00100  
RRE00110  
RRE00120  
RRE00130  
RRE00140  
RRE00150  
RRE00160  
RRE00170  
RRE00180  
RRE00190  
RRE00200  
RRE00210  
RRE00220  
RRE00230  
RRE00240  
RRE00250  
RRE00260  
RRE00270  
RRE00280  
RRE00290  
RRE00300  
RRE00310  
RRE00320  
RRE00330  
RRE00340  
RRE00350  
RRE00360  
RRE00370  
RRE00380  
RRE00390  
RRE00400  
RRE00410  
RRE00420  
RRE00430  
RRE00440  
RRE00450  
RRE00460  
RRE00470  
RRE00480  
RRE00490  
RRE00500  
RRE00510  
RRE00520  
RRE00530  
RRE00540  
RRE00550  
RRE00560  
RRE00570  
RRE00580  
RRE00590  
RRE00600  
RRE00610  
RRE00620  
RRE00630  
RRE00640  
RRE00650  
RRE00660  
RRE00670  
RRE00680  
RRE00690  
RRE00700  
RRE00710  
RRE00720  
RRE00730  
RRE00740  
RRE00750  
RRE00760  
RRE00770  
RRE00780  
RRE00790

FILE: RREAD

	READ(LUD*K1) BUFFER	RRE 00800
	DO 220 K2=1,RIFFSIZ	RRE 00810
	WHERE(I1)=BUFFER(K2)	RRE 00820
	I1=I1+1	RRE 00830
220	CONTINUE	RRE 00840
	GOTO 210	RRE 00850
C		RRE 00860
C	READ THE FINAL RECORD, MOVE THE REQUIRED PORTION TO THE	RRE 00870
C	OUTPUT AREA AND RETURN.	RRE 00880
230	READ(LUD*K1) BUFFER	RRE 00890
	DO 240 K2=1,J4	RPE 00900
	WHERE(I1)=BUFFER(K2)	RPE 00910
	I1=I1+1	RPE 00920
240	CONTINUE	RPE 00930
	RETURN	RPE 00940
C		RRE 00950
C	HERE WE BEGIN AND END IN THE SAME RECORD, THEREFORE WE	RRE 00960
C	ONLY MOVE THE REQUIRED PORTION OF THE DATA TO THE OUTPUT	RRE 00970
C	AREA AND RETURN.	RRE 00980
300	READ(LUD*HEGPEC) BUFFER	RPE 00990
	DO 310 K2=J1,J4	RRE 01000
	WHERE(I1)=BUFFER(K2)	RRE 01010
	I1=I1+1	RRE 01020
310	CONTINUE	RRE 01030
	RETURN	RPE 01040
	END	RRE 01050
		RRE 01060
		RRE 01070

FILE: RWRITE

SUBROUTINE RWRITE(BEGADD, /WHERE/, TOTWDS, STATUS)

THIS SUBROUTINE SIMULATES THE RANDOM ACCESS WRITE OF A WORKFILE USED TO STORE PROGRAM DATA TEMPORARILY DURING A LARSYS PROCESSOR RUN. THE CALLING ARGUMENTS ARE:

BEGADD - THE NUMBER OF WORDS FROM THE BEGINNING OF THE FILE WHERE THE WRITE IS TO BEGIN.  
WHERE - WHERE THE DATA TO BE WRITTEN IS STORED (INPUT AREA).  
TOTWDS - THE TOTAL NUMBER OF WORDS TO BE WRITTEN.  
STATUS - SET TO ZERO WHEN I/O IS COMPLETE (NO LONGER USED, BUT MUST BE RETURNED AS 0)

IMPLICIT INTEGER (A-Z)

HANSEN / VERSION 0800/8/31/77

DIMENSION BUFFER(200), WHEPE(1)  
RUFFSIZ=200

BUFFER AND RUFFSIZ ARE SET TO THE MOST EFFICIENT SIZE TO MATCH THE PHYSICAL RECORD SIZE OF THE I/O BUFFER. IN CMS IT IS 800 BYTES - 200 WORDS.

STATUS=0  
LUD=22

LUD IS THE LOGICAL UNIT NUMBER WHERE THE FORTRAN DIRECT ACCESS FILE IS STORED.

I1=1  
J1=MOD(BEGADD, RUFFSIZ)  
IF (J1.EQ.0) J1=RUFFSIZ

J1 IS THE RELATIVE ADDRESS OF THE BEGINNING WORD IN THE FIRST RECORD TO BE WRITTEN. IF IT IS 0, IT IS THE LAST WORD IN THE RECORD.

J2=BEGADD  
J3=BEGADD + TOTWDS - 1

J2 AND J3 ARE THE BEGINNING AND ENDING WORDS OF THE DATA TO BE WRITTEN.

J4=MOD(J3, RUFFSIZ)  
IF (J4.EQ.0) J4=RUFFSIZ

J4 IS THE RELATIVE ADDRESS OF THE ENDING WORD IN THE FINAL RECORD TO BE WRITTEN. IF IT IS 0, IT IS THE LAST WORD IN THE RECORD.

RFGREC=((J2-1)/RUFFSIZ) + 1  
ENDREC=((J3-1)/RUFFSIZ) + 1  
IF (RFGREC.EQ.ENDREC) GOTO 300

RFGREC AND ENDREC ARE THE RELATIVE ADDRESSES (RECORD NUMBERS) OF THE FIRST AND LAST RECORDS TO BE WRITTEN. IF THEY ARE EQUAL, THEN WE ARE TO BEGIN AND END IN THE SAME RECORD.

K1=RFGREC

READ THE FIRST RECORD (THERE MIGHT BE DATA IN THE PORTION OF THE RECORD WE ARE NOT WRITING). MOVE THE REQUIRED PORTION OF THE DATA FROM THE INPUT AREA TO THE BUFFER AND WRITE IT OUT.

READ(LUD\*K1) BUFFER  
DO 200 K2=J1, RUFFSIZ  
BUFFER(K2)=WHEPE(I1)  
I1=I1+1  
CONTINUE  
WRITE(LUD\*K1) BUFFER

200

PUMP THE RECORD COUNTER AND CHECK TO SEE WHETHER WE ARE AT THE FINAL RECORD TO BE WRITTEN. IF WE ARE, GO

RRE 00010  
RRE 00020  
RRE 00030  
RRE 00040  
RRE 00050  
RRE 00060  
RRE 00070  
RRE 00080  
RRE 00090  
RRE 00100  
RRE 00110  
RRE 00120  
RRE 00130  
RRE 00140  
RRE 00150  
RRE 00160  
RRE 00170  
RRE 00180  
RRE 00190  
RRE 00200  
RRE 00210  
RRE 00220  
RRE 00230  
RRE 00240  
RRE 00250  
RRE 00260  
RRE 00270  
RRE 00280  
RRE 00290  
RRE 00300  
RRE 00310  
RRE 00320  
RRE 00330  
RRE 00340  
RRE 00350  
RRE 00360  
RRE 00370  
RRE 00380  
RRE 00390  
RRE 00400  
RRE 00410  
RRE 00420  
RRE 00430  
RRE 00440  
RRE 00450  
RRE 00460  
RRE 00470  
RRE 00480  
RRE 00490  
RRE 00500  
RRE 00510  
RRE 00520  
RRE 00530  
RRE 00540  
RRE 00550  
RRE 00560  
RRE 00570  
RRE 00580  
RRE 00590  
RRE 00600  
RRE 00610  
RRE 00620  
RRE 00630  
RRE 00640  
RRE 00650  
RRE 00660  
RRE 00670  
RRE 00680  
RRE 00690  
RRE 00700  
RRE 00710  
RRE 00720  
RRE 00730  
RRE 00740  
RRE 00750  
RRE 00760  
RRE 00770  
RRE 00780  
RRE 00790

FILE: RWRITE

```
C TO THE FINAL RECORD WRITE CODE ELSE MOVE AND WRITE
C THE ENTIRE RECORD.
C
210 K1=K1+1
    IF (K1.EQ.ENDREC) GOTO 230
    DO 220 K2=1,RUFSIZ
    RUFFER(K2)=WHERE(I1)
    I1=I1+1
220 CONTINUE
    WRITE(LUD*K1) RUFFER
    GOTO 210
C
C READ THE FINAL RECORD. MOVE THE REQUIRED PORTION OF THE
C DATA FROM THE INPUT AREA TO THE BUFFER, WRITE IT OUT
C AND RETURN.
C
230 READ(LUD*K1) BUFFER
    DO 240 K2=1,J4
    RUFFER(K2)=WHERE(I1)
    I1=I1+1
240 CONTINUE
    WRITE(LUD*K1) RUFFER
    RETURN
C
C HERE WE BEGIN AND END IN THE SAME RECORD. THEREFORE WE
C ONLY MOVE THE REQUIRED PORTION OF THE DATA FROM THE
C INPUT AREA TO THE BUFFER, WRITE IT OUT AND RETURN.
C
300 READ(LUD*REGREC) BUFFER
    DO 310 K2=J1,J4
    RUFFER(K2)=WHERE(I1)
    I1=I1+1
310 CONTINUE
    WRITE(LUD*REGREC) RUFFER
    RETURN
END
```

```
RRE00800
RRE00810
RRE00820
RRE00830
RRE00840
RRE00850
RRE00860
RRE00870
RRE00880
RRE00890
RRE00900
RRE00910
RRE00920
RRE00930
RRE00940
RRE00950
RRE00960
RRE00970
RRE00980
RRE00990
RRE01000
RRE01010
RRE01020
RRE01030
RRE01040
RRE01050
RRE01060
RRE01070
RRE01080
RRE01090
RRE01100
RRE01110
RRE01120
RRE01130
RRE01140
RRE01150
```

FILE: SAVFIL

```
      SUBROUTINE SAVFIL (FLDSAV, VERTEX, CLSID, SUBNO, SUBDES,  
      NOFLD, NOCLS, NOSUB)  
C  
      IMPLICIT INTEGER (A-Z)  
C      INCLUDE COMRK1.LIST  
C      INCLUDE COMRK2.LIST  
      COMMON/INFORM/NOCLS?, NOSUR?, NOFET?, VARSZ?, TOTVT?, NOFLD?,  
      AVAR?, COVAR?, CLSID?, SUBNO?, SURDS?, FLDSV?, VERTX?,  
      FETVC2(30), SUHVC2(75), SURPTR(75), CLSVC2(60),  
      KEPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),  
      GRPCHK(61), GROUPS(124)  
      COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, RMKEY,  
      HISTFIL, HISKEY, TRFORM, ERPTP, ERPKY, MAPUNT, NOFILE,  
      DRUMAD, DRMWUS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL,  
      NHSTUN, NHSTFI, SCTRUN, MAPFIL,  
      DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,  
      CRDUNT, PRUNT, RANDIO  
CSEND  
      DIMENSION FLDSAV(4, NOFLD), VERTEX(1), CLSID(1), SUBNO(1),  
      SUBDES(1)  
      IR=0  
      DO 40 J=1, NOFLD  
      READ(SAVTAP) (FLDSAV(I, J), I=1, 4)  
      FLD = 2 * FLDSAV(4, J)  
      READ(SAVTAP) (VERTEX(I+IR), I=1, FLD)  
      IR = IR + FLD  
      CONTINUE  
      READ(SAVTAP) (CLSID(I), I=1, NOCLS), (SUBNO(I), I=1, NOCLS),  
      (SUBDES(I), I=1, NOSUB)  
      RETURN  
C  
      END
```

SAV00010  
SAV00020  
SAV00030  
SAV00040  
SAV00050  
SAV00060  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00010  
COM00020  
COM00030  
COM00040  
COM00050  
COM00060  
SAV00070  
SAV00080  
SAV00090  
SAV00100  
SAV00110  
SAV00120  
SAV00130  
SAV00140  
SAV00150  
SAV00160  
SAV00170  
SAV00180  
SAV00190  
SAV00200  
SAV00210  
SAV00220

FILE: SEARCH

```

SUBROUTINE SEARCH(*,*,ENDTAP,IRUF,IRPDS,NDSPR)
IMPLICIT INTEGER (A-Z)
C*
C* INTERNAL ROUTINE TO SEARCH FOR CORRECT SCAN LINE
C*
LOGICAL * I LSCAN(4), ISCAN(4), KSCAN(4)
DIMENSION IJUF(765), KDUM(1), LDUM(1)
COMMON /TAPERD/ IUNIT, IFRST, FSCAN, SAMEND, SAMINC, HEADY, NSCAN,
* I LINC, IJ(200), DSL, LJIUF(30), JHEC(30), IJYTE(30), NBUF, FILENO, LINENO,
* LINC, NSAMP, NOCHAN, FORMT
EQUIVALENCE (I LSCAN, SCAN), (KSCAN(1), KSCAN(1)), (LDUM(1), LSCAN(1))
SCAN=0
WRITE (6,600) FSCAN
600 FORMAT(' SEARCHING FOR LINE',I5)
WRITE (6,610) IRPDS, NDSPR
610 FORMAT(' RECORDS PER SCAN',I5, ' SCANS PER RECORD',I5)
IRACK=-5
IF (IRPDS.GT.5) IRACK=-NRPDS
IF (FSCAN.LE.5) IRACK=-NRPDS
PSKIP=-IRACK
DO 620 I=1,PSKIP
620 BACKSPACE IUNIT
ITRY=1
5 READ (IUNIT,910,END=630) (IRUF(I),I=1,16)
910 FORMAT(1M44)
GO TO 640
630 ENDTAP = -1
RETURN 1
640 CONTINUE
KDUM(1)=IRUF(1)
LDUM(1)=IRUF(16)
IF (FORMT.EQ.1) ISCAN(3) = LSCAN(3)
IF (FORMT.EQ.1) ISCAN(4) = LSCAN(4)
IF (FORMT.EQ.2) ISCAN(3) = KSCAN(1)
IF (FORMT.EQ.2) ISCAN(4) = KSCAN(2)
WRITE (6,650) SCAN
650 FORMAT(' SCAN NO',I5,3X)
IF (SCAN.EQ.FSCAN) GO TO 10
IF (SCAN.EQ.FSCAN+NDSPR) GO TO 30
IF (ITRY.EQ.10) GO TO 20
6 CONTINUE
ITRY=ITRY+1
GO TO 5
10 WRITE (6,700) ITRY
700 FORMAT(' FOUND IT AFTER',I3,' TRIES')
DO 15 I=1,IRPDS
15 BACKSPACE IUNIT
RETURN
20 IF (ITRY.LT.2*NRPDS) GO TO 6
WRITE (6,800) ITRY
800 FORMAT(' FAILED AFTER',I5,' TRIES--ABORTING')
CALL CMERR
90 CONTINUE
WRITE (6,900) FSCAN
900 FORMAT(' SCAN',I5, ' IS MISSING--USING PREVIOUS SCAN INSTEAD')
BACKSPACE IUNIT
RETURN 2
END
SFA00010
SFA00020
SFA00030
SFA00040
SFA00050
SFA00060
SFA00070
SFA00080
SFA00090
SFA00100
SFA00110
SFA00120
SFA00130
SFA00140
SFA00150
SFA00160
SFA00170
SFA00180
SFA00190
SFA00200
SFA00210
SFA00220
SFA00230
SFA00240
SFA00250
SFA00260
SFA00270
SFA00280
SFA00290
SFA00300
SFA00310
SFA00320
SFA00330
SFA00340
SFA00350
SFA00360
SFA00370
SFA00380
SFA00390
SFA00400
SFA00410
SFA00420
SFA00430
SFA00440
SFA00450
SFA00460
SFA00470
SFA00480
SFA00490
SFA00500
SFA00510
SFA00520
SFA00530
SFA00540
SFA00550
SFA00560
SFA00570
SFA00580
```

FILE: SETMRG

SUBROUTINE SETMRG(A,B,C)  
IMPLICIT INTEGER (A-C)  
RETURN  
END

SET00010  
SET00020  
SET00030  
SET00040

ORIGINAL PAGE IS  
OF POOR QUALITY

~~19-83~~  
456





FILE: SETUP7

```

STDMAK=4.5
MAP=20
STATKY=0
ISTOP=10
NMIN=30
KRN=20
NCLASS=1
MAXCLS = 60
ICHN=0
PERCEN = .2
IPCT = 80
ISUNC=0
ISUNT=0
NSDOTS=0
DO 1 I=1,MAXPOP
1 SYMPTX(I) = SMBLS(I)
NOFEAT = 0
ISTART = 0
DO 2 I=1,19,2
2 SEQUEN(I)=SRCD
DO 3 I=2,18,2
3 SEQUEN(I)=CRCD
5 SEQUEN(20)=0
CONTINUE
IF (ITIME,NE,1) WRITE(6,HEAD)
WRITE(6,630)
C
C PUT THE NEXT CARD IN THE REREAD BUFFER
RRUNIT=30
10 READ(21,1000) (ACARD(I),I=1,20)
1000 FORMAT(20A4)
WRITE(RRUNIT,1000) (ACARD(I),I=1,20)
REWIND RRUNIT
C
C READ IN CARD
READ(30,480) CODE,CARD
REWIND RRUNIT
WRITE(6,550) CODE,CARD
COL = 0
DO 20 I=1,30
IF (CODE,EO,INVEC(I)) GO TO
* (30,50,70,80,90,100,110,130,140,150,280,160,
* 170,250,210,240,200,235,230,220,260,270,246,249,
* 256,283,284,300,340,380),I
20 CONTINUE
WRITE(6,490) CODE,CARD
GO TO 10
C
C CHANNEL CARD
30 J=NXTCHR(CARD,COL)
IF (J,EO,BLANK) GO TO 10
IF (ITIME,EO,1) GO TO 35
WRITE(6,640)
GO TO 10
35 CONTINUE
IF (J,EO,SRCD) GO TO 37
IF (J,EO,DRCD) GO TO 43
36 WRITE(6,645)
645 FORMAT(' ERROR ON CHANNEL CARD')
GO TO 10
37 M=FIND12(CARD,COL,EQUVEC)
IF (M,EO,-1) GO TO 36
NOCHAN=NUMBER(CARD,COL,CHNVC,NOCHAN)
COL = COL - 1
CALL ORDER(CHNVC,NOCHAN)
GO TO 30
43 M=FIND12(CARD,COL,EQUVEC)
IF (M,EO,-1) GO TO 36
NOFEAT = NUMBER(CARD,COL,FETVEC,NOFEAT)
COL = COL - 1
CALL ORDER(FETVEC,NOFEAT)
GO TO 30
C*
C* ISTOP CARD (MAXIMUM NUMBER OF ITERATIONS)
C*
50 J = NXTCHR(CARD,COL)
IF (J,EO,BLANK) GO TO 10

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

SET00800
SET00810
SET00820
SET00830
SET00840
SET00850
SET00860
SET00870
SET00880
SET00890
SET00900
SET00910
SET00920
SET00930
SET00940
SET00950
SET00960
SET00970
SET00980
SET00990
SET01000
SET01010
SET01020
SET01030
SET01040
SET01050
SET01060
SET01070
SET01080
SET01090
SET01100
SET01110
SET01120
SET01130
SET01140
SET01150
SET01160
SET01170
SET01180
SET01190
SET01200
SET01210
SET01220
SET01230
SET01240
SET01250
SET01260
SET01270
SET01280
SET01290
SET01300
SET01310
SET01320
SET01330
SET01340
SET01350
SET01360
SET01370
SET01380
SET01390
SET01400
SET01410
SET01420
SET01430
SET01440
SET01450
SET01460
SET01470
SET01480
SET01490
SET01500
SET01510
SET01520
SET01530
SET01540
SET01550
SET01560
SET01570
SET01580

```

FILE: SETUP7

```
COL=COL-1
J = NUMBER(CARD,COL,ISTOP,ISTART)
GO TO 10
CC*
CC*
NMIN CARD (MINIMUM NUMBER OF POINTS PER CLUSTER)
70 J = NXTCHR(CARD,COL)
IF (J.EQ.BLANK) GO TO 10
COL = COL-1
J = NUMBER(CARD,COL,NMIN,ISTART)
GO TO 10
CC*
CC*
KRN CARD (NUMBER OF ITERATIONS PER FULL OUTPUT)
80 J = NXTCHR(CARD,COL)
IF (J.EQ.BLANK) GO TO 10
COL = COL-1
J = NUMBER(CARD,COL,KRN,ISTART)
GO TO 10
CC*
CC*
STDMAX CARD (MAXIMUM STANDARD DEVIATION PER CLUSTER)
90 J = FLTNUM(CARD,COL,STDMAX,1)
GO TO 10
CC*
CC*
DLMIN CARD (MINIMUM DISTANCE BETWEEN CLUSTER MEANS)
100 J = FLTNUM(CARD,COL,DLMIN,1)
GO TO 10
CC*
CC*
SEP CARD (DISTANCE FOR SPLITTING)
110 J = FLTNUM(CARD,COL,SEP,1)
SPTRIG=1
GO TO 10
CC*
CC*
HED1 CARD
130 READ (30,500)HED1
REWIND RRUNIT
GO TO 10
CC*
CC*
HED2 CARD
140 READ (30,500)HED2
REWIND RRUNIT
GO TO 10
CC*
CC*
DATE CARD
150 READ (30,510) DATE
REWIND RRUNIT
GO TO 10
CC*
CC*
COMMENT CARD
160 READ (30,500)COMENT
REWIND RRUNIT
GO TO 10
CC*
CC*
SYMBOLS CARD
170 CONTINUE
180 ICNT=ICNT + 1
IF (ICNT.GT.MAXPOP) GO TO 10
SYMMTX(ICNT)=BLANK
190 M=NXTCHR(CARD,COL)
IF (M.EQ.BLANK) GO TO 10
IF (M.NF.COMMA) GO TO 180
SYMMTX(ICNT) = M
195 M=NXTCHR(CARD,COL)
IF (M.EQ.BLANK) GO TO 10
IF (M.NF.COMMA) GO TO 195
GO TO 180
CC*
CC*
MAXCLASS CARD (NO. CLASSES FOR THIS EXECUTION OF ISOCIS- STATISTIC
FILE WILL BE WRITTEN AFTER 'NCLASS' CLASSES HAVE
BEEN CLUSTERED)
SET01590
SET01600
SET01610
SET01620
SET01630
SET01640
SET01650
SET01660
SET01670
SET01680
SET01690
SET01700
SET01710
SET01720
SET01730
SET01740
SET01750
SET01760
SET01770
SET01780
SET01790
SET01800
SET01810
SET01820
SET01830
SET01840
SET01850
SET01860
SET01870
SET01880
SET01890
SET01900
SET01910
SET01920
SET01930
SET01940
SET01950
SET01960
SET01970
SET01980
SET01990
SET02000
SET02010
SET02020
SET02030
SET02040
SET02050
SET02060
SET02070
SET02080
SET02090
SET02100
SET02110
SET02120
SET02130
SET02140
SET02150
SET02160
SET02170
SET02180
SET02190
SET02200
SET02210
SET02220
SET02230
SET02240
SET02250
SET02260
SET02270
SET02280
SET02290
SET02300
SET02310
SET02320
SET02330
SET02340
SET02350
SET02360
SET02370
```

FILE: SETUP7

```

200 IF (ITIME.EQ.1) GO TO 205
    WRITE(6,650)
    GO TO 10
205 J=NUMBER(CARD,COL,NCLASS,ISTART)
    GO TO 10
C*
C*
C*
    MAP CARD (NUMBER OF ITERATIONS TO OUTPUT MAP)
210 J=NXTCHR(CARD,COL)
    IF (J.EQ.BLANK) GO TO 10
    COL=COL-1
    J = NUMBER(CARD,COL,MAP,ISTART)
    GO TO 10
C*
C*
C*
    PUNCH CARD (PUNCH STATISTICS ON CARDS)
215 M = FIND12(CARD,COL,CHAR)
    IF (M.NE.?) PUNCH = 1
    IF (M.NE.?) GO TO 245
    J = NUMBER(CARD,COL,PUNCH,ISTART)
    GO TO 245
C*
C*
C*
    MAP FORMAT CARD
220 M = NXTCHR(CARD,COL)
    IF (M.EQ.URCD) MAPFMT = 1
    IF (M.EQ.LRCD) MAPFMT = 2
    IF (M.EQ.BLANK) MAPFMT=1
    GO TO 10
C*
C*
C*
    CLUSTERS CARD (MAX. NO. OF CLUSTERS PER CLASS)
230 J=NXTCHR(CARD,COL)
    IF (J.EQ.BLANK) GO TO 10
    COL=COL-1
    J = NUMBER(CARD,COL,MAXCLS,ISTART)
    GO TO 10
C*
C*
    CHAIN CARD (CHAIN CLUSTERS WHICH ARE DLIN UNITS APART)
235 ICHN=1
    J=FLINUM(CARD,COL,CHNTHS,1)
    GO TO 10
C*
C*
C*
    OPTION CARD
240 J = NXTCHR(CARD,COL)
    IF (J.EQ.BLANK) GO TO 10
C*
C*
C*
    ORDER COLOR KEYS
    IF (J.EQ.ORCD) IORDER = 1
C*
C*
C*
    ERROR COMPUTATION
    IF (J.NE.ERCD) GO TO 241
    COL=COL+1
    J=NXTCHR(CARD,COL)
    IF (J.EQ.CRCD) ERCOMP=1
C
    GO TO 245
241 CONTINUE
C*
C*
C*
    PUNCH CARD
    IF (J.EQ.PRCD) GO TO 215
C*
C*
C*
    STATS
    IF (J.NE.SRCD) GO TO 242
    J=NXTCHR(CARD,COL)
    IF (J.EQ.TRCD) STATKY=1
242 CONTINUE
    CLUSTERS FOR MAPTAP
    IF (J.EQ.CPCD) MAPKEY=2
C*
C*
C*
    FIND12 A COMMA
245 J=FIND12(CARD,COL,COMVEC)
    IF (J.LE.0) GO TO 10

```

```

SET02380
SET02390
SET02400
SET02410
SET02420
SET02430
SET02440
SET02450
SET02460
SET02470
SET02480
SET02490
SET02500
SET02510
SET02520
SET02530
SET02540
SET02550
SET02560
SET02570
SET02580
SET02590
SET02600
SET02610
SET02620
SET02630
SET02640
SET02650
SET02660
SET02670
SET02680
SET02690
SET02700
SET02710
SET02720
SET02730
SET02740
SET02750
SET02760
SET02770
SET02780
SET02790
SET02800
SET02810
SET02820
SET02830
SET02840
SET02850
SET02860
SET02870
SET02880
SET02890
SET02900
SET02910
SET02920
SET02930
SET02940
SET02950
SET02960
SET02970
SET02980
SET02990
SET03000
SET03010
SET03020
SET03030
SET03040
SET03050
SET03060
SET03070
SET03080
SET03090
SET03100
SET03110
SET03120
SET03130
SET03140
SET03150
SET03160

```

10-87  
460

FILE: SETUP7

```

C
C
C
GO TO 240
SEQUENCE CARD
246 I=1
247 M=NXTCHR(CARD,COL)
IF(M.EQ.BLANK) GO TO 248
SEQUEN(I)=M
I=I+1
GO TO 247
248 NOSEQ = I - 1
GO TO 10
C
C
C
PERCENT CARD
249 J=NXTCHR(CARD,COL)
IF(J.EQ.BLANK) GO TO 10
COL=COL-1
J=NUMBER(CARD,COL,IPCT,J)
PERCEN=J.-FLOAT(IPCT)/100.
GO TO 10
C
C
C
MEANS CARD
250 J = NXTCHR(CARD,COL)
IF(J.EQ.BLANK) GO TO 10
IF(J.EQ.CRCD) GO TO 255
IF(J.NE.FBCD) GO TO 10
ISOKEY=1
GO TO 10
255 ISOKEY=1
CALL RDMEAN(ARRAY)
GO TO 10
C*
C*
C*
READ MODULE DECK AND WRITE TO INPUT STAT UNIT AND FILE.
256 SAVE1=SAVTAP
SAVE2=STAFIL
SAVAP=INUNIT
STAFIL=INFILE
CALL CRDSTA(ARRAY, TOP)
SAVTAP=SAVE1
STAFIL=SAVE2
INITM=1
GO TO 10
C
C
C
DATA FILE CARD
260 M = NXTCHR(CARD,COL)
IF(M.EQ.BLANK) GO TO 10
IF(M.EQ.URCD) GO TO 265
IF(M.EQ.FRCD) GO TO 267
263 WRITE(6,750)
750 FORMAT(' ERROR ON DATA FILE CARD')
GO TO 10
265 J = FIND12(CARD,COL,EQUVEC)
IF(J.EQ.-1) GO TO 263
M = NUMBER(CARD,COL,DATAPÉ,ZERO)
COL = COL - 1
GO TO 260
267 J = FIND12(CARD,COL,EQUVEC)
IF(J.EQ.-1) GO TO 263
M = NUMBER(CARD,COL,DATFIL,ZERO)
DATFIL = DATFIL - 1
IF(DATFIL.LT.0) DATFIL = 0
COL = COL - 1
GO TO 260
C
C
C
STAT FILE CARD
270 M=NXTCHR(CARD,COL)
271 IF(M.EQ.URCD) GO TO 278
IF(M.EQ.FRCD) GO TO 275
IF(M.EQ.ORCI) GO TO 272
IF(M.EQ.COMMA) GO TO 270
IF(M.EQ.BLANK) GO TO 10
273 WRITE(6,755)
755 FORMAT(' ERROR ON STATFILE CARD')

```

SET03170  
 SET03180  
 SET03190  
 SET03200  
 SET03210  
 SET03220  
 SET03230  
 SET03240  
 SET03250  
 SET03260  
 SET03270  
 SET03280  
 SET03290  
 SET03300  
 SET03310  
 SET03320  
 SET03330  
 SET03340  
 SET03350  
 SET03360  
 SET03370  
 SET03380  
 SET03390  
 SET03400  
 SET03410  
 SET03420  
 SET03430  
 SET03440  
 SET03450  
 SET03460  
 SET03470  
 SET03480  
 SET03490  
 SET03500  
 SET03510  
 SET03520  
 SET03530  
 SET03540  
 SET03550  
 SET03560  
 SET03570  
 SET03580  
 SET03590  
 SET03600  
 SET03610  
 SET03620  
 SET03630  
 SET03640  
 SET03650  
 SET03660  
 SET03670  
 SET03680  
 SET03690  
 SET03700  
 SET03710  
 SET03720  
 SET03730  
 SET03740  
 SET03750  
 SET03760  
 SET03770  
 SET03780  
 SET03790  
 SET03800  
 SET03810  
 SET03820  
 SET03830  
 SET03840  
 SET03850  
 SET03860  
 SET03870  
 SET03880  
 SET03890  
 SET03900  
 SET03910  
 SET03920  
 SET03930  
 SET03940  
 SET03950

FILE: SETUP7

272	GO TO 10	SET03960
	J=FIND12(CARD,COL,SLASH)	SET03970
	IF(J.EQ.-1)GO TO 273	SET03980
274	M=NXTCHR(CARD,COL)	SET03990
	IF(M.EQ.COMMA) GO TO 274	SET04000
	IF(M.EQ.FHCD) GO TO 277	SET04010
	IF(M.NE.URCD) GO TO 271	SET04020
275	J=FIND12(CARD,COL,EQUVEC)	SET04030
	IF(J.EQ.-1)GO TO 273	SET04040
	M=NUMBER(CARD,COL,SAVTAP,ZERO)	SET04050
	COL=COL-1	SET04060
	GO TO 274	SET04070
277	J=FIND12(CARD,COL,EQUVEC)	SET04080
	IF(J.EQ.-1)GO TO 273	SET04090
	M=NUMBER(CARD,COL,STAFIL,ZERO)	SET04100
	COL=COL-1	SET04110
	STAFIL=STAFIL-1	SET04120
	IF(STAFIL.LT.0)STAFIL=0	SET04130
	GO TO 274	SET04140
278	J=FIND12(CARD,COL,SLASH)	SET04150
	IF(J.EQ.-1)GO TO 273	SET04160
	INITM = 1	SET04170
279	M=NXTCHR(CARD,COL)	SET04180
	IF(M.EQ.COMMA) GO TO 279	SET04190
	IF(M.EQ.FHCD) GO TO 282	SET04200
	IF(M.NE.URCD) GO TO 271	SET04210
281	J=FIND12(CARD,COL,EQUVEC)	SET04220
	IF(J.EQ.-1)GO TO 273	SET04230
	M=NUMBER(CARD,COL,INUNIT,ZERO)	SET04240
	COL=COL-1	SET04250
	GO TO 279	SET04260
282	J=FIND12(CARD,COL,EQUVEC)	SET04270
	IF(J.EQ.-1)GO TO 273	SET04280
	M=NUMBER(CARD,COL,INFILE,ZERO)	SET04290
	COL=COL-1	SET04300
	GO TO 279	SET04310
C*		SET04320
C*	SUBCLASSES CARD--USE THE MEANS FOR THESE SUBCLASSES FROM THE	SET04330
C*	STAT FILE FOR INITIAL MEANS	SET04340
C*		SET04350
283	NOSUB2=NUMBER(CARD,COL,SUBVEC,NOSUB2)	SET04360
	GO TO 10	SET04370
C*		SET04380
C*	MINIMUM POPULATION FOR STATISTICS PASS.	SET04390
C*		SET04400
284	J=NXTCHR(CARD,COL)	SET04410
	IF(J.NE.MINUS)COL=COL-1	SET04420
	M=NUMBER(CARD,COL,PMIN,ZERO)	SET04430
	IF(J.EQ.MINUS)PMIN=0-PMIN	SET04440
	GO TO 10	SET04450
C	DOTFIL INPUT/UNIT=N,FILE=M	SET04460
	OR UNIT=N,FILE=M	SET04470
300	J=NXTCHR(CARD,COL)	SET04480
301	IF(J.EQ.BLANK) GO TO 320	SET04490
	IF(J.NE.URCD) GO TO 305	SET04500
	J=FIND12(CARD,COL,EQUVEC)	SET04510
	IF(J.NE.?) GO TO 320	SET04520
	ISTART=0	SET04530
	J=NUMBER(CARD,COL,ARRAY(TOP-30),ISTART)	SET04540
	DOTUNT = ARRAY(TOP - 30)	SET04550
	J=FIND12(CARD,COL,EQUVEC)	SET04560
	IF(J.NE.?) GO TO 320	SET04570
	ISTART=0	SET04580
	J=NUMBER(CARD,COL,ARRAY(TOP-30),ISTART)	SET04590
	DOTFIL=ARRAY(TOP-30)	SET04600
	DOTFIL = DOTFIL - 1	SET04610
	GO TO 10	SET04620
305	IF(J.NE.IPCD) GO TO 310	SET04630
	J=FIND12(CARD,COL,SLASH)	SET04640
	IF(J.NE.?) GO TO 320	SET04650
	J = NXTCHR(CARD,COL)	SET04660
	IF(J.EQ.FHCD) GO TO 315	SET04670
	IF(J.EQ.UHCD) GO TO 301	SET04680
	GO TO 320	SET04690
310	IF(J.NE.FHCD) GO TO 320	SET04700
315	J = FIND12(CARD,COL,EQUVEC)	SET04710
	IF(J.NE.?) GO TO 320	SET04720
	ISTART = 0	SET04730
	J = NUMBER(CARD,COL,ARRAY(TOP - 30),ISTART)	SET04740

FILE: SETUP7

```
DOTFIL = ARRAY(TOP - 30) - 1
J = FIND12(CARD,COL,EQUVEC)
IF (J.NE.2) GO TO 320
ISTART = 0
J = NUMBER(CARD,COL,ARRAY(TOP - 30),ISTART)
DOTUNT = ARRAY(TOP - 30)
GO TO 10
320 WRITE(6,760)
760 FORMAT(' ERROR ON DOTFILE CARD')
GO TO 10
C
SUNANG
340 J=NATCHR(CARD,COL)
IF (J.NE.TRCD) GO TO 345
ISUNT=1
GO TO 10
345 ISTART=0
COL = COL - 1
J=NUMBER(CARD,COL,ARRAY(TOP-30),ISTART)
IF (J.GT.8) J = 8
IF (J.EQ.0) GO TO 10
ISUNC=0
DO 350 JJ=1,J
ISUNC=ISUNC+1
350 SUNANG(JJ)=ARRAY(TOP-31+ISUNC)
GO TO 10
C
DOTS
380 ISTART=0
J = NUMBER(CARD,COL,STDOTS(NSDOTS + 1),ISTART)
NSDOTS = NSDOTS + J
IF (NSDOTS.GT.60) NSDOTS=60
GO TO 10
C
C
*END* CARD
280 CONTINUE
IF (NOFEAT .GT. 0) GO TO 285
NOFEAT=30
DO 261 I=1,30
261 FETVEC(I)=I
285 CONTINUE
C
C
PRINT USER REQUEST
WRITE(6,660)
WRITE(6,670)ISTOP,NMIN,KRN,MAP,MAXCLS,NCLASS,
* (FETVEC(I),I=1,NOFEAT)
WRITE(6,680)DLMIN,STDMAX
WRITE(6,685)IPCT,PMIN,NSDOTS,ISUNC,ISUNT
IF (SPTRIG.EQ.1)WRITE(6,690)SEP
IF (ICHN.EQ.1) WRITE(6,710)CHNTHS
IF (PUNCH.EQ.1) WRITE(6,700)
IF (IORDER.EQ.1)WRITE(6,715)
IF (MAPFMT.EQ.1)WRITE(6,720)
IF (MAPFMT.EQ.2)WRITE(6,725)
IF (NOFEAT .GT. NMIN) WRITE(6,740)
RETURN
C
480 FORMAT(A4,6X,62A1)
490 FORMAT(' INVALID INPUT CARD--IGNORED'/T5,A4,6X,62A1)
500 FORMAT(10X,15A4)
510 FORMAT(10X,3A4)
550 FORMAT(7X,A4,4X,62A1)
630 FORMAT('///' INPUT SUMMARY'///)
640 FORMAT(' CHANNELS CANNOT BE CHANGED UNTIL THIS EXECUTION OF ISOCLS
* IS COMPLETED')
650 FORMAT(' NO. OF CLASSES CANNOT BE CHANGED UNTIL THIS EXECUTION OF
* ISOCLS IS COMPLETED')
660 FORMAT('///' YOU HAVE SELECTED THE FOLLOWING PARAMETER VALUES AND
* OPTIONS'///)
670 FORMAT(' STOP AFTER',I5,' ITERATION(S)' /
* ' ALLOW A MINIMUM OF',I5,' PIXELS PER CLUSTER' /
* ' PRINT A CLUSTER SUMMARY EVERY',I5,' ITERATION(S)' /
* ' PRINT A CLUSTER MAP EVERY',I5,' ITERATION(S)' /
* ' ALLOW A MAXIMUM OF',I5,' CLUSTERS PER CLASS' /
* ' THE STATISTICS FILE WILL BE WRITTEN AFTER',I4,' CLASS(ES)
* HAVE BEEN CLUSTERED' /
* ' CHANNELS ARE ---',I3)
680 FORMAT(' DLMIN =',F7.3 / ' STDMAX=',F7.3)
685 FORMAT(' PERCENT =',I5 / ' PMIN =',I5 / ' NSDOTS =',I5 /
```

SET04750  
SET04760  
SET04770  
SET04780  
SET04790  
SET04800  
SET04810  
SET04820  
SET04830  
SET04840  
SET04850  
SET04860  
SET04870  
SET04880  
SET04890  
SET04900  
SET04910  
SET04920  
SET04930  
SET04940  
SET04950  
SET04960  
SET04970  
SET04980  
SET04990  
SET05000  
SET05010  
SET05020  
SET05030  
SET05040  
SET05050  
SET05060  
SET05070  
SET05080  
SET05090  
SET05100  
SET05110  
SET05120  
SET05130  
SET05140  
SET05150  
SET05160  
SET05170  
SET05180  
SET05190  
SET05200  
SET05210  
SET05220  
SET05230  
SET05240  
SET05250  
SET05260  
SET05270  
SET05280  
SET05290  
SET05300  
SET05310  
SET05320  
SET05330  
SET05340  
SET05350  
SET05360  
SET05370  
SET05380  
SET05390  
SET05400  
SET05410  
SET05420  
SET05430  
SET05440  
SET05450  
SET05460  
SET05470  
SET05480  
SET05490  
SET05500  
SET05510  
SET05520  
SET05530

FILE: SETUP7

```

* NO. SUN ANGLES FROM CARDS = .15 / SUN ANGLE TAPE SW = .15/
690 FORMAT(10 SEP = . F7.3)
700 FORMAT(10 PUNCH THE MODULE STAT DECK)
710 FORMAT(10 CHAIN CLUSTERS WHICH ARE . F7.3. UNITS APART)
715 FORMAT(1X, 'ORDER COLOR KEYS')
720 FORMAT(1X, 'WRITE A CLUSTER MAP OUTPUT TAPE IN UNIVERSAL FORMAT')
725 FORMAT(1X, 'WRITE A CLUSTER MAP OUTPUT TAPE IN LARSYS II FORMAT')
740 FORMAT(//, '**WARNING** NMIN IS LESS THAN NO. OF CHANNELS. COVARIAN
* CFS WILL NOT BE INVERTIBLE')
END

```

SET05640  
SET05650  
SET05660  
SET05670  
SET05680  
SET05690  
SET05700  
SET05710  
SET05720  
SET05730  
SET05740  
SET05750  
SET05760  
SET05770  
SET05780  
SET05790  
SET05800  
SET05810  
SET05820  
SET05830  
SET05840  
SET05850  
SET05860  
SET05870  
SET05880  
SET05890  
SET05900  
SET05910  
SET05920  
SET05930  
SET05940  
SET05950  
SET05960  
SET05970  
SET05980  
SET05990  
SET06000

19-91  
464

FILE: SUNFAC

```

SUBROUTINE SUNFAC(SUNCOR,SUNANG,FETVEC,NOFEAT,ISUNC,ISUNT)
INTEGER SUNANG,FETVEC,SUNA
EQUIVALENCE (EXTRA(1),DUM1(1)),(EXTRA(109),DUM2(1)),
*(EXTRA(217),DUM3(1))
DIMENSION EXTRA(324),SUNANG(1),FETVEC(1),SUNCOR(1),
DUM1(108),DUM2(108),DUM3(108)
C
INCLUDE COMMK6
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,RMKEY,
HISFIL,HISKEY,TRFORM,EHIPTP,EPPKEY,MAPUNT,NOFILE,
DRUMAD,DRM,OS,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVL,
NHSTUN,NHSTFI,SCTRUN,MAPFIL,
DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
CRDUNT,PRUNT,RANDIO
CSEND
C
DATA DUM1 /16.413,14.887,14.089,13.401,
*13.015,11.901,11.323,10.816,10.755,9.893,9.449,9.052,
*9.094,8.413,8.063,7.745,7.832,7.285,7.002,6.744,
*6.880,6.425,6.190,5.973,6.098,5.722,5.527,5.346,
*5.474,5.156,4.991,4.838,4.967,4.694,4.551,4.419,
*4.530,4.297,4.174,4.061,
*4.163,3.961,3.855,3.757,3.852,3.675,3.583,3.497,
*3.580,3.426,3.344,3.269,3.342,3.207,3.135,3.069,
*3.135,3.015,2.952,2.893,2.953,2.846,2.789,2.737,
*2.788,2.693,2.643,2.597,2.642,2.557,2.513,2.471,
*2.511,2.435,2.395,2.358,
*2.393,2.325,2.289,2.255,2.285,2.223,2.191,2.161,
*2.187,2.131,2.102,2.076,2.097,2.047,2.021,1.997,
*2.016,1.971,1.947,1.925,1.940,1.878,1.858,
*1.871,1.834,1.815,1.797,1.807,1.773,1.756,1.740/
DATA DUM2 /1.748,1.717,1.701,1.687,
*1.693,1.665,1.651,1.637,1.642,1.617,1.603,1.591,
*1.594,1.571,1.559,1.548,1.550,1.529,1.518,1.508,
*1.508,1.489,1.478,1.470,1.469,1.452,1.443,1.435,
*1.432,1.417,1.409,1.402,1.398,1.384,1.377,1.370,
*1.366,1.353,1.347,1.341,
*1.336,1.324,1.318,1.313,1.307,1.297,1.291,1.286,
*1.280,1.271,1.266,1.261,1.255,1.246,1.242,1.238,
*1.230,1.223,1.219,1.216,1.208,1.201,1.198,1.195,
*1.186,1.180,1.177,1.174,1.166,1.161,1.158,1.155,
*1.146,1.142,1.140,1.137,
*1.128,1.124,1.122,1.120,1.111,1.107,1.106,1.104,
*1.094,1.091,1.090,1.089,1.078,1.076,1.075,1.074,
*1.064,1.062,1.061,1.060,1.050,1.048,1.047,1.047,
*1.036,1.035,1.035,1.034,1.023,1.023,1.022,1.022/
DATA DUM3 /1.011,1.011,1.011,1.011,
*1.000,1.000,1.000,1.000,0.989,0.989,0.990,0.990,
*.979,0.979,0.980,0.980,0.969,0.970,0.971,
*.960,0.961,0.961,0.962,0.951,0.952,0.953,0.954,
*.943,0.944,0.945,0.946,0.935,0.937,0.937,0.938,
*.929,0.929,0.930,0.931,
*.921,0.923,0.924,0.925,0.914,0.916,0.917,0.918,
*.908,0.910,0.911,0.912,0.902,0.904,0.906,0.907,
*.897,0.899,0.901,0.902,0.892,0.894,0.896,0.897,
*.887,0.890,0.891,0.892,0.882,0.885,0.887,0.888,
*.878,0.881,0.883,0.884,
*.875,0.878,0.879,0.881,0.871,0.874,0.876,0.878,
*.868,0.871,0.873,0.875,0.865,0.869,0.870,0.872,
*.863,0.866,0.868,0.870,0.861,0.864,0.866,0.868,
*.850,0.862,0.864,0.866,0.857,0.861,0.863,0.864/
DATA PLANK /1/
KS = 0
DO 200 I=1,NOFEAT
K=(FETVEC(I)-1)/NCHPAS
IF (ISUNC.NE.0.AND.I.EQ.1) KS=K
KR=FETVEC(I)-K*NCHPAS
K=K+1
SUNA = SUNANG(K-KS)
100 JND=(SUNA-5)*NCHPAS+KR
SUNCOR(I)=EXTRA(JND)
200 CONTINUE
WRITE(6,90)
90 FORMAT(/T61,'SUN ANGLES')
WRITE(6,210) (SUNANG(I), I = 1,8)
210 FORMAT(T45,A15)
WRITE(6,215)
215 FORMAT(/T52,'CORRECTIONS FOR SUN ANGLES')
C
NOFETR = NOFEAT

```

SUN00010  
SUN00020  
SUN00030  
SUN00040  
SUN00050  
SUN00060  
SUN00070  
SUN00080  
SUN00090  
SUN00100  
SUN00110  
SUN00120  
SUN00130  
SUN00140  
SUN00150  
SUN00160  
SUN00170  
SUN00180  
SUN00190  
SUN00200  
SUN00210  
SUN00220  
SUN00230  
SUN00240  
SUN00250  
SUN00260  
SUN00270  
SUN00280  
SUN00290  
SUN00300  
SUN00310  
SUN00320  
SUN00330  
SUN00340  
SUN00350  
SUN00360  
SUN00370  
SUN00380  
SUN00390  
SUN00400  
SUN00410  
SUN00420  
SUN00430  
SUN00440  
SUN00450  
SUN00460  
SUN00470  
SUN00480  
SUN00490  
SUN00500  
SUN00510  
SUN00520  
SUN00530  
SUN00540  
SUN00550  
SUN00560  
SUN00570  
SUN00580  
SUN00590  
SUN00600  
SUN00610  
SUN00620  
SUN00630  
SUN00640  
SUN00650  
SUN00660  
SUN00670  
SUN00680  
SUN00690  
SUN00700  
SUN00710  
SUN00720  
SUN00730  
SUN00740  
SUN00750  
SUN00760  
SUN00770  
SUN00780  
SUN00790



FILE: SUNFAC

```
      ISTART = 1
      IEND = 16
217  IF (IEND .GE. NOFETR) IEND = NOFETR
      IENDS = ISTART + IEND - 1
      WRITE (6,220) (HLANK,FETVEC(I),I=ISTART,IENDS)
220  FORMAT (/1X,16(A1,'CH(':12.1)'.1X))
      WRITE (6,230) (SUNCOR(I),I=ISTART,IENDS)
230  FORMAT (16(2X,F6.4))
C
      NOFETR = NOFFTR - IEND
      ISTART = IENDS + 1
      IF (NOFETR .LE. 0) RETURN
      GO TO 217
      END
```

SUN00800  
SUN00810  
SUN00820  
SUN00830  
SUN00840  
SUN00850  
SUN00860  
SUN00870  
SUN00880  
SUN00890  
SUN00900  
SUN00910  
SUN00920  
SUN00930

~~19-93~~  
266

FILE TAPHDR

```

SUBROUTINE TAPHDR(DATAPE,IFILE)
IMPLICIT INTEGER(A-Z)
.....
TAPERD READS THE MULTISPECTRAL SCANNER DATA TAPE, UNPACKS THE
REQUESTED DATA AND RETURNS IT UNPACKED TO THE CALLING ROUTINE.
THERE ARE THREE SEPARATE SUBROUTINES: TAPHDR, FLDINT
AND LINERD, NEEDED TO READ A TAPE
TAPHDR MUST BE CALLED ONCE TO READ THE HEADER RECORD AND UNPACK
NECESSARY DATA FROM THE RECORD
CALL TAPHDR(DATAPE,IFILE)
DATAPE-INPUT UNIT NUMBER FOR DATA TAPE
IFILE - NO. OF E-O-F'S IN DATA TAPE USER WISHES TO READ OVER
IN ORDER TO POSITION TAPE TO DESIRED FILE
FLDINT MUST BE CALLED ONCE FOR EACH FIELD. THE TAPE IS POSITIONED
TO THE CORRECT RECORD AND PARAMETERS ARE INITIALIZED FOR THE FIELD
CALL FLDINT(BLOCK,FETVEC,NOFEAT)
BLOCK(1)=LINE START
BLOCK(2)=LINE END
BLOCK(3)=LINE INCREMENT
BLOCK(4)=SAMPLE START
BLOCK(5)=SAMPLE END
CONTINUE
BLOCK(6)=SAMPLE INCREMENT
FETVEC- (INPUT) VECTOR CONTAINING FEATURES REQUESTED
NOFEAT (INPUT) NO. OF FEATURES IN FETVEC
CALL LINERD(IDATA,ENDTAP)
IDATA-(OUTPUT) ARRAY CONTAINING UNPACKED DATA
ENDTAP - TRIGGER INDICATING WHETHER OR NOT AN E-O-F HAS BEEN
REACHED WHILE TRYING TO READ A GIVEN SCAN LINE NO.
IF AN E-O-F IS FOUND ENDTAP = -1, OTHERWISE, ENDTAP
= 0.
.....
READY IS A INDICATOR TO TEST WHETHER THE TAPES HAS BEEN
POSITIONED AND PARAMETERS SET FOR A FIELD
THE APRAYS NR AND HWRD ARE PRECALCULATED WORD AND BIT
POSITIONS OF INFORMATION IN THE HEADER RECORD OF THE UNIVERSAL
FORMAT WHICH MUST BE EXTRACTED.
CONTINUE
NRPDS - NO. OF RECORDS PER DATA SET
NCPV - NO. OF CHANNELS PER RECORD ON RECORDS PAST ANCILLARY RECORD
NPRC - NO. OF PHYSICAL RECORDS PER CHANNEL
ANCLNG - ANCILLARY LENGTH IN BYTES
NC - NO. OF CHANNELS
NS - NO. OF SAMPLES PER CHANNEL PER SCAN
NHITS - NO. OF BITS PER PIXEL
DOI - DATA ORDER INDICATOR
NDSPH - NO. OF DATA SETS PER RECORD
NCAR - NO. OF CHANNELS OF VIDEO DATA ON SAME RECORD
WITH ANCILLARY DATA)
SVD - START OF VIDEO DATA. (BYTE POSITION WITHIN DATA FOR
A GIVEN CHANNEL)
LOGICAL*1 IINH(30600)
LOGICAL*1 IROW(4), ILINE(4)
DIMENSION FNM(3,4)
DIMENSION NA(28), HWRD(28)
COMMON /TAPDR/ IUNIT,IFPST,FSCAN,SAMEND,SAMINC,READY,NSCAN,
* LINC,ID(200),NSL,LUHF(30),JREC(30),IHYTE(30),NBUFS,FILENO,LINEND
* LININC,NSAMP,NCHAN,FORMT
COMMON /IDSTR/IDD(250)
COMMON /ISOLR/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP
EQUIVALENCE (I,OWD,WORD)
EQUIVALENCE (IFPST,ILINE)
EQUIVALENCE (ID(1),NRPDS ),(ID(2),NCPV ),
,(ID(3),NPRC ),(ID(4),ANCLNG),
,(ID(5),NC ),(ID(6),NS )

```

TAP00010  
TAP00020  
TAP00030  
TAP00040  
TAP00050  
TAP00060  
TAP00070  
TAP00080  
TAP00090  
TAP00100  
TAP00110  
TAP00120  
TAP00130  
TAP00140  
TAP00150  
TAP00160  
TAP00170  
TAP00180  
TAP00190  
TAP00200  
TAP00210  
TAP00220  
TAP00230  
TAP00240  
TAP00250  
TAP00260  
TAP00270  
TAP00280  
TAP00290  
TAP00300  
TAP00310  
TAP00320  
TAP00330  
TAP00340  
TAP00350  
TAP00360  
TAP00370  
TAP00380  
TAP00390  
TAP00400  
TAP00410  
TAP00420  
TAP00430  
TAP00440  
TAP00450  
TAP00460  
TAP00470  
TAP00480  
TAP00490  
TAP00500  
TAP00510  
TAP00520  
TAP00530  
TAP00540  
TAP00550  
TAP00560  
TAP00570  
TAP00580  
TAP00590  
TAP00600  
TAP00610  
TAP00620  
TAP00630  
TAP00640  
TAP00650  
TAP00660  
TAP00670  
TAP00680  
TAP00690  
TAP00700  
TAP00710  
TAP00720  
TAP00730  
TAP00740  
TAP00750  
TAP00760

FILE TAPHDR

```

      (ID(7),NMITS ),(ID(8),DOI),
      (ID(9),NDSPM),(ID(10),NCAR ),
      (ID(11),SVD),(ID(16),PRSZ)
* DATA FRM/UNIV,'ERSA',L',LARS',YS 2,
      'LAND',SAT 1,1/2',LAND',SAT 1,3
* DATA HWRD/104,102,103,105,90,1747,91,107,1774,1745,92,104,110,
      1749,1791,100,2201,2203,2205,2207,2209,2211,2213,2215,
* 61,62,63,67/
* DATA NR/8,A,16,A,16,3,8,A,16,16,16,16,16,16,
* 16,16,16,16,16,16,16,8,8,8,16/
TAP00770
TAP00780
TAP00790
TAP00800
TAP00810
TAP00820
TAP00830
TAP00840
TAP00850
TAP00860
TAP00870
TAP00880
TAP00890
TAP00900
TAP00910
TAP00920
TAP00930
TAP00940
TAP00950
TAP00960
TAP00970
TAP00980
TAP00990
TAP01000
TAP01010
TAP01020
TAP01030
TAP01040
TAP01050
TAP01060
TAP01070
TAP01080
TAP01090
TAP01100
TAP01110
TAP01120
TAP01130
TAP01140
TAP01150
TAP01160
TAP01170
TAP01180
TAP01190
TAP01200
TAP01210
TAP01220
TAP01230
TAP01240
TAP01250
TAP01260
TAP01270
TAP01280
TAP01290
TAP01300
TAP01310
TAP01320
TAP01330
TAP01340
TAP01350
TAP01360
TAP01370
TAP01380
TAP01390
TAP01400
TAP01410
TAP01420
TAP01430
TAP01440
TAP01450
TAP01460
TAP01470
TAP01480
TAP01490
TAP01500
TAP01510
TAP01520

      ENTRY FOR READING HEADER INFORMATION

      INFORMATION IN ERCDIC OR IIR FLOATING POINT IS NOT UNPACKED
      FROM THE HEADER RECORD AT THIS TIME.

      READY = -1
      IFRST = 0
      FILENO = IFILE
      IUNIT=DATAPE
      REWIND IUNIT

      SKIP DIRECTORY FILE FOR LANDSAT 3

      IF (FORMAT.NE.4) GO TO 2
      READ(IUNIT,510,END=4) DUMMY
      GO TO 3
      FILENO=FILENO*3
      REC = 0
      KHUF = 3060
      IF (ISUNC.NE.0) GO TO 6
      DO 5 I = 1,9
      SUNANG(I) = 60
      CONTINUE

      NRPOS=1
      IF ( FILENO .EQ. 0) GO TO 600
      DO A I=1,FILENO
      IF (FORMAT.NE.4) WRITE(6,610) I
610   IF (FORMAT.EQ. SKIPPING FILE'.IB)
      IF (FORMAT.NE.4) GO TO 7
      IJ=I/3
      IF (MOD(I,3).EQ.0) WRITE(6,610) IJ
7   READ(IUNIT,510,END=8) DUMMY
      GO TO 7
      CONTINUE
600 IF (FORMAT.EQ.2) GO TO 11
      IF (FORMAT.EQ.3) GO TO 1000
      IF (FORMAT.EQ.4) GO TO 1000
9   CALL HUFILL(WEC,IUNIT,KHUF,IHUF,NRPOS,ENDTAP,IERR)
      IF (IERR.EQ.-1) GO TO 11
      FORMAT = 1
      GO TO 20
11  KHUF = 400
      REC = 0
      CALL HUFILL(WEC,IUNIT,KHUF,LD,NRPOS,ENDTAP,IERR)
      IF ( IERR .EQ. -1) GO TO 10
      FORMAT = 2
      GO TO 20
10  WRITE (6,280)
      WRITE (6,340)
      CALL CMERR
20  IF (FORMAT.EQ.'1) GO TO 40
      IF (FORMAT.NE. 2) GO TO 265
      DO 12 I=4,200
12  ID(I)=ID(I)
      NCS=NC*NS
      PRSZ = NCS * 4
      IF (PRSZ.LE. 30600) GO TO 35
      WRITE (6,350) NC*NS
      CALL CMERR
35  CONTINUE
      NRPOS=1
      MAXREC = PRSZ

```

FILE TAPHDR

```

NCAR=NC
ANCLNG=4
SVD=1
NBITS=8
DOI=0
NCPR=0
NDSPR=1
NPRC = 0
WRITE(6,481) (FRM(I,2),I=1,3),NC,NS
SMSTR=1
GO TO 100
C*
C* UNPACK NECESSARY INFORMATION FROM HEADER RECORD-UNIVERSAL FORMAT
C*
40 ILIM = 156
DO 60 I = 1,ILIM
IWD = 112 * (I - 29) * 4
IF (I.LT.29) IWD = MWORD(I)
WORD = 0
NBYTES = 4
IF (I.LT.29) NBYTES = NB(I)/8
DO 55 J=1,NBYTES
LOC = 4 * J - NBYTES
IPDS=IWD*J-1
55 IWORD(LOC) = IBUF(IWD*J-1)
ID(I) = WORD
60 CONTINUE
SMSTR = ID(12)
SMSTP = ID(13)
SMINC = ID(14)
LINSKP = ID(15)
IF (ISUNT.EQ.0) GO TO 65
DO 62 I=1,8
IF (ID(16+I).LT.5.OR.ID(16+I).GT.85) ID(16+I) = 60
62 SUNANG(I) = ID(16+I)
65 CONTINUE
WORD = 0
DO 66 I=1,4
IPAT = 2254 * (I - 1) * 8
IWORD(4) = IBUF(IPAT)
ID(159 + I) = WORD
66
C***
C*** CODE JUST ABOVE ADDED OCT.20,1978 TO UNPACK SOIL LINES
C***
IF (ISUNT.GT.0) WRITE(6,482) (SUNANG(I), I=1,8)
482 FORMAT(14// 'SUN ANGLES : ',8I6)
WRITE(6,481) (FRM(I,1),I=1,3),NC,NS
MAXREC = PRS2
70 IF (NPRC.LF.1) GO TO 80
WRITE(6,360)
CALL CMERR
80 CONTINUE
IF (SVD.LE.0) SVD=1
IF (NDSPR.LE.0) NDSPR=1
IF (NBITS.EQ.8) GO TO 90
WRITE(6,390) NBITS
NBITS=8
90 IF (DOI.EQ.0) GO TO 100
WRITE(6,400) DOI
CALL CMERR
100 CONTINUE
KPTS=0
IPD=0
C*
C* DATA SET LENGTH IN BYTES
DSL=ANCLNG*NS*NC
C*
C* READ FIRST DATA SET TO DETERMINE FIRST SCAN LINE NUMBER
C*
IBUF=1
REC=0
CALL BUFILL(PEC, IUNIT, MAXREC, IBUF, NRPDS, ENDTAP, IERR)
C*
C* IFIRST = ILINE(1-4)
IF (FORMT .EQ. 1) ILINE(3) = IBUF(71)

```

TAP01530  
TAP01540  
TAP01550  
TAP01560  
TAP01570  
TAP01580  
TAP01590  
TAP01600  
TAP01610  
TAP01620  
TAP01630  
TAP01640  
TAP01650  
TAP01660  
TAP01670  
TAP01680  
TAP01690  
TAP01700  
TAP01710  
TAP01720  
TAP01730  
TAP01740  
TAP01750  
TAP01760  
TAP01770  
TAP01780  
TAP01790  
TAP01800  
TAP01810  
TAP01820  
TAP01830  
TAP01840  
TAP01850  
TAP01860  
TAP01870  
TAP01880  
TAP01890  
TAP01900  
TAP01910  
TAP01920  
TAP01930  
TAP01940  
TAP01950  
TAP01960  
TAP01970  
TAP01980  
TAP01990  
TAP02000  
TAP02010  
TAP02020  
TAP02030  
TAP02040  
TAP02050  
TAP02060  
TAP02070  
TAP02080  
TAP02090  
TAP02100  
TAP02110  
TAP02120  
TAP02130  
TAP02140  
TAP02150  
TAP02160  
TAP02170  
TAP02180  
TAP02190  
TAP02200  
TAP02210  
TAP02220  
TAP02230  
TAP02240  
TAP02250  
TAP02260  
TAP02270  
TAP02280

FILE TAPHDR

```

IF (FORMT .EQ. 1) ILINE(4) = IBUF(72)
IF (FORMT .EQ. 2) ILINE(3) = IBUF(1)
IF (FORMT .EQ. 2) ILINE(4) = IBUF(2)
IF (IFRST.GT.0) GO TO 120
WRITE (6,300)
WRITE (6,340)
CALL CMERR
120 FSCAN=IFRST
WRITE (6,500) IFRST, SMSTR
RETURN
265 WRITE (6,340)
CALL CMERR
RETURN

C
C
C
1000 SET UP FOR LANDSAT 1 OR 2 FORMAT
      KBUF=40
      REC=0
      CALL BUFill (REC, IUNIT, KBUF, IBUF, NRPDS, ENDTAP, IERR)
      IF (IERR.EQ.-1) GO TO 10

C
C
C
      UNPACK DATA FROM LANDSAT 1 OR 2 HEADER

      NRITS=8
      DOI=2
      NRPDS=1
      NCPR=4
      NPRC=1
      ANCLNG=0
      NC=4
      WORD=0
      IWORD(3)=IBUF(39)
      IWORD(4)=IBUF(40)
      NS=WORD/4
      NDSPR=1
      NCAR=4
      SVD=1
      IWORD(3)=IBUF(17)
      IWORD(4)=IBUF(18)
      PWSZ=WORD
      DSL=NS*NC
      FSCAN=1
      IFRST=1
      SMSTR=1
      WRITE (6,481) (FRM(I,3), I=1, NC, NS)
      WRITE (6,500) IFRST, SMSTR
      RETURN

C
C
C
2000 SET UP FOR LANDSAT 3
      KBUF=3596
      REC=0
      CALL BUFill (REC, IUNIT, KBUF, IBUF, NRPDS, ENDTAP, IERR)
      IF (IERR.EQ.-1) GO TO 10
      NRITS=8
      DOI=0
      WORD=0
      IWORD(4)=IBUF(120)

C
C
C
      TYPE INDICATOR 0=SEQUENTIAL 1=INTERLEAVED

      TYPE=0
      IF (WORD.NE.0) TYPE =1
      IF (TYPE.EQ.0) NRPDS=1
      NCPR=1
      NPRC=1
      ANCLNG=0
      IF (TYPE.EQ.0) NC=1
      IF (TYPE.EQ.0) GO TO 2200

C
C
C
      SET NC AND NRPDS FOR INTERLEAVED FORMAT

      WORD=0
      IWORD(4)=IBUF(46)
      IF (WORD.EQ.3) NRPDS=5
      IF (WORD.NE.3) NRPDS=4

```

TAP02290  
TAP02300  
TAP02310  
TAP02320  
TAP02330  
TAP02340  
TAP02350  
TAP02360  
TAP02370  
TAP02380  
TAP02390  
TAP02400  
TAP02410  
TAP02420  
TAP02430  
TAP02440  
TAP02450  
TAP02460  
TAP02470  
TAP02480  
TAP02490  
TAP02500  
TAP02510  
TAP02520  
TAP02530  
TAP02540  
TAP02550  
TAP02560  
TAP02570  
TAP02580  
TAP02590  
TAP02600  
TAP02610  
TAP02620  
TAP02630  
TAP02640  
TAP02650  
TAP02660  
TAP02670  
TAP02680  
TAP02690  
TAP02700  
TAP02710  
TAP02720  
TAP02730  
TAP02740  
TAP02750  
TAP02760  
TAP02770  
TAP02780  
TAP02790  
TAP02800  
TAP02810  
TAP02820  
TAP02830  
TAP02840  
TAP02850  
TAP02860  
TAP02870  
TAP02880  
TAP02890  
TAP02900  
TAP02910  
TAP02920  
TAP02930  
TAP02940  
TAP02950  
TAP02960  
TAP02970  
TAP02980  
TAP02990  
TAP03000  
TAP03010  
TAP03020  
TAP03030  
TAP03040

FILE TAPHDR

2200	NC=NRPDS	TAP03050
	WORD=0	TAP03060
	IWORD(3)=IBUF(131)	TAP03070
	IWORD(4)=IBUF(132)	TAP03080
	NS=WORD	TAP03090
	NDSPR=1	TAP03100
	NCAR=1	TAP03110
	SVD=13	TAP03120
	PRSZ=3596	TAP03130
	DSI=3596	TAP03140
	JREC(1)=TYPE	TAP03150
C		TAP03160
	SKIP REMAINDER OF HEADER FILE	TAP03170
C		TAP03180
2210	RFAD(IUNIT,S10,END=2220)DUMMY	TAP03190
	GO TO 2210	TAP03200
2220	CONTINUE	TAP03210
C		TAP03220
	FSCAN=1	TAP03230
	IFRST=1	TAP03240
	SMSTR=1	TAP03250
	WRITE(6,481)(FRM(I,4),I=1,3),NC,NS	TAP03260
	WRITE(6,500)IFRST,SMSTR	TAP03270
	RETURN	TAP03280
280	FORMAT(' UNRECOVERABLE ERROR READING HEADER RECORD')	TAP03290
300	FORMAT(' A LINE NO. IS LESS THAN OR EQUAL ZERO')	TAP03300
310	FORMAT(' LAST SCAN LINE READ',I5,' ISTAT=',I5)	TAP03310
330	FORMAT('/ INTERNAL DIMENSIONS TOO SMALL FOR DATA/' NO. OF CHANNELS	TAP03320
	*S ON DATA TAPE=',I7,' NO. OF POINTS/CHANNEL=',I7/)	TAP03330
340	FORMAT(' CHECK THE FOLLOWING POSSIBLE ERRORS/' 1. DATA TAPE IS NOT	TAP03340
	*T IN REQUESTED FORMAT')	TAP03350
360	FORMAT(' ONLY ONE OR LESS RECORDS PER CHANNEL ACCEPTABLE AT THIS	TAP03360
	*IME')	TAP03370
370	FORMAT(' NO. OF RECORDS PER DATA SET=',I5,' MUST BE LESS THAN OR	TAP03380
	*EQUAL I5')	TAP03390
390	FORMAT(' NO. OF BITS/PIXEL=',I5,' ONLY 8 BITS ACCEPTABLE AT THIS	TAP03400
	*IME')	TAP03410
400	FORMAT(' DATA ORDER INDICATOR=',I5/' DATA MUST BE ORDERED BY PIXEL	TAP03420
	*')	TAP03430
481	FORMAT(1H ///' INPUT IMAGE DATA TAPE INFORMATION'//	TAP03440
	* 5X,'FORMAT',T30,I4/	TAP03450
	* 5X,'NO. OF CHANNELS',T30,I4 /	TAP03460
	* 5X,'NO. OF PIXELS/LINE',T30,I4)	TAP03470
500	FORMAT(5X,'FIRST SCAN LINE NO.',T30,I4/	TAP03480
	* 5X,'FIRST PIXEL REFERENCE PT.',T30,I4)	TAP03490
510	FORMAT(1A4)	TAP03500
	END	TAP03510

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: WRTBMT

```
SUBROUTINE WRTBMT(BMAT,NOFET4,NOFET2,FETVC2)
DIMENSION CH(2)
DATA CH/'CH( 1,  )' /
INTEGER FETVC2(30)
C INCLUDE COMPKA.LIST
COMMON/GLOBAL/HEAD(63),MAYTAP,DATAP,SAVTAP,RMFILE,RMKEY,
* HISFIL,HISKEY,TRFORM,ENIPTP,ERPKEY,MAPUNT,NOFILF,
* DRUMAD,DRMWD,SPAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* NHSTUN,NHSTFI,SCTHUN,MAPFIL
* DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRJUNT,RANDIO
C$END
DIMENSION BMAT(NOFET4,NOFET2)
DOUBLE PRECISION BMAT
GO TO 4
ENTRY WRTBM(BMAT,NOFET4,NOFET2,FETVC2)
DIMENSION BMAT(NOFET4,NOFET2)
K=1
4 CONTINUE
WRITE(6,HEAD)
WRITE(6,100)NOFET4,NOFET2
IR=1
IK=12
5 IF(IK.GT.NOFET2)IK=NOFET2
WRITE(6,200)(CH(1),CH(2),I=IR,IK)
WRITE(6,300)(FETVC2(I),I=IR,IK)
WRITE(6,350)
IF(K.EQ.0)GO TO 11
DO 12 J=1,NOFET4
12 WRITE(6,400)J,(BMAT(J,I),I=IR,IK)
GO TO 13
11 CONTINUE
DO 10 J=1,NOFET4
10 WRITE(6,400)J,(BMAT(J,I),I=IR,IK)
13 CONTINUE
IF(IK.EQ.NOFET2)GO TO 20
IR=IK+1
IK=IK+12
GO TO 5
20 RETURN
100 FORMAT(///45X,'LINEAR TRANSFORMATION (B) MATRIX'//
* 50X,'NO. LINEAR COMB. -',I3/
* 50X,'NO. CHANNELS -',I3/)
200 FORMAT(12X,12(A4,A4,2X))
300 FORMAT(1F+,14X,11(I2,8X),I2)
350 FORMAT(1X,'LIN. COMB. ')
400 FORMAT(1X,I5,4X,12(1X,F9.3))
END
```

WRT00010  
WRT00020  
WRT00030  
WRT00040  
WRT00050  
WRT00060  
WRT00070  
WRT00080  
WRT00090  
WRT00100  
WRT00110  
WRT00120  
WRT00130  
WRT00140  
WRT00150  
WRT00160  
WRT00170  
WRT00180  
WRT00190  
WRT00200  
WRT00210  
WRT00220  
WRT00230  
WRT00240  
WRT00250  
WRT00260  
WRT00270  
WRT00280  
WRT00290  
WRT00300  
WRT00310  
WRT00320  
WRT00330  
WRT00340  
WRT00350  
WRT00360  
WRT00370  
WRT00380  
WRT00390  
WRT00400  
WRT00410  
WRT00420  
WRT00430  
WRT00440  
WRT00450  
WRT00460  
WRT00470  
WRT00480





FILE: WRTFLD

```
SUBROUTINE WRTFLD(FLDSAV,VERTEX,NOFLD,KEY,CLSNAM,SUBNAM)
IMPLICIT INTEGER(A-Z)
DIMENSION CLSNAM(1),SUBNAM(1)
C*
C* THIS SUBROUTINE PRINTS SAVED TRAINING OR TEST FIELDS
C*
C DIMENSION FLDSAV(4,NOFLD),VERTEX(2,1)
DATA LPRN/'/'
C INCLUDE COMRK6.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERHPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWDIS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCTHUN,MAPFIL
* ,DOTUNT,DOTFIL,NCMPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CROUNT,PRTUNT,RANDIO
C*END
IR=1
WRITE(6,HEAD)
IF(KEY.EQ.1)WRITE(6,100)
IF(KEY.EQ.2)WRITE(6,200)
IF(KEY.NE.3)WRITE(6,300)
IF(KEY.EQ.3)WRITE(6,250)
DO 10 I=1,NOFLD
NV=FLDSAV(4,I)
NQ=NV-1
NR=NQ-5
IF(NQ.GT.5)NQ=5
IE=IR+NQ-1
IC=FLDSAV(2,I)
IS=FLDSAV(3,I)
FLDNAM=FLDSAV(1,I)
IF(KEY.NE.3)GO TO 5
WRITE(6,700)I,FLDNAM,(LPRN,VERTEX(1,J),VERTEX(2,J),J=IB,IE)
IF(IS.FQ.1)WRITE(6,705)
IF(IS.FQ.2)WRITE(6,710)
GO TO 6
5 CONTINUE
IF(IS.FQ.0)WRITE(6,400)I,FLDNAM,IC,CLSNAM(IC),
(LPRN,VERTEX(1,J),VERTEX(2,J),J=IR,IE)
IF(IS.NE.0)WRITE(6,500)I,FLDNAM,IC,CLSNAM(IC),IS,SUBNAM(IS),
(LPRN,VERTEX(1,J),VERTEX(2,J),J=IB,IE)
6 CONTINUE
IF(NR.IE.0)GO TO 7
IR=IE+1
IE=IR+NR-1
WRITE(6,650)(LPRN,VERTEX(1,J),VERTEX(2,J),J=IB,IE)
7 CONTINUE
IR=IE+2
10 CONTINUE
RETURN
100 FORMAT(/// 20X,'AREA USED TO COMPUTE TRAINING STATISTICS'//)
200 FORMAT(/// 50X,'INPUT FIELDS'//)
250 FORMAT(///45X,'DESIGNATED FIELDS'///T18,'FIELD',T40,'DESIGNATED',
* TPO,'VERTICES (SAMPLE,LINE)'//)
300 FORMAT(1X,T1X,'FIELD',T34,'CLASS',T47,'SUBCLASS',T80,'VERTICES (S'
*AMPLE,LINE)'//)
400 FORMAT(T15,I3,T20,A4,T30,I3,T35,A4,T65,5(A1,I4,',',I4,')',1X))
500 FORMAT(T15,I3,T20,A4,T30,I3,T35,A4,T45,I3,T50,A4,T65,
* 5(A1,I4,',',I4,')',1X))
650 FORMAT(1X,T65,5(A1,I4,',',I4,')',1X))
700 FORMAT(T15,I3,T20,A4,T65,5(A1,I4,',',I4,')',1X))
705 FORMAT(1H+,T40,'UNIDENTIFIABLE')
710 FORMAT(1H+,T40,'OTHER')
END
```

WRT00010  
WRT00020  
WRT00030  
WRT00040  
WRT00050  
WRT00060  
WRT00070  
WRT00080  
WRT00090  
WRT00100  
WRT00110  
WRT00120  
WRT00130  
WRT00140  
WRT00150  
WRT00160  
WRT00170  
WRT00180  
WRT00190  
WRT00200  
WRT00210  
WRT00220  
WRT00230  
WRT00240  
WRT00250  
WRT00260  
WRT00270  
WRT00280  
WRT00290  
WRT00300  
WRT00310  
WRT00320  
WRT00330  
WRT00340  
WRT00350  
WRT00360  
WRT00370  
WRT00380  
WRT00390  
WRT00400  
WRT00410  
WRT00420  
WRT00430  
WRT00440  
WRT00450  
WRT00460  
WRT00470  
WRT00480  
WRT00490  
WRT00500  
WRT00510  
WRT00520  
WRT00530  
WRT00540  
WRT00550  
WRT00560  
WRT00570  
WRT00580  
WRT00590  
WRT00600  
WRT00610  
WRT00620  
WRT00630  
WRT00640

FILE WRTHED

```

SUBROUTINE WRTHED(NCHAN,FEAT,NSAMP,FRMAT,IUNIT)
IMPLICIT INTEGER (A-Z)
.....
THE PURPOSE OF TAPWRT IS TO WRITE A DATA TAPE IN EITHER UNIVER-
SAL FORMAT OR LARSYS II FORMAT. THERE ARE TWO ENTRY POINTS TO
THE SUBROUTINE -- WRTHED AND WRTLN.
WRTHED WRITES THE HEADER RECORD IN 32 BIT BYTES FOR LARSYS II
AND 8 BIT BYTES FOR UNIVERSAL. ONE CALL TO WRTHED MUST BE MADE
FOR EACH REEL OF TAPE. THIS INFORMATION IS PACKED.

CALL WRTHED(INC,FEAT,NSAMP,FORMAT,TRFORM)
NC -- NO. OF CHANNELS TO BE WRITTEN FOR EACH DATA SET
FEAT -- ARRAY CONTAINING CHANNELS TO BE WRITTEN
NSAMP -- NO. OF SAMPLES PER CHANNEL
FRMAT -- =1 FOR UNIVERSAL
TRFORM -- NO. OF TAPE OUTPUT UNIT
        =2 FOR LARSYS II

WRTLN WRITES THE DATA IN 8 BIT BYTES AND IS ALSO PACKED. A
CALL TO THIS ROUTINE MUST BE MADE FOR EACH DATA SET TO BE WRIT-
TEN

CALL WRTLN (IDATA,LSTLIN)
IDATA -- ARRAY CONTAINING DATA TO BE WRITTEN
LSTLIN -- = 0 FOR N-1 DATA SETS
        =-1 FOR LAST DATA SET

        ICHAN -- ACTIVE CHANNELS HAVE CORRESPONDING BIT POSITION
        PROFLG -- TURNED ON

CONTINUE
NCS -- NO. OF CHANNELS
NRITS -- NO. OF BITS PER BYTE
SVD -- START OF VIDEO DATA
NVE -- SAME AS NSAMP
PRSZ -- PHYSICAL RECORD SIZE IN BYTES
NCPZ -- NO. CHANNELS PER RECORD
NPRC -- NO. PHYSICAL RECORDS PER CHANNEL
NRPDS -- NO. OF RECORDS PER DATA SET
ANCLNG -- LENGTH OF ANCILLARY BLOCK IN BYTES
DOI -- DATA ORDER INDICATOR
SAMSTR -- SAMPLE START
COMWRD -- SIZE OF COMPUTER WORD IN BITS
NDSPR -- NO. OF DATA SETS PER RECORD
NSCAP -- NO. OF CHANNELS ON ANCILLARY RECORD

PACRAY -- DATA IS PACKED INTO THIS ARRAY AND THEN WRITTEN ON
TAPE BY CALLING INTRAN
ICOUNT -- RUNNING TOTAL OF NO. OF DATA SETS WRITTEN
.....
LOGICAL*1 PACRAY(3060)
REAL RAY(200)
DIMENSION NR(18),PACK(765)
DIMENSION IHYTES(18),FEAT(30),IRAY(200)
LOGICAL*1 VARIAB(2400)

C*** COMMON BLOCK CREATED AUG. 3,1979 TO SAVE LARSYS III HEADER
C
COMMON /IDSTUR/ IDD(250)
COMMON /WRTAP/ICOUNT,FRMAT,UNIT,VARBL(600),IREMD
COMMON/TAPFRD/IUNIT7,IFRST,FSCAN,SAMEND,SAMINC,READY,
1 NSCAN,LINC,LD(200),DSL,LHUF(30),JREC(30),IHYTE(30),
2 NRUES,LINEF,LINEF,LININC,NSAMPZ,NCHAN7,FORMTZ
DATA IHYTES/81,89,90,91,92,96,100,102,103,104,105,107,108,110,
* 753,1778,1785,1787/
DATA NR/4,1,1,1,2,2,2,1,1,1,2,1,2,2,1,1,2,2/
EQUIVALENCE (VARBL,VARIAB),(PACK,PACRAY),(IRAY,RAY)
EQUIVALENCE (VARBL(1),ICHAN), (VARBL(7),PRSZ), (VARBL(14),NOSAM),
1 (VARBL(2),PROFLG), (VARBL(8),NCPZ), (VARBL(15),COMWRD) WRT00710
2 (VARBL(3),NC), (VARBL(9),NPRC), (VARBL(16),NDSPR) WRT00720
3 (VARBL(4),NRITS), (VARBL(10),NRPDS), (VARBL(17),NCAR) WRT00730
4 (VARBL(5),SVD), (VARBL(11),ANCLNG), (VARBL(18),NSAM) WRT00740
5 (VARBL(6),NVE), (VARBL(12),DOI), (VARBL(13),SAMSTR) WRT00750
6 WRT00760
7

```



FILE WRTHED

```
DO 100 K=1,18
NBYTES = NH(K)
DO 90 L=1,NBYTES
LOC = 4 * L - NBYTES + (K-1)*4
BYTE = IRYTES(K)
90 PACRAY(BYTE+L-1) = VARIAB(LOC)
100 CONTINUE
PACRAY(61) = VARIAB(73)
PACRAY(62) = VARIAB(74)
PACRAY(63) = VARIAB(75)
PACRAY(67) = VARIAB(79)
PACRAY(68) = VARIAB(80)
DO 110 L = 1,512
110 PACRAY(111 + L) = VARIAB(111 + L)
DO 120 L = 1,16
120 PACRAY(2200 + L) = VARIAB(2200 + L)
DO 130 I = 1,4
IPAT = 2254 + (I - 1)*8
130 PACRAY(IPAT) = VARIAB(IPAT)
C
C*** THE ABOVE THREE LINES IS AN AD HOC ADDITION FOR SOIL LINES
C*** ADDED OCT. 23,1978
C
CALL WRTREC(UNIT,PRSZHD,PACRAY)
RETURN
END
```

WRT01530  
WRT01540  
WRT01550  
WRT01560  
WRT01570  
WRT01580  
WRT01590  
WRT01600  
WRT01610  
WRT01620  
WRT01630  
WRT01640  
WRT01650  
WRT01660  
WRT01670  
WRT01680  
WRT01690  
WRT01700  
WRT01710  
WRT01720  
WRT01730  
WRT01740  
WRT01750  
WRT01760  
WRT01770  
WRT01780

FILE WRTLN

```

C      SURROUTINE WRTLN(/IDATA/,LSTLIN)
      IMPLICIT INTEGER (A-Z)
      COMMON /WRTAP/ICOUNT,FORMAT,UNIT,VARBL(600),IREMD
      LOGICAL*1 PACRAY(13500),ISCAN(4),IDATA(1),IRECNO(4)
      LOGICAL*1 ZERO(4),LONE(4)
      EQUIVALENCE (LONE,IONE)
      DATA IONE/ZFFFF/
      EQUIVALENCE (ICOUNT,ISCAN)
      EQUIVALENCE (RECNO,IRECNO)
      EQUIVALENCE (ZERO,IZERO)
      EQUIVALENCE (VARHL(3),NC),(VARBL(7),PRSZ),
      *          (VARBL(8),NCPR),(VARBL(9),NPRC),
      *          (VARBL(10),NRPDS),(VARBL(11),ANCLNG),
      *          (VARHL(18),NSAMP),(VARBL(17),NCAR)
      ICOUNT = ICOUNT + 1
      IZFRO = 0
      ANC = ANCLNG + 2
      IF (FORMAT .EQ. 1)      GO TO 140
C
C      WRITES PACKED DATA ON TAPE IN LARSYS II FORMAT
C
C 150 FORMAT(3I5)
C      PACKING ONE SET OF DATA INTO ONE RECORD
C
      NRYTES = 8
      ANCLNG = 4
      NRYTES = (NSAMP + 6)*NC
      PACRAY(1) = ISCAN(3)
      PACRAY(2) = ISCAN(4)
      PACRAY(3) = LONE(4)
      PACRAY(4) = LONE(4)
C
C*** ADDED AUG 10,1979 TO ADD CALIBRATION SPACE
C
      IV = 0
      III = 0
      DO 120 II = 1,NC
      DO 110 I = 1,NSAMP
      III = III + 1
      IV = IV + 1
      PACRAY(IV + 4) = IDATA(III*4)
110     CONTINUE
      IV = IV + 6
120     CONTINUE
      NRYTES=NRYTES+ANCLNG
      IIDUM=(NRYTES/4)*4
      IIDUM=NRYTES-IIDUM
      IF (IIDUM.NE.0) NRYTES=NRYTES+4-IIDUM
      CALL WRTREC(UNIT,NRYTES,PACRAY)
      IF (LSTLIN .EQ. -1) ENDFILE UNIT
      RETURN
C
C      WRITE PACKED DATA ON TAPE IN UNIVERSAL FORMAT
C
140 DO 150 I=1,72
150 PACRAY(I) = ZERO(4)
      ROW = 1
      NR = NRPDS - 1
      IF (NCPR.EQ.0) GO TO 155
      NCLR = MOD((NC-NCAR),NCPR)
155 WORD = 73
C
C      PACKING ANCILLARY INFORMATION INTO PACRAY
C
      RECNO = 1
      PACRAY(2) = IRECNO(4)
      PACRAY(71) = ISCAN(3)
      PACRAY(72) = ISCAN(4)
C
C      DATA IS NOT PACKED WITH ANCILLARY RECORD
C
      IF (NCAR .NE. 0)      GO TO 160
      NRYTES = NSAMP * NC
      KA = 1
      GO TO 210

```

WRT00010  
WRT00020  
WRT00030  
WRT00040  
WRT00050  
WRT00060  
WRT00070  
WRT00080  
WRT00090  
WRT00100  
WRT00110  
WRT00120  
WRT00130  
WRT00140  
WRT00150  
WRT00160  
WRT00170  
WRT00180  
WRT00190  
WRT00200  
WRT00210  
WRT00220  
WRT00230  
WRT00240  
WRT00250  
WRT00260  
WRT00270  
WRT00280  
WRT00290  
WRT00300  
WRT00310  
WRT00320  
WRT00330  
WRT00340  
WRT00350  
WRT00360  
WRT00370  
WRT00380  
WRT00390  
WRT00400  
WRT00410  
WRT00420  
WRT00430  
WRT00440  
WRT00450  
WRT00460  
WRT00470  
WRT00480  
WRT00490  
WRT00500  
WRT00510  
WRT00520  
WRT00530  
WRT00540  
WRT00550  
WRT00560  
WRT00570  
WRT00580  
WRT00590  
WRT00600  
WRT00610  
WRT00620  
WRT00630  
WRT00640  
WRT00650  
WRT00660  
WRT00670  
WRT00680  
WRT00690  
WRT00700  
WRT00710  
WRT00720  
WRT00730  
WRT00740  
WRT00750  
WRT00760

FILE WRTLN

```

C
C 160 ALL DATA IS PACKED ON ANCILLARY RECORD
      IF (NCAR .NE. NC) GO TO 170
      NBYTES = IREMD - ANC
      KA = 2
      GO TO 210
C
C 170 PART OF DATA IS PACKED ON ANCILLARY RECORD
      NBYTES = NCAR*NSAMP
      KA = 3
      GO TO 210
C
C 180 DATA IS PACKED ON MORE THAN ONE RECORD
      ANC = 2
      WORD = 3
      KA = 4
      J = 0
185 J = J + 1
      IF ( J.GT.NR) GO TO 200
      WRITE(6,660)NR
660 FORMAT(' NR',I5)
      RECNO = RECNO + 1
      PACRAY(2) = IRECNO(4)
      NBYTES = NCPR * NSAMP
      IF (NCLR .NE. 0 .AND. J .EQ. NR) NBYTES = NCLR*NSAMP
      GO TO 210
200 CONTINUE
      IF (LSTLIN .EQ. -1) ENDFILE UNIT
      RETURN
210 IF (NCAR .EQ. 0) GO TO 220
      II = (ROW-1)*NSAMP*4
      DO 215 I=1,NBYTES
215 PACRAY(WORD*I-1) = IDATA(4*I+II)
220 CALL WRTREC(UNIT,PRSZ,PACRAY)
      IF (KA .NE. 4) ROW = NCAR + ROW
      IF (KA .EQ. 4) ROW = ROW + NCPR
      GO TO (180,200,180,185),KA
END

```

WRT00770  
WRT00780  
WRT00790  
WRT00800  
WRT00810  
WRT00820  
WRT00830  
WRT00840  
WRT00850  
WRT00860  
WRT00870  
WRT00880  
WRT00890  
WRT00900  
WRT00910  
WRT00920  
WRT00930  
WRT00940  
WRT00950  
WRT00960  
WRT00970  
WRT00980  
WRT00990  
WRT01000  
WRT01010  
WRT01020  
WRT01030  
WRT01040  
WRT01050  
WRT01060  
WRT01070  
WRT01080  
WRT01090  
WRT01100  
WRT01110  
WRT01120  
WRT01130  
WRT01140  
WRT01150  
WRT01160

FILE: WRTMTX

```
C      SUBROUTINE WRTMTX(MATICE,SIZE,BCD)
C      IMPLICIT INTEGER (A-H,O-Z)
C      DIMENSION FORMAT(6)
C      REAL MATICE(1)
-----
C      CALL..    CALL WRTMTX(MATICE,SIZE,FREQ,BCD,MAXFET)
C      ARGV..    MATICE - COVARIANCE MATRICE
C                SIZE  - RANK OF 'MATRICE' ('DMATIC')
C                FREQ  - FREQUENCY MATRIX
C                BCD   - CONTAINS BCD PRECISION FOR PRINTOUT
C                MAXFET - NUMBER OF FEATURES PER LINE
C      REQUIRES.. NONE
C      PURPOSE..  PRINTS THE SINGLE-PRECISION COVARIANCE MATRICES
C      RETURNS..  NO CHANGE
-----
C      CONTINUE
C      CALL..    CALL DWRTMX(MATICE,SIZE,FREQ,BCD,MAXFET)
C      ARGV..    SEE ABOVE
C      PURPOSE.. PRINTS THE DOUBLE-PRECISION COVARIANCE MATRICES
C      RETURNS.. SEE ABOVE
-----
-----
C      DATA FORMAT/'(1H0',',',6X,',',12F9',',',',',',',') '/
C      DOUPRE = 0
C      GO TO 10
C      ENTRY DWRTMX(DMATIC,SIZE,BCD)
C      DOUBLE PRECISION DMATIC(1)
C      DOUPRE = 1
10  FORMAT(5)=BCD
DO 100 LOC=1,SIZE,12
STOP = LOC+11
IF ( STOP .GT. SIZE) STOP = SIZE
II = 1
KINC = 1
DO 90 I=LOC,SIZE
K = I*(I+1)/2-II+1
JK = K+KINC-1
IF(DOUPRE.EQ.0) WRITE(6,FORMAT) (MATICE(J),J=K,JK)
IF(DOUPRE.EQ.1) WRITE(6,FORMAT) (DMATIC(J),J=K,JK)
II = II+1
90 IF(KINC.LT.12.AND.KINC.LT.STOP)KINC=KINC+1
WRITE(6,1004)
100 CONTINUE
1004 FORMAT('0')
RETURN
-----
C      END
```

19-107  
4-80

FILE WRTREC

```
      SUBROUTINE WRTREC(UNIT,LENGTH,IBUF)
      IMPLICIT INTEGER (A-Z)
C
C      OUTPUTS A SCAN LINE OF DATA
C
      DIMENSION IBUF(3000)
      LENTH = LENGTH/4
C      WRITE(6,200)UNIT,LENTH,IBUF(18)
C      WRITE(UNIT,100)((IBUF(I),I=1,LENTH)
      WRITE(6,200)UNIT,LENGTH,IBUF(18)
200  FORMAT(/3I6)
100  FORMAT(3I(250A4))
      RETURN
      END
```

```
WRT00010
WRT00020
WRT00030
WRT00040
WRT00050
WRT00060
WRT00070
WRT00080
WRT00090
WRT00100
WRT00110
WRT00120
WRT00130
WRT00140
```



20. DAMRG PROCESSOR

FILE DAMRG

```

SUBROUTINE DAMRG (ARRAY, TOP)
IMPLICIT INTEGER (A-Z)
REAL SUNCOR, DIM
LOGICAL IDL (400), VARIAB (2400), LOGSUN (32)
EQUIVALENCE (IDL, IDL), (VARIAB, VARIAB), (SUNANG, LOGSUN)
COMMON/GLOBAL/HEAD (63), MAPTAP, DATAPE, SAVTAP, HMF, ILE, BMKEY,
* HISFIL, HISKEY, TRFORM, ERIP, EHPKEY, MAPUNT, NOFILE,
* DMUMAD, DMUMOS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL,
* NMSTIIN, NMSTFI, SCTRIN, MAPFIL,
* DOTUNT, DOTFIL, NCHPAS, TRNSFL, RMTRFL, MISTFL, PCMUNT,
* CRDUNT, PRTUNT, MANDIO
COMMON/TAPERD/INIT, IFIRST, FSCAN, SAMEND, SAMINC, READY, NSCAN,
* LINC, ID (200), DSL, LAUF (30), JREC (30), IBYTE (30), NBUFS, FILENO, LINENO,
* LINC, NSAMP, NOCHAN, FORMT
COMMON/TAP/ICOUNT, DUMMY, UNIT, VAR9L (600), IREM0
COMMON/ISOLNK/SUNANG (8), ISUNT, ISUNC, SMSTP, SMINC, LINSKP
COMMON/MFG/DAT/IMPT, ISOPT, NUMFIL, IDATP (6), IDATFL (6),
* NOFEAT, NFEAT (6), FETVEC (30,6), ISUN (8,6), SUNCOR (30),
* FLINF (6,6), NOSAMP, NOLINE, NSS (6), NACROS, NLINES (6), LINPTR (7),
* LINF (600), FORMM
DIMENSION ARRAY (1)
CALL SETIH (ARRAY, TOP)
C*** MAJOR LOOP ON FILES
    SWITCH=0
    NI = 0
    IDPPP = 0
    ICCT = 0
    DO 700 I = 1, NUMFIL
        IF (I.EQ.2.AND.SWITCH.EQ.1) GO TO 80
C*** CALL TAPE HEADER READ PROGRAM WITH UNIT AND FILE
        IDATU = IDATP (I)
        IDATF = IDATFL (I) - 1
        CALL TAPHDR (IDATU, IDATF)
C*** CODE ADDED TO REFORMAT LANDSAT III INTO LARSYS OR UNIVERSAL
        DAMRG FOR LANDSAT III SEQUENTIAL FORMAT IS A SIMPLE
        CHANNEL CHANNEL MERGE OF TWO FILES
        DAMRG FOR LANDSAT III INTERLEAVED FORMAT IS FORCED
        TO LOOK LIKE A SPATIAL MERGE OF 1 FIELD ACROSS AND 1 OR 2 DOWN
        ONLY A REFORMAT CAN BE DONE FOR INTERLEAVED NO MERGE IS
        POSSIBLE UNTIL THE REFORMAT IS COMPLETE
        IF (FORMT.EQ.4.AND).JREC (1).EQ.1) SWITCH=1
        IF (SWITCH.EQ.0) GO TO 100
        NOFEAT=NFEAT (1)
        NACROS=1
        IMPT=2
        FLDSTH=FLDINF (1,1)
        FLDLST=FLDINF (2,1)
        IF (FLDINF (1,1).LE.149) GO TO 50
C*** FIELD ENTIRELY ON SECOND TAPE
        FLDINF (1,1)=FLDINF (1,1)-149
        FLDINF (2,1)=FLDINF (2,1)-149
        NUMFIL=1
        GO TO 80
C*** IF (FLDINF (2,1).LE.149) GO TO 75
        FIELD OVERLAPS BOTH TAPES
        FLDINF (2,2)=FLDINF (2,1)-149
        FLDINF (2,1)=149-MOD ((149-FLDINF (1,1)), FLDINF (3,1))
        FLDINF (1,2)=FLDINF (2,1)+FLDINF (3,1)-149
        FLDINF (3,2)=FLDINF (3,1)
        NSS (2)=NSS (1)
        NLINES (1)=(FLDINF (2,1)-FLDINF (1,1))/FLDINF (3,1)+1
        NLINES (2)=(FLDINF (2,2)-FLDINF (1,2))/FLDINF (3,2)+1
        GO TO 100
C*** FIELD ENTIRELY ON FIRST TAPE

```

20-1  
482

FILE DAMRG

```

C
75   NUMFIL=1
    GO TO 100
C
    PROCESSING SECOND FILE FOR LARSYS III INTERLEAVED
C
80   IUNIT=IDATTP(2)
90   HEAD(IUNIT,1100,END=100)DUMMY
1100  FORMAT(1A*)
    GO TO 90
C*** NUMBER OF CHANNELS FOR THIS FILE
100  NF = NFEAT(I)
    CALL FLDINT(FLDINF(1,I),FETVEC(1,I),NF)
C*** SET FEATURE COUNTER
    IF(I.GT.1) NI = NI + NFEAT(I - 1)
C*** SET SCALAR FIELD DESCRIPTION FOR THIS FILE
    SAMSTR = FLDINF(4,I)
    SAMINC = FLDINF(6,I)
    SAMEND = FLDINF(5,I)
    LINSTR = FLDINF(1,I)
    LININC = FLDINF(3,I)
    LINEND = FLDINF(2,I)
C*** NO. SAMPLES/LINE FOR FILE I COMPUTED IN FLDINT
    NS = NSAMP
C*** TOTAL NUMBER OF RADIANCE VALUES PER LINE
    NV = NS * NF
C*** WRITE FIELD INFORMATION FOR FILE I
    IF(SWITCH.EQ.1.AND.I.EQ.2)GO TO 1090
    WRITE (PRTUNT,1060) I
1060  FORMAT('O INPUT FIELD DESCRIPTION FOR FILE',I8)
    WRITE (PRTUNT,1070)
1070  FORMAT('O START LINE END LINE LINE INC START PIXEL END
    *PIXEL PIXEL INC')
    IF(SWITCH.EQ.1)GO TO 1085
    WRITE (PRTUNT,1080) (FLDINF(J,I), J=1,6)
1080  FORMAT('O ',6(4X,14.4X))
    GO TO 1090
1085  WRITE(PRTUNT,1080)FLDSTR*FLDLST,(FLDINF(J,I),J=3,6)
C*** STORE SUN ANGLES
1090  IF(ISUNT.NE.0)GO TO 610
    IF(ISUNC.EQ.0)GO TO 620
C*** SUN ANGLES FROM CARDS
    DO 600 J = 1,8
    ID(16 + J) = ISUN(J,I)
    SUNANG(J) = ISUN(J,I)
600  CONTINUE
    GO TO 620
C*** SUN ANGLES FROM TAPE HEADER
610  DO 615 J = 1,8
    ISUN(J,I) = SUNANG(J)
615  CONTINUE
    CALL SUNFAC(SUNCOR,SUNANG,FETVEC(1,I),NF,ISUNT,ISUNC)
620  CONTINUE
    IF (FORMM.NE.1) GO TO 660
    WRITE (PRTUNT,1000) I
1000  FORMAT('O DATA FOR INPUT FILE',I4)
    WRITE (PRTUNT,1010)IDL(100),IDL(104),IDL(108),IDL(111),IDL(112)
1010  FORMAT('O INPUT FILE DATE AND SITE',5(3X,Z2))
C*** LOAD VARJAB WITH EXTRA HEADER INFORMATION

```

DAM00770  
DAM00780  
DAM00790  
DAM00800  
DAM00810  
DAM00820  
DAM00830  
DAM00840  
DAM00850  
DAM00860  
DAM00870  
DAM00880  
DAM00890  
DAM00900  
DAM00910  
DAM00920  
DAM00930  
DAM00940  
DAM00950  
DAM00960  
DAM00970  
DAM00980  
DAM00990  
DAM01000  
DAM01010  
DAM01020  
DAM01030  
DAM01040  
DAM01050  
DAM01060  
DAM01070  
DAM01080  
DAM01090  
DAM01100  
DAM01110  
DAM01120  
DAM01130  
DAM01140  
DAM01150  
DAM01160  
DAM01170  
DAM01180  
DAM01190  
DAM01200  
DAM01210  
DAM01220  
DAM01230  
DAM01240  
DAM01250  
DAM01260  
DAM01270  
DAM01280  
DAM01290  
DAM01300  
DAM01310  
DAM01320  
DAM01330  
DAM01340  
DAM01350  
DAM01360  
DAM01370  
DAM01380  
DAM01390  
DAM01400  
DAM01410  
DAM01420  
DAM01430  
DAM01440  
DAM01450  
DAM01460  
DAM01470  
DAM01480  
DAM01490  
DAM01500  
DAM01510  
DAM01520

FILE DAMRG

```

C
C*** LOAD DATE AND SITE FROM FIRST FILE
C
    IF(I.NE.1)GO TO 625
    VARIAB(73) = IDL(100)
    VARIAB(74) = IDL(104)
    VARIAB(75) = IDL(108)
    VARIAB(79) = IDL(111)
    VARIAB(80) = IDL(112)
C
C*** FOR PURPOSES OF UNIVERSAL HEADER WRITE LOAD VARIAB WITH SUN ANGLES
C*** AND GAINS AND PHASES ONLY IF CHANNEL MERGE OPTION
C
625 IF(IMOPT.NE.1)GO TO 660
    DO 650 J = 1,NF
    IDUM = (FETVEC(J,1) - 1) * 2
    I1 = 112 + IDUM
    I2 = 112 + N1 * 2 + (J - 1) * 2
    VARIAB(I2) = IDL(I1 + 3)
    VARIAB(I2 + 1) = IDL(I1 + 4)
    I1 = 240 + IDUM
    I2 = 240 + N1 * 2 + (J - 1) * 2
    VARIAB(I2) = IDL(I1 + 3)
    VARIAB(I2 + 1) = IDL(I1 + 4)
    I1 = 368 + IDUM
    I2 = 368 + N1 * 2 + (J - 1) * 2
    VARIAB(I2) = IDL(I1 + 3)
    VARIAB(I2 + 1) = IDL(I1 + 4)
    I1 = 496 + IDUM
    I2 = 496 + N1 * 2 + (J - 1) * 2
    VARIAB(I2) = IDL(I1 + 3)
    VARIAB(I2 + 1) = IDL(I1 + 4)
650 CONTINUE
    KS = 0
    DO 655 J = 1,NF
    IDUM = FETVEC(J,1)
    IDUM = (IDUM - 1) / NCHPAS
    IF(ISUNT.EQ.0.AND.J.EQ.1) KS = IDUM
    I1 = (IDUM - KS) * 4 + 3
    I2 = 2201 + (N1 + J - 1) * 2
    VARIAB(I2) = LOGSUN(I1)
    VARIAB(I2 + 1) = LOGSUN(I1 + 1)
655 CONTINUE
660 CONTINUE
C
C*** THE NEXT LINE WAS ADDED OCT. 23,1978 AS AN AD HOC ADDITION
C*** TO ADD SOIL LINES TO THE UNIVERSAL HEADER
C
    VARIAB(2246 + 8*I) = IDL(640)
C
C
C*** INITIALIZATION FOR LINE EXTRACTION PARAMETERS NEEDED FOR
C*** SPATIAL MERGE
C
    LOC = (I - 1) / NACROS
    N5 = 0
    IF (LOC.EQ.0) GO TO 666
    DO 661 J = 1,LOC
    661 N5 = N5 + NLINES(J)
    666 CONTINUE
    LREM = (I - 1) - LOC * NACROS
    N2 = 0
    IF (LREM.EQ.0)GO TO 663
    DO 662 J = 1,LREM
    662 N2 = N2 + NSS(J)
    663 CONTINUE
    ICT = 0
C
C*** PARAMETERS NEEDED IF PSEUDO MERGE OPTION
C
    LPTR = LNPTR(I)
    NL = NLINES(I)
    NLM = NL + LPTR - 1
C
C*** EXTRACT FIELD FOR THIS FILE LINE BY LINE
C

```

DAM01530  
DAM01540  
DAM01550  
DAM01560  
DAM01570  
DAM01580  
DAM01590  
DAM01600  
DAM01610  
DAM01620  
DAM01630  
DAM01640  
DAM01650  
DAM01660  
DAM01670  
DAM01680  
DAM01690  
DAM01700  
DAM01710  
DAM01720  
DAM01730  
DAM01740  
DAM01750  
DAM01760  
DAM01770  
DAM01780  
DAM01790  
DAM01800  
DAM01810  
DAM01820  
DAM01830  
DAM01840  
DAM01850  
DAM01860  
DAM01870  
DAM01880  
DAM01890  
DAM01900  
DAM01910  
DAM01920  
DAM01930  
DAM01940  
DAM01950  
DAM01960  
DAM01970  
DAM01980  
DAM01990  
DAM02000  
DAM02010  
DAM02020  
DAM02030  
DAM02040  
DAM02050  
DAM02060  
DAM02070  
DAM02080  
DAM02090  
DAM02100  
DAM02110  
DAM02120  
DAM02130  
DAM02140  
DAM02150  
DAM02160  
DAM02170  
DAM02180  
DAM02190  
DAM02200  
DAM02210  
DAM02220  
DAM02230  
DAM02240  
DAM02250  
DAM02260  
DAM02270  
DAM02280

FILE DAMRG

```

DO 690 II = LINSTR,LINEND,LININC
ICT = ICT + 1
CALL LINERD(ARRAY(1),ENDTAP)
IF (ENDTAP.EQ.-1) CALL CMERR
IF (IMOPT.NE.3) GO TO 670
C*** LOOK FOR LINE MATCH IF PSEUDO MERGE
C
DO 665 J = LPTR,NLM
IF (II.EQ.LINES(J)) GO TO 670
665 CONTINUE
GO TO 690
670 CONTINUE
ICCT = ICCT + 1
C
IF (IPPPP.EQ.1) WRITE (PRTUNT,1020) (ARRAY(K),K=1,NV)
1020 FORMAT(/10I7)
IF (IMOPT.NE.1) GO TO 675
C*** CHANNEL MERGE MODE WRITE NV VALUES TO DIRECT ACCESS FILE
C
IF (ISOPT.EQ.0) GO TO 672
C*** DO SUN ANGLE CORRECTION
C
DO 671 J = 1,NF
DO 671 JJ = 1,NS
ITEMP = (JJ + (J - 1) * NS)
DUM = SUNCOR(J) * FLOAT(ARRAY(ITEMP))
ARRAY(ITEMP) = IFIX(DUM)
671 CONTINUE
672 ADDRES = DRUMAD + N1*NS + (ICT - 1)*NS*NOFEAT
CALL RWRITE(ADDRES,ARRAY(1),NV,STATUS)
GO TO 690
675 IF (IMOPT.NE.2) GO TO 680
C*** SPATIAL MERGE MODE WRITE NSS(I)*NF VALUES TO DIRECT ACCESS FILE
C
IF (ISOPT.EQ.0) GO TO 677
DO 676 J = 1,NF
DO 676 JJ = 1,NS
ITEMP = (JJ + (J - 1) * NS)
DUM = SUNCOR(J) * FLOAT(ARRAY(ITEMP))
ARRAY(ITEMP) = IFIX(DUM)
676 CONTINUE
677 N4 = NSS(I)
DO 679 J=1,NF
ADDRES = DRUMAD + (NS + ICT-1)*NOSAMP*NF + NOSAMP*(J-1)*N2
ITEMP = 1 + (J-1)*N4
CALL RWRITE(ADDRES,ARRAY(ITEMP),N4,STATUS)
679 CONTINUE
GO TO 690
C*** PSEUDO MERGE OPTION
C
680 IF (ISOPT.EQ.0) GO TO 682
DO 681 J = 1,NF
DO 681 JJ = 1,NS
ITEMP = (JJ + (J - 1) * NS)
DUM = SUNCOR(J) * FLOAT(ARRAY(ITEMP))
ARRAY(ITEMP) = IFIX(DUM)
681 CONTINUE
682 ADDRES = DRUMAD + (ICCT - 1) * NV
CALL RWRITE(ADDRES,ARRAY(1),NV,STATUS)
690 CONTINUE
C*** LINE LOOP COMPLETE
C
700 CONTINUE
C*** LOOP FOR FILE I COMPLETE
C*** WRITE OUTPUT FILE
C
DATFI = DATFIL - 1
C

```

DAM02290  
DAM02300  
DAM02310  
DAM02320  
DAM02330  
DAM02340  
DAM02350  
DAM02360  
DAM02370  
DAM02380  
DAM02390  
DAM02400  
DAM02410  
DAM02420  
DAM02430  
DAM02440  
DAM02450  
DAM02460  
DAM02470  
DAM02480  
DAM02490  
DAM02500  
DAM02510  
DAM02520  
DAM02530  
DAM02540  
DAM02550  
DAM02560  
DAM02570  
DAM02580  
DAM02590  
DAM02600  
DAM02610  
DAM02620  
DAM02630  
DAM02640  
DAM02650  
DAM02660  
DAM02670  
DAM02680  
DAM02690  
DAM02700  
DAM02710  
DAM02720  
DAM02730  
DAM02740  
DAM02750  
DAM02760  
DAM02770  
DAM02780  
DAM02790  
DAM02800  
DAM02810  
DAM02820  
DAM02830  
DAM02840  
DAM02850  
DAM02860  
DAM02870  
DAM02880  
DAM02890  
DAM02900  
DAM02910  
DAM02920  
DAM02930  
DAM02940  
DAM02950  
DAM02960  
DAM02970  
DAM02980  
DAM02990  
DAM03000  
DAM03010  
DAM03020  
DAM03030  
DAM03040

FILE DAMRG

C*** POSITION OUTPUT FILE	DAM03050
C	DAM03060
OPEN DATAPE	DAM03070
CALL FFSMFL(DATAPE, DATFI, ISTAT)	DAM03080
C	DAM03090
C*** SET OUTPUT CHANNELS 1,2,....,NOFEAT	DAM03100
C	DAM03110
DO R00 I = 1,NOFEAT	DAM03120
FETVEC(I,1) = I	DAM03130
R00 CONTINUE	DAM03140
C	DAM03150
C*** WRITE HEADER OF OUTPUT FILE	DAM03160
C	DAM03170
CALL WRTHED(NOFEAT, FETVEC(1,1), NOSAMP, FORMM, DATAPE)	DAM03180
C	DAM03190
C*** EXTRACT SCAN LINES ONE AT A TIME WRITE TO OUTPUT FILE	DAM03200
C	DAM03210
LSTLIN = 0	DAM03220
NV = NOSAMP * NOFEAT	DAM03230
DO R50 I = 1, NOLINE	DAM03240
IF (I.EQ. NOLINE) LSTLIN = - 1	DAM03250
ADDRES = DRUMAD + (I - 1) * NV	DAM03260
CALL PREAD(ADDRES, ARRAY(1), NV, ISTAT)	DAM03270
CALL WRTLN(ARRAY(1), LSTLIN)	DAM03280
IF (IPPPP.EQ.1) *WRITE(PRTUNT, 1030) I	DAM03290
1030 FORMAT(*, 'OUTPUT LINE', I6)	DAM03300
IF (IPPPP.EQ.1) *WRITE(PRTUNT, 1020) (ARRAY(K), K=1, NV)	DAM03310
R50 CONTINUE	DAM03320
C	DAM03330
C*** RETURN TO MONTOR	DAM03340
C	DAM03350
C	DAM03360
C*** OUTPUT FILE COMPLETED	DAM03370
C	DAM03380
RETURN	DAM03390
END	DAM03400



FILE: SET18

```
90 CONTINUE
WRITE (PRUNIT,1030) CODE,CARD
1030 FORMAT(' INVALID CARD - IGNORED',/T5,A4,6X,62A1)
GO TO 80
C*** NUMBER OF CHANNELS PER PASS CARD IMAGE
C
100 J = NXTCHR(CARD,COL)
IF (J.EQ.BLANK) GO TO 80
J = NUMBER(CARD,COL,NCHPAS,ISTART)
GO TO 80
C*** FORMAT CARD IMAGE - OUTPUT FILE
C
150 J = NXTCHR(CARD,COL)
IF (J.EQ.LRCD) FOKMM = 2
GO TO 80
C*** DATE CARD
C
200 READ (RRUNIT,1040) DATE
1040 FORMAT(10X,3A4)
REWIND RRUNIT
GO TO 80
C*** HED1 CARD
C
250 READ (RRUNIT,1050) HED1
1050 FORMAT(10A,15A4)
REWIND RRUNIT
GO TO 80
C*** HED2 CARD
C
300 READ (RRUNIT,1050) HED2
REWIND RRUNIT
GO TO 80
C*** NLINES (6) CARD FOR PSEUDO OPTION
C
350 J = NUMBER(CARD,COL,ARRAY(1),ISTART)
IF (J.GT.6) J = 6
DO 360 JJ = 1,J
360 NLINFS(JJ) = ARRAY(JJ)
GO TO 80
C*** OPTION CARD IMAGE
C
400 J = NXTCHR(CARD,COL)
IF (J.EQ.CRCD) IMOPT = 1
IF (J.EQ.SRCD) IMOPT = 2
IF (J.EQ.PRCD) IMOPT = 3
IF (J.EQ.ABCD) ISOPT = 1
GO TO 80
C*** SPATIAL OPTION ... NUMBER FIELDS TO BE JOINED ACROSS
C
450 J = NUMBER(CARD,COL,NACROS,ISTART)
GO TO 80
C*** SUN ANGLE CARD IMAGES
C
500 J = NXTCHR(CARD,COL)
IF (J.EQ.IRCD) GO TO 510
ISUNT = 1
GO TO 80
510 ISUNCT = ISUNCT + 1
COL = 0
ISUNC = 1
J = NUMBER(CARD,COL,ARRAY(1),ISTART)
DO 520 JJ = 1,J
ISUN(JJ,ISUNCT) = ARRAY(JJ)
520 CONTINUE
GO TO 80
C*** DATA TAPE CARD IMAGES
C
550 J = NXTCHR(CARD,COL)
IF (J.EQ.IHCD) GO TO 570
```

SE T00800  
SE T00810  
SE T00820  
SE T00830  
SE T00840  
SE T00850  
SE T00860  
SE T00870  
SE T00880  
SE T00890  
SE T00900  
SE T00910  
SE T00920  
SE T00930  
SE T00940  
SE T00950  
SE T00960  
SE T00970  
SE T00980  
SE T00990  
SE T01000  
SE T01010  
SE T01020  
SE T01030  
SE T01040  
SE T01050  
SE T01060  
SE T01070  
SE T01080  
SE T01090  
SE T01100  
SE T01110  
SE T01120  
SE T01130  
SE T01140  
SE T01150  
SE T01160  
SE T01170  
SE T01180  
SE T01190  
SE T01200  
SE T01210  
SE T01220  
SE T01230  
SE T01240  
SE T01250  
SE T01260  
SE T01270  
SE T01280  
SE T01290  
SE T01300  
SE T01310  
SE T01320  
SE T01330  
SE T01340  
SE T01350  
SE T01360  
SE T01370  
SE T01380  
SE T01390  
SE T01400  
SE T01410  
SE T01420  
SE T01430  
SE T01440  
SE T01450  
SE T01460  
SE T01470  
SE T01480  
SE T01490  
SE T01500  
SE T01510  
SE T01520  
SE T01530  
SE T01540  
SE T01550  
SE T01560  
SE T01570  
SE T01580

FILE: SET1A

```
IF(J.NE.ORCD)GO TO 595
C
C*** OUTPUT FILE
C
J = FIND12(CARD,COL,SLASH)
IF(J.NE.2)GO TO 595
J = NXTCHR(CARD,COL)
IF(J.EQ.FRCD)GO TO 555
C
C*** UNIT NUMBER OF OUTPUT FILE, THEN FILE
C
J = FIND12(CARD,COL,EQUEVC)
IF(J.NE.2)GO TO 595
J = NUMBER(CARD,COL,DATAPE,ISTART)
J = FIND12(CARD,COL,EQUEVC)
IF(J.NE.2)GO TO 595
ISTART = 0
J = NUMBER(CARD,COL,DATFIL,ISTART)
GO TO A0
C
C*** FILE NUMBER, THEN UNIT
C
555 J = FIND12(CARD,COL,EQUEVC)
IF(J.NE.2)GO TO 595
J = NUMBER(CARD,COL,DATFIL,ISTART)
J = FIND12(CARD,COL,EQUEVC)
IF(J.NE.2)GO TO 595
ISTART = 0
J = NUMBER(CARD,COL,DATAPE,ISTART)
GO TO A0
C
C*** INPUT FILES
C
570 NUMFIL = NUMFIL + 1
IF(NUMFIL.GT.6)GO TO 590
J = FIND12(CARD,COL,SLASH)
IF(J.NE.2)GO TO 590
J = NXTCHR(CARD,COL)
IF(J.EQ.FRCD)GO TO 575
C
C*** UNIT NUMBER OF INPUT FILE, THEN FILE NUMBER
C
J = FIND12(CARD,COL,EQUEVC)
IF(J.NE.2)GO TO 590
J = NUMBER(CARD,COL,IDATTP(NUMFIL),ISTART)
J = FIND12(CARD,COL,EQUEVC)
IF(J.NE.2)GO TO 590
ISTART = 0
J = NUMBER(CARD,COL,IDATFL(NUMFIL),ISTART)
GO TO A0
575 J = FIND12(CARD,COL,EQUEVC)
IF(J.NE.2)GO TO 590
J = NUMBER(CARD,COL,IDATFL(NUMFIL),ISTART)
J = FIND12(CARD,COL,EQUEVC)
IF(J.NE.2)GO TO 590
ISTART = 0
J = NUMBER(CARD,COL,IDATTP(NUMFIL),ISTART)
GO TO A0
C
C*** ERRORS
C
500 NUMFIL = NUMFIL - 1
505 WRITE(PRTUNT,1060)
1060 FORMAT(' ERROR ON ABOVE INPUT CONTROL CARD')
GO TO A0
C
C*** LINES IN PSEUDO OPTION
C
600 NOLINE = NUMBER(CARD,COL,LINES,NOLINE)
GO TO A0
C
C*** CHANNELS CARD
C
650 ICHNCT = ICHNCT + 1
IF(ICHNCT.GT.6)GO TO A0
J = NUMBER(CARD,COL,ARRAY(1),ISTART)
IF(J.GT.30)J = 30
DO 660 JJ = 1,J
FETVEC(JJ,ICHNCT) = ARRAY(JJ)
SET01590
SET01600
SET01610
SET01620
SET01630
SET01640
SET01650
SET01660
SET01670
SET01680
SET01690
SET01700
SET01710
SET01720
SET01730
SET01740
SET01750
SET01760
SET01770
SET01780
SET01790
SET01800
SET01810
SET01820
SET01830
SET01840
SET01850
SET01860
SET01870
SET01880
SET01890
SET01900
SET01910
SET01920
SET01930
SET01940
SET01950
SET01960
SET01970
SET01980
SET01990
SET02000
SET02010
SET02020
SET02030
SET02040
SET02050
SET02060
SET02070
SET02080
SET02090
SET02100
SET02110
SET02120
SET02130
SET02140
SET02150
SET02160
SET02170
SET02180
SET02190
SET02200
SET02210
SET02220
SET02230
SET02240
SET02250
SET02260
SET02270
SET02280
SET02290
SET02300
SET02310
SET02320
SET02330
SET02340
SET02350
SET02360
SET02370
```

20-8  
489



FILE: SET1A

```
660 CONTINUE
NFEAT(ICHNCT) = J
GO TO 80
C
C*** *END CARD
C
700 LIMIT = 1
IF(IMOPT.EQ.2)LIMIT = NUMFIL
DO 710 J = 1,LIMIT
ARRAY(2) = 0
JJ = LAREAD(ARRAY(1),ARRAY(3),FLDINF(1,J),ARRAY(2))
JSAVE = J
IF(JJ.NE.1)GO TO 750
710 CONTINUE
JJ = LAREAD(ARRAY(1),ARRAY(3),ARRAY(4),ARRAY(2))
IF(JJ.NE.0)GO TO 750
GO TO 770
C
C*** ERPOR IN FIELD CARDS
C
750 WRITE(PRTUNT,1070)JSAVE
1070 FORMAT(' ERROR IN FIELD',I10,' OR SEND CARD MISSING')
CALL CMERR
C
C*** PROCESS INFORMATION
C
770 CONTINUE
IF(ICHNCT.EQ.NUMFIL)GO TO 775
WRITE(PRTUNT,1080)ICHNCT,NUMFIL
1080 FORMAT(' NUMBER OF CHANNEL CARDS',I5,' DOES NOT MATCH NUMBER OF ',
*DATA FILES',I5)
ICHNCT = NUMFIL
775 IF(ISUNCT.EQ.1.AND.ISUNCT.NE.NUMFIL)GO TO 780
GO TO 785
780 WRITE(PRTUNT,1090)ISUNCT,NUMFIL
1090 FORMAT(' NUMBER OF SUN ANGLE CARDS',I5,' DOES NOT MATCH NUMBER',
*OF DATA FILES',I5)
ISUNCT = NUMFIL
785 IF(IMOPT.EQ.1)GO TO 795
C
C*** CHECK NO. CHANNELS EQUAL ON SPATIAL OR PSEUDO OPTION
C
NOFEAT = NFEAT(1)
IF(NUMFIL.EQ.1) GO TO 795
DO 790 J = 2,NUMFIL
IF(NFEAT(J).EQ.NFEAT(1))GO TO 790
WRITE(PRTUNT,1100)J,NFEAT(J),NFEAT(1)
1100 FORMAT(' NUMBER OF FEATURES OF',I5,' FILE',I5,
*IS NOT EQUAL TO FIRST',I5)
NFEAT(J) = NFEAT(1)
790 CONTINUE
GO TO 806
C
C*** FEATURES IN CHANNEL MERGE
C
795 IDUM = 0
DO 800 I = 1,NUMFIL
IDUM = IDUM + NFEAT(I)
800 CONTINUE
IF(IDUM.LE.30)GO TO 805
WRITE(PRTUNT,1105)(NFEAT(I),I = 1,NUMFIL)
1105 FORMAT(' FEATURES ADD UP TO A NUMBER GREATER THAN 30',6I5,
*EXITING')
CALL CMERR
805 NOFEAT = IDUM
C
C*** SET NOSAMP AND NOLINE
C
805 IF (IMOPT.NE.2) NACROS = 1
NOSAMP = 0
DO 810 J = 1,NACROS
NSS(J) = (FLDINF(5,J) - FLDINF(4,J))/FLDINF(6,J) + 1
NOSAMP = NOSAMP + NSS(J)
810 CONTINUE
NDOWN = NUMFIL / NACROS
IF(IMOPT.NE.2)NDOWN = 1
NOLINE = 0
ICT = 0
NDOWN1 = NDOWN*NACROS
```

SET02390  
SET02390  
SET02400  
SET02410  
SET02420  
SET02430  
SET02440  
SET02450  
SET02460  
SET02470  
SET02480  
SET02490  
SET02500  
SET02510  
SET02520  
SET02530  
SET02540  
SET02550  
SET02560  
SET02570  
SET02580  
SET02590  
SET02600  
SET02610  
SET02620  
SET02630  
SET02640  
SET02650  
SET02660  
SET02670  
SET02680  
SET02690  
SET02700  
SET02710  
SET02720  
SET02730  
SET02740  
SET02750  
SET02760  
SET02770  
SET02780  
SET02790  
SET02800  
SET02810  
SET02820  
SET02830  
SET02840  
SET02850  
SET02860  
SET02870  
SET02880  
SET02890  
SET02900  
SET02910  
SET02920  
SET02930  
SET02940  
SET02950  
SET02960  
SET02970  
SET02980  
SET02990  
SET03000  
SET03010  
SET03020  
SET03030  
SET03040  
SET03050  
SET03060  
SET03070  
SET03080  
SET03090  
SET03100  
SET03110  
SET03120  
SET03130  
SET03140  
SET03150  
SET03160

FILE: SET1A

```
      DO A15 J = 1,NDOWN1,NACROS
      ICT = ICT + 1
      IDUM = (FLDINF(2,J) - FLDINF(1,J))/FLDINF(3,J) + 1
      NOLINE = NOLINE + IDUM
      IF (IMOPT.EQ.2) NLINES(ICT) = IDUM
A15  CONTINUE
      IF(NDOWN.EQ.1)GO TO 830
      NDOWN1 = NDOWN - 1
      DO A20 J = 1,NDOWN1
      DO A20 JJ = 1,NACROS
      ITFMP = JJ + J * NACROS
      NSS(ITFMP) = NSS(JJ)
A20  CONTINUE
C*** SET LINPTR
C
A30  LINPTR(1) = 1
C
C*** STORE FLDINF
C
      IF(IMOPT.EQ.2)GO TO 833
      DO A32 J = 2,6
      DO A32 JJ = 1,6
      FLDINF(JJ,J) = FLDINF(JJ,1)
A32  CONTINUE
A33  IF(IMOPT.NE.3)GO TO 840
      DO A35 J = 1,NUMFIL
      LINPTR(J + 1) = LINPTR(J) + NLINES(J)
A35  CONTINUE
A40  RETURN
      END
```

```
SET03170
SET03180
SET03190
SET03200
SET03210
SET03220
SET03230
SET03240
SET03250
SET03260
SET03270
SET03280
SET03290
SET03300
SET03310
SET03320
SET03330
SET03340
SET03350
SET03360
SET03370
SET03380
SET03390
SET03400
SET03410
SET03420
SET03430
SET03440
SET03450
SET03460
SET03470
```

## 21. GTDDM PROCESSOR

FILE: GTDDM

C	WRITTEN BY C W AHLERS	GTD00010
C	GROUND TRUTH TAPE DUMP ROUTINE	GTD00020
C	SUBROUTINE GTDDM (ARRAY, TOP)	GTD00030
	IMPLICIT INTEGER (A-Z)	GTD00040
	DIMENSION ARWAY (1)	GTD00050
	CALL SET14	GTD00060
	CALL DDM (ARRAY, TOP)	GTD00070
	RETURN	GTD00080
	END	GTD00090
		GTD00100
		GTD00110

FILE: ALPHA

```
FUNCTION ALPHA(S)
IMPLICIT INTEGER (A-Z)
DIMENSION A(26)
DATA A/'A','B','C','D','E','F','G','H','I','J','K','L','M',
      'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
DO 10 I=1,26
ALPHA=I
IF(S.FO.A(I)) RETURN
10 CONTINUE
WRITE(6,20) S
20 FORMAT(1H,5X,'THE SYMBOL ',A1,' CAN NOT BE USED.')
```

ALP00010  
ALP00020  
ALP00030  
ALP00040  
ALP00050  
ALP00060  
ALP00070  
ALP00080  
ALP00090  
ALP00100  
ALP00110  
ALP00120  
ALP00130

~~2x-2~~  
493

ORIGINAL PAGE 1  
OF POOR QUALITY

FILE: ODM

```
C      WRITTEN BY C W AHLERS                                ODM00010
      SUBROUTINE ODM (ARRAY, TOP)                            ODM00020
      IMPLICIT INTEGER (A-Z)                                ODM00030
      DIMENSION ARRAY(1)                                    ODM00040
      DIMENSION OMTX(11,19)                                ODM00050
      DIMENSION FETVEC(30)                                  ODM00060
      DATA FETVEC/30*0/                                     ODM00070
      DIMENSION IDATA(3060)                                ODM00080
      DIMENSION RLOCK(6)                                    ODM00090
      COMMON /TAPEFD/ IUNIT, IFIRST, FSCAN, SAMEND, SAMINC, READY, NSCAN,
      * LINC, ID(200), DSL, LRUF(30), JREC(30), IBYTE(30), NBUFS, FILENO, LINENO
      * LINC, NSAMP, NOCHAN, IFORMT                        ODM00110
      * LININC, NSAMP, NOCHAN, IFORMT                      ODM00120
      * COMMON /GTRK/NRDR, NPRT, PRKEY, VLB(6), GTRDU, GTRDF,
      * GTWRU, GTWRF, GTNOF                                ODM00130
      * GTWRU, GTWRF, GTNOF                                ODM00140
      * GTWRU, GTWRF, GTNOF                                ODM00150
      * GTWRU, GTWRF, GTNOF                                ODM00160
      * GTWRU, GTWRF, GTNOF                                ODM00170
      * GTWRU, GTWRF, GTNOF                                ODM00180
      * GTWRU, GTWRF, GTNOF                                ODM00190
      * GTWRU, GTWRF, GTNOF                                ODM00200
      * GTWRU, GTWRF, GTNOF                                ODM00210
      * GTWRU, GTWRF, GTNOF                                ODM00220
      * GTWRU, GTWRF, GTNOF                                ODM00230
      * GTWRU, GTWRF, GTNOF                                ODM00240
      * GTWRU, GTWRF, GTNOF                                ODM00250
      * GTWRU, GTWRF, GTNOF                                ODM00260
      * GTWRU, GTWRF, GTNOF                                ODM00270
      * GTWRU, GTWRF, GTNOF                                ODM00280
      * GTWRU, GTWRF, GTNOF                                ODM00290
      * GTWRU, GTWRF, GTNOF                                ODM00300
      * GTWRU, GTWRF, GTNOF                                ODM00310
      * GTWRU, GTWRF, GTNOF                                ODM00320
      * GTWRU, GTWRF, GTNOF                                ODM00330
      * GTWRU, GTWRF, GTNOF                                ODM00340
      * GTWRU, GTWRF, GTNOF                                ODM00350
      * GTWRU, GTWRF, GTNOF                                ODM00360
      * GTWRU, GTWRF, GTNOF                                ODM00370
      * GTWRU, GTWRF, GTNOF                                ODM00380
      * GTWRU, GTWRF, GTNOF                                ODM00390
      * GTWRU, GTWRF, GTNOF                                ODM00400
      * GTWRU, GTWRF, GTNOF                                ODM00410
      * GTWRU, GTWRF, GTNOF                                ODM00420
      * GTWRU, GTWRF, GTNOF                                ODM00430
      * GTWRU, GTWRF, GTNOF                                ODM00440
      * GTWRU, GTWRF, GTNOF                                ODM00450
      * GTWRU, GTWRF, GTNOF                                ODM00460
      * GTWRU, GTWRF, GTNOF                                ODM00470
      * GTWRU, GTWRF, GTNOF                                ODM00480
      * GTWRU, GTWRF, GTNOF                                ODM00490
      * GTWRU, GTWRF, GTNOF                                ODM00500
      * GTWRU, GTWRF, GTNOF                                ODM00510
      * GTWRU, GTWRF, GTNOF                                ODM00520
      * GTWRU, GTWRF, GTNOF                                ODM00530
      * GTWRU, GTWRF, GTNOF                                ODM00540
      * GTWRU, GTWRF, GTNOF                                ODM00550
      * GTWRU, GTWRF, GTNOF                                ODM00560
      * GTWRU, GTWRF, GTNOF                                ODM00570
      * GTWRU, GTWRF, GTNOF                                ODM00580
      * GTWRU, GTWRF, GTNOF                                ODM00590
      * GTWRU, GTWRF, GTNOF                                ODM00600
      * GTWRU, GTWRF, GTNOF                                ODM00610
      * GTWRU, GTWRF, GTNOF                                ODM00620
      * GTWRU, GTWRF, GTNOF                                ODM00630
      * GTWRU, GTWRF, GTNOF                                ODM00640
      * GTWRU, GTWRF, GTNOF                                ODM00650
      * GTWRU, GTWRF, GTNOF                                ODM00660
      * GTWRU, GTWRF, GTNOF                                ODM00670
      * GTWRU, GTWRF, GTNOF                                ODM00680
      * GTWRU, GTWRF, GTNOF                                ODM00690
      * GTWRU, GTWRF, GTNOF                                ODM00700
      * GTWRU, GTWRF, GTNOF                                ODM00710
      * GTWRU, GTWRF, GTNOF                                ODM00720
      * GTWRU, GTWRF, GTNOF                                ODM00730
      * GTWRU, GTWRF, GTNOF                                ODM00740
      * GTWRU, GTWRF, GTNOF                                ODM00750
      * GTWRU, GTWRF, GTNOF                                ODM00760
      * GTWRU, GTWRF, GTNOF                                ODM00770
      * GTWRU, GTWRF, GTNOF                                ODM00780
      * GTWRU, GTWRF, GTNOF                                ODM00790
      * GTWRU, GTWRF, GTNOF                                ODM00790

C      EQUIVALENCE (IDATA(1), ARRAY(1))
C      EQUIVALENCE (IDATA(1), ARRAY(3061))
      UNIT=GTWRU
      IUNIT=GTRDU
      FORMT=IFORMT
      NOFF=0
      FILE=0
      NOFFAT=NOCHAN
      FETVEC(1)=1
      DAY=0
      MON=0
      YEAR=0
      SITE=0
      PX=196
      RLOCK(1)=1
      RLOCK(2)=117
      RLOCK(3)=1
      RLOCK(4)=1
      RLOCK(5)=196
      RLOCK(6)=1
10     CONTINUE
      REWIND GTRDU
      REWIND GTWRU
      RDF=GTRDF-1
      WRF=GTWRF-1
C 10    CONTINUE
      NRPOS=1
      MAXREC=3060
      ICOUNT=0
      NREC=0
      FILE=FILE+1
      LINE=0
95     WRITE (NPRT, 951)
      FORMAT (1H1)
      CALL TAPHDR (GTRDU, RDF)
      CALL FSEFEL (GTWRU, WRF, ISTATW)
      WRITE (NPRT, 601) GTRDF, GTWRF
601    FORMAT (//, ' TAPE FILE ', I4, 5X, ' BEING DUMPED TO DOT FILE ', I4)
      DAY=ID(25)
      MON=ID(26)
      YEAR=ID(27)
      SITE=ID(28)
      WRITE (NPRT, 306) SITE, DAY, MON, YEAR
      IF (PRKEY.EQ.1) WRITE (GTWRU, 356) SITE, DAY, MON, YEAR
      IF (PRKEY.EQ.2) WRITE (GTWRU, 366) SITE, DAY, MON, YEAR
      IF (PRKEY.EQ.3) WRITE (GTWRU, 376) SITE, DAY, MON, YEAR
306    FORMAT (' SITE = ', I5, 5X, ' DAY = ', I5, 5X, ' MONTH = ', I5, 5X,
      * ' YEAR = ', I4)
356    FORMAT (' SITE = ', I5, 1X, ' DAY = ', I5, 1X, ' MONTH = ', I5, 1X,
      * ' YEAR = ', I5, ' MASK = TY ', I, ' TYPE = GT ')
366    FORMAT (' SITE = ', I5, 1X, ' DAY = ', I5, 1X, ' MONTH = ', I5, 1X,
      * ' YEAR = ', I5, ' MASK = PHASH 3 ', I, ' TYPE = GT ')
376    FORMAT (' SITE = ', I5, 1X, ' DAY = ', I5, 1X, ' MONTH = ', I5, 1X,
      * ' YEAR = ', I5, ' MASK = INPUT ', I, ' TYPE = GT ')
      WRITE (NPRT, 600)
      CALL FLDIR (RLOCK, FETVEC, NOFFAT)
600    FORMAT (///, ' THE 209 OUT LABELS ')
20    CONTINUE
      IF (NREC.EQ. RLOCK(2)) GO TO 50
      CALL LITEMO (IDATA, ENUTAP)
      IF (ENUTAP.EQ.-1) GO TO 55
      NOFF=0
      NREC=NREC+1
C      IF (NREC.LT.2) WRITE (NPRT, 950) (IDATA(K), K=1, 63)
```

FILE: DDM

```
850  FORMAT(1H,15I4)
      N10=MOD(NREC,10)
C     IF (NREC.EQ.1) WRITE(NPRT,850) (IDATA(K),K=1,GTREC)
      IF (N10.EQ.0) WRITE(NPRT,836)
      • (IDATA(1),I=10,140,10)
836  FORMAT(1H,5X,19I5)
      IF (N10.EQ.0) CALL GTDOTS
      • (IDATA,DMTX,LINE)
839  FORMAT(1H,5X,19I5)
      GO TO 20
50   CONTINUE
      CALL GTDTL(DMTX,NSYM)
      TYPE=PRKEY
      CALL GTDWP(DMTX,TYPE,NSYM)
      WRITE(NPRT,700) NREC
700  FORMAT(///,' NUMBER OF SCAN LINES READ=',I5)
      WRITE(NPRT,750) GTDWF,GTWRF
750  FORMAT(' FILE ',I5,5X,' WAS DUMPED TO FILE ',I5)
      GTDWF=GTWDF+1
      GTWRF=GTWRF+1
      PRF=PRF+1
      VRF=VRF+1
      ENDFILE GTWRU
      IF (FILE.LT.GTNOF) GO TO 10
55   CONTINUE
      REWIND GTWRU
      REWIND GTWRU
      WRITE(NPRT,250)
250  FORMAT(' PROGRAM GTDDM RUN COMPLETED')
      RETURN
      END
```

DDM00800  
DDM00810  
DDM00820  
DDM00830  
DDM00840  
DDM00850  
DDM00860  
DDM00870  
DDM00880  
DDM00890  
DDM00900  
DDM00910  
DDM00920  
DDM00930  
DDM00940  
DDM00950  
DDM00960  
DDM00970  
DDM00980  
DDM00990  
DDM01000  
DDM01010  
DDM01020  
DDM01030  
DDM01040  
DDM01050  
DDM01060  
DDM01070  
DDM01080  
DDM01090  
DDM01100

~~214~~  
495

FILE: GTDOTS

```
SUBROUTINE GTDOTS(/IDATA/.DMTX.LINE)
  IMPLICIT INTEGER (A-Z)
  COMMON /GT-K/NROW,NPRT,PRTKEY,VLB(6),GTRDU,
         GTRIF,GTWHU,GTWPF,GINOF
  DIMENSION DMTX(11,19),IDATA(1)
  WRITE(6,901)
  FORMAT(1H .10X,'GTDOTS')
  LINE=LINE+1
  DO 10 I=10,190,10
    K=I/10
    DMTX(LINE,K)=IDATA(I)
  10 CONTINUE
  RETURN
  END
```

GTD00010  
GTD00020  
GTD00030  
GTD00040  
GTD00050  
GTD00060  
GTD00070  
GTD00080  
GTD00090  
GTD00100  
GTD00110  
GTD00120  
GTD00130  
GTD00140

THIS COPY IS  
PROPERTY

~~215~~  
496

FILE: GTOTL

```
C      WRITTEN C W AHLENS
SUBROUTINE GTOTL(DMTX,NSYM)
      IMPLICIT INTEGER (A-7)
      COMMON /TR/ TRNS1(256),TRNS2(26),TRNS3(26),TY(11,19)
      DATA B/1,1/
      DIMENSION DMTX(11,19)
      DO 5 I=1,26
        TRNS2(I)=B
        TRNS3(I)=B
5      CONTINUE
      DO 10 I=1,11
        DO 10 J=1,19
          GD=DMTX(I,J)
          SYM=TRNS1(GD)
          NUM=ALPHA(SYM)
          TRNS2(NUM)=SYM
          DMTX(I,J)=SYM
10     CONTINUE
C      WRITE(6,40) (TRNS2(I),I=1,26)
60     FORMAT(1H,5X,26A1)
      NSYM=0
      DO 100 I=1,26
        IF (TRNS2(I).NE.B) NSYM=NSYM+1
        IF (TRNS2(I).NE.H) TRNS3(NSYM)=TRNS2(I)
100    CONTINUE
      WRITE(6,110) (TRNS3(I),I=1,26)
110    FORMAT(///,5X,'CATEGORIES FOUND: ',26A1)
      WRITE(6,20)
20     FORMAT(1H,///,5X,'THE 209 DOTS TRANSFORMED')
      DO 40 I=1,11
        WRITE(6,30) (DMTX(I,J),J=1,19)
40     CONTINUE
30     FORMAT(1H,5X,19(4X,A1))
      RETURN
      END
```

GT000010  
GT000020  
GT000030  
GT000040  
GT000050  
GT000060  
GT000070  
GT000080  
GT000090  
GT000100  
GT000110  
GT000120  
GT000130  
GT000140  
GT000150  
GT000160  
GT000170  
GT000180  
GT000190  
GT000200  
GT000210  
GT000220  
GT000230  
GT000240  
GT000250  
GT000260  
GT000270  
GT000280  
GT000290  
GT000300  
GT000310  
GT000320  
GT000330  
GT000340  
GT000350



FILE GTDWR

```

C      WRITTEN BY C W AHLERS
C      THIS PROGRAM WRITES LACIE FORMAT DOT FILES
C      GIVEN A MATRIX OF DOT LABELS & A TYPE MASK MATRIX
      SUBROUTINE GTDWR(DMTX,TYPE,NSYM)
      IMPLICIT INTEGER (A-Z)
      DIMENSION DMTX(11,19)
      DIMENSION DWR(15)
      COMMON /TR/TRNS1(256),TRNS2(26),TRNS3(26),TY(11,19)
      COMMON /GT4K/NRDH,NPRT,PRTKEY,VLB(6),GTRDU,GTRDF,
      * GTWRU,GTWRF,GTNOF
      WRITE(NPRT,11)
11     FORMAT(1H, '//,10X,'LACIE FORMAT DOT LABELS')
      WRITE(NPRT,111) GTWRU
111    FORMAT(1H,10X,'WRITTEN TO UNIT ',I5)
      IF (TYPE.EQ.1) WRITE(NPRT,12)
      IF (TYPE.EQ.2) WRITE(NPRT,13)
      IF (TYPE.EQ.3) WRITE(NPRT,14)
12     FORMAT('//,10X,'TYPES BASED ON A TRANSITION YEAR MASK')
13     FORMAT('//,10X,'TYPES BASED ON A PHASE THREE MASK')
14     FORMAT('//,10X,'TYPES BASED ON AN INPUT MASK')
      NTYPES=2
      DO 30 TT=1,NTYPES
      DO 30 SS=1,NSYM
      COUNT=0
      DO 25 I=1,11
      DO 25 J=1,19
      IJ=J+19*(I-1)
      IF (TY(I,J).EQ.TT.AND.DMTX(I,J).EQ.TRNS3(SS)) COUNT=COUNT+1
      IF (TY(I,J).EQ.TT.AND.DMTX(I,J).EQ.TRNS3(SS)) DWR(COUNT)=IJ
      IF (COUNT.EQ.15) WRITE(NPRT,33) TT,TRNS3(SS),(DWR(K),K=1,COUNT)
      IF (COUNT.EQ.15) WRITE(GTWRU,43) TT,TRNS3(SS),(DWR(K),K=1,COUNT)
      IF (COUNT.EQ.15) COUNT=0
25     CONTINUE
      IF (COUNT.GT.0) WRITE(NPRT,33) TT,TRNS3(SS),(DWR(K),K=1,COUNT)
      IF (COUNT.GT.0) WRITE(GTWRU,43) TT,TRNS3(SS),(DWR(K),K=1,COUNT)
33     FORMAT(1H,5X,'DOT ',I1,2X,A1,3X,15I4)
43     FORMAT('DOT ',I1,2X,A1,3X,15I4)
30     CONTINUE
      WRITE(GTWRU,53)
53     FORMAT('*$END*')
      RETURN
      END
      GTD00010
      GTD00020
      GTD00030
      GTD00040
      GTD00050
      GTD00060
      GTD00070
      GTD00080
      GTD00090
      GTD00100
      GTD00110
      GTD00120
      GTD00130
      GTD00140
      GTD00150
      GTD00160
      GTD00170
      GTD00180
      GTD00190
      GTD00200
      GTD00210
      GTD00220
      GTD00230
      GTD00240
      GTD00250
      GTD00260
      GTD00270
      GTD00280
      GTD00290
      GTD00300
      GTD00310
      GTD00320
      GTD00330
      GTD00340
      GTD00350
      GTD00360
      GTD00370
      GTD00380
      GTD00390
      GTD00400
      GTD00410
      GTD00420

```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: GTRNS

```
SUBROUTINE GTRNS"  
  IMPLICIT INTEGER (A-Z)  
  DATA W,S,R,F,O,N/IW,IS,IB,IR,IF,IO,IN/  
  COMMON /TR/ TRNS1(256), TRNS2(256), TRNS3(256)  
  DO 10 I=1,256  
    TRNS1(I)=N  
10  CONTINUE  
    TRNS1(99)=W  
    TRNS1(124)=W  
    TRNS1(100)=S  
    TRNS1(125)=S  
    TRNS1(101)=R  
    TRNS1(126)=R  
    TRNS1(102)=R  
    TRNS1(127)=R  
    TRNS1(103)=F  
    TRNS1(128)=F  
    TRNS1(104)=O  
    TRNS1(129)=O  
  DO 20 I=1,15  
    II=I+15  
    TRNS1(I)=W  
    TRNS1(II)=S  
20  CONTINUE  
  RETURN  
  END
```

GTT00010  
GTT00020  
GTT00030  
GTT00040  
GTT00050  
GTT00060  
GTT00070  
GTT00080  
GTT00090  
GTT00100  
GTT00110  
GTT00120  
GTT00130  
GTT00140  
GTT00150  
GTT00160  
GTT00170  
GTT00180  
GTT00190  
GTT00200  
GTT00210  
GTT00220  
GTT00230  
GTT00240  
GTT00250  
GTT00260

FILE: SET19

ORIGINAL PART I  
OF POOR QUALITY

```

C      WRITTEN BY C W AHLERS
SUBROUTINE SET19
IMPLICIT INTEGER (A-Z)
DIMENSION CODE(10),CARD(62),EQUOM(3),ACARD(20)
DIMENSION SLASH(2)
DATA SLASH /1,'/'/
DATA CODE/'TRAN','READ','WRITE',
* 'MASK','DATE','COMM','HED1','HED2','END','DUMP'/
DATA EQUOM/2,'='/'/
DATA V/'V','PLNK'/'/'/'U'/'U'/'FF'/'F'/'00'/'0'/'P'/'P'/'
DATA T/'T'/'
DATA QQ/'?'/'A,II'/'I'/'
C      INCLUDE COMRK1.LIST
C      INCLUDE COMRK4.LIST
C      INCLUDE COMRK6.LIST
C      INCLUDE COMRK14.LIST
COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
* AVAR2,CVAR2,CLSID2,SURN02,SUPDS2,FLDSV2,VERTX2,
* FETVC2(30),SUFVC2(75),SHPTR(75),CLSV2(60),
* KEPTS(60),NOGRP,GKPNAM(60),GRPOEX(61),
* GRCHK(61),GROUPS(124)
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
COMMON/GLOBAL/HEAD(63),MAPTAP,DATEP,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERITP,ERPKEY,MAPUNT,NOFILF,
* DRUMAD,DRMND5,PAGSIZ,DATEFIL,STAFIL,ASAV,ASAVFL
* .NHSTUN,NHSTFI,SCRUN,MAFFIL
* .DOTUNT,DOTFIL,NCHPAS,TRNSFL,RMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
COMMON /GTRK/NRDR,NPRT,MSKKEY,VLH(6),GTRDU,GTRDF,
* GTWRU,GTWRF,GTNOF
COMMON /TAPERD/IUNIT,IFRST,FCSCAN,SAMEND,SAMINC,READY,NSCAN,
* LINC.ID(200),DSL,LRUF(30),JREC(30),IHYE(30),NBUFS,FILENO,LINEND
* .LININC,NSAMP,NOCHAN,IFORMT
COMMON /TR/TRNS1(256),TRNS2(26),TRNS3(26),TY(11,19)
C$END
DIMENSION PH31(19),PH32(19),PH33(19),PH34(19),PH35(19)
DIMENSION PH36(19),PH37(19),PH38(19),PH39(19),PH310(19)
DIMENSION PH311(19)
DATA PH31/5*1.2,2.1,1.2,2.2,8*1/
DATA PH32/1.2,1.2,1.2,1.2,2.1,2.2,1.2,1.2,1.1,2.1,1.2,1/
DATA PH33/1.2,2.3*1.2,1.2,3*1.3*2.1,2.1,2/
DATA PH34/2.2,1.1,2.1,2.3*1.2,1.2,2.1,2.2,2/
DATA PH35/1.2,2.3*1.3*2.1,1.2,1.2,2.3*1/
DATA PH36/1.2,2.1,4*2.1,1.4*2.1,2.1,2/
DATA PH37/1.2,2.1,2.3*1.2,2.1,2.1,3*2.1,1.1,2/
DATA PH38/3*1.1,4*2.1/
DATA PH39/1.2,1.2,1.1,2.2,1.1,2.1,1.2,1.4*2.1/
DATA PH310/4*1.4*2.1,1.2,3*1.2,1.1,2.2/
DATA PH311/1.2,1.1,2.5*1.2,1.1,2.2,3*1.2/
DIMENSION NVEC(30)
C
7FPO = 0
GOOF = 0
GTRDU = 12
GTWRU = 23
GTRDF = 1
GTWRF = 1
GTNOF = 1
LINE = 0
NPRT = PRUNT
NPDR = CRDUNT
NOLAP = 0
MSKKEY = 1
NPUT = 10
GTNOF = 1
NOCHAN = 1
IFORMT = 1
DO 50 I=1,256
50 TRNS1(I)=00
C
DO 55 I=1,11
DO 55 J=1,19
55 TY(I,J)=-100
CONTINUE
WRITE(NPRT,100)

```

```

SET00010
SET00020
SET00030
SET00040
SET00050
SET00060
SET00070
SET00080
SET00090
SET00100
SET00110
SET00120
SET00130
SET00140
SET00150
SET00160
SET00170
SET00180
SET00190
SET00200
SET00210
SET00220
SET00230
SET00240
SET00250
SET00260
SET00270
SET00280
SET00290
SET00300
SET00310
SET00320
SET00330
SET00340
SET00350
SET00360
SET00370
SET00380
SET00390
SET00400
SET00410
SET00420
SET00430
SET00440
SET00450
SET00460
SET00470
SET00480
SET00490
SET00500
SET00510
SET00520
SET00530
SET00540
SET00550
SET00560
SET00570
SET00580
SET00590
SET00600
SET00610
SET00620
SET00630
SET00640
SET00650
SET00660
SET00670
SET00680
SET00690
SET00700
SET00710
SET00720
SET00730
SET00740
SET00750
SET00760
SET00770
SET00780
SET00790

```

FILE: SET19

```
100 FORMAT(/11X,'INPUT SUMMARY'//)
C
C   SET UP REREAD BUFFER
C
C   RRUNIT = 30
C   CALL REREAD(RRUNIT,80)
C
C   PUT CARD IN RUFFER
105 READ(NRDR,103)(ACARD(I),I=1,20)
103 FORMAT(20A4)
WRITE(RRUNIT,103)(ACARD(I),I=1,20)
REWIND RRUNIT
C
C   READ(RRUNIT,110)CODE1,CARD
C   REWIND RRUNIT
C   COL = 0
C   WRITE(NPRT,120)CODE1,CARD
120 FORMAT(1X,A4,6X,62A1)
110 FORMAT(A4,6X,62A1)
DO 130 I=1,NPRT
IF (CODE1.EQ.CODE(I)) GO TO (150,180,210,330,370,
* 390,400,410,420,500),I
130 CONTINUE
135 WRITE(NPRT,140)
140 FORMAT(' INVALID CONTROL CARD - IGNORED ')
GO TO 105
C
C   TRANSFORMATION
C
150 M = NIXCHR(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
GO TO 155
152 WRITE(NPRT,153)
153 FORMAT(' ERROR ON TRANSFORMATION CARDS')
GO TO 105
155 J = FIND12(CARD,COL,EQUOM)
IF (J.NE.2) GO TO 152
NOLAB=0
NOLAB = NUMBER(CARD,COL,VLR,NOLAB)
COL = COL - 1
IF (NOLAB.GT.2) GO TO 152
IF (VLR(1).GT.VLR(2)) GO TO 152
IF (VLR(2).GT.256) GO TO 152
NR=VLR(1)
NE=VLR(2)
DO 156 I=NR,NE
TRANS(I)=M
156 CONTINUE
GO TO 105
C
C   READ TAPE
C
180 M = NIXCHR(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
IF (M.EQ.U) GO TO 190
IF (M.EQ.FF) GO TO 200
185 WRITE(NPRT,187)
187 FORMAT(' ERROR ON READ TAPE CARD')
GO TO 105
190 J = FIND12(CARD,COL,EQUOM)
IF (J.NE.2) GO TO 185
M = NUMBER(CARD,COL,GTRDU,ZERO)
COL = COL - 1
GO TO 180
200 J = FIND12(CARD,COL,EQUOM)
IF (J.NE.2) GO TO 185
M = NUMBER(CARD,COL,GTRDF,ZERO)
COL = COL - 1
GO TO 180
C
C   WRITE FILE
C
210 CONTINUE
214 M = NIXCHR(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
IF (M.EQ.U) GO TO 230
IF (M.EQ.FF) GO TO 240
215 WRITE(NPRT,220)
```

SET00800  
SET00810  
SET00820  
SET00830  
SET00840  
SET00850  
SET00860  
SET00870  
SET00880  
SET00890  
SET00900  
SET00910  
SET00920  
SET00930  
SET00940  
SET00950  
SET00960  
SET00970  
SET00980  
SET00990  
SET01000  
SET01010  
SET01020  
SET01030  
SET01040  
SET01050  
SET01060  
SET01070  
SET01080  
SET01090  
SET01100  
SET01110  
SET01120  
SET01130  
SET01140  
SET01150  
SET01160  
SET01170  
SET01180  
SET01190  
SET01200  
SET01210  
SET01220  
SET01230  
SET01240  
SET01250  
SET01260  
SET01270  
SET01280  
SET01290  
SET01300  
SET01310  
SET01320  
SET01330  
SET01340  
SET01350  
SET01360  
SET01370  
SET01380  
SET01390  
SET01400  
SET01410  
SET01420  
SET01430  
SET01440  
SET01450  
SET01460  
SET01470  
SET01480  
SET01490  
SET01500  
SET01510  
SET01520  
SET01530  
SET01540  
SET01550  
SET01560  
SET01570  
SET01580

FILE: SET19

ORIGINAL PAGE 1  
OF POOR QUALITY

220	FORMAT(' ERROR ON WRITE FILE CARD')	SET01590
	GO TO 105	SET01600
230	J = FIND12(CARD,COL,EDUCOM)	SET01610
	IF (J.NF.2) GO TO 215	SET01620
	M = NUMBER(CARD,COL,GTWRU,ZERO)	SET01630
	COL = COL - 1	SET01640
	GO TO 214	SET01650
240	J = FIND12(CARD,COL,EDUCOM)	SET01660
	IF (J.NF.2) GO TO 215	SET01670
	M = NUMBER(CARD,COL,GTWRF,ZERO)	SET01680
	COL = COL - 1	SET01690
	GO TO 214	SET01700
	MASK CARD	SET01710
		SET01720
		SET01730
330	M = NXTCHR(CARD,COL)	SET01740
	IF (M.EQ.'BLNK') GO TO 105	SET01750
	IF (M.EQ.'P') GO TO 340	SET01760
	IF (M.EQ.'T') GO TO 341	SET01770
	IF (M.EQ.'I') GO TO 342	SET01780
333	WRITE(PRINT,335)	SET01790
335	FORMAT(' ERROR ON MASK CARD --- TRANSITION YEAR MASK USED')	SET01800
	GO TO 341	SET01810
340	MSKKEY = 2	SET01820
	DO 20 J=1,10	SET01830
	TY(1,J)=PH31(J)	SET01840
	TY(2,J)=PH32(J)	SET01850
	TY(3,J)=PH33(J)	SET01860
	TY(4,J)=PH34(J)	SET01870
	TY(5,J)=PH35(J)	SET01880
	TY(6,J)=PH36(J)	SET01890
	TY(7,J)=PH37(J)	SET01900
	TY(8,J)=PH38(J)	SET01910
	TY(9,J)=PH39(J)	SET01920
	TY(10,J)=PH310(J)	SET01930
	TY(11,J)=PH311(J)	SET01940
20	CONTINUE	SET01950
	GO TO 105	SET01960
341	MSKKEY=1	SET01970
	DO 10 I=1,11	SET01980
	DO 10 J=1,10	SET01990
	IJ=J+10*(I-1)	SET02000
	TY(I,J)=1	SET02010
	IF (IJ/2.EQ.IJ) TY(I,J)=2	SET02020
10	CONTINUE	SET02030
	IF (GOOF.EQ.1) GO TO 461	SET02040
	GO TO 105	SET02050
342	MSKKEY=3	SET02060
	LINE=LINE+1	SET02070
	IF (LINE.GT.11) GO TO 333	SET02080
	J=FIND12(CARD,COL,EDUCOM)	SET02090
	IF (J.NF.2) GO TO 333	SET02100
	NOMSK=0	SET02110
	NOMSK=NUMBER(CARD,COL,MVEC,NOMSK)	SET02120
	IF (NOMSK.GT.19) GOOF=1	SET02130
	IF (NOMSK.GT.19) GO TO 105	SET02140
	DO 345 J=1,NOMSK	SET02150
	TY(LINE,J)=MVEC(J)	SET02160
345	CONTINUE	SET02170
	GO TO 105	SET02180
	DATE CARD	SET02190
		SET02200
		SET02210
370	M = NXTCHR(CARD,COL)	SET02220
	IF (M.EQ.'BLNK') GO TO 105	SET02230
	READ(PRUNIT,340)DATE	SET02240
380	FORMAT(10X,15A4)	SET02250
	REWIND PRUNIT	SET02260
	GO TO 105	SET02270
	COMMENT CARD	SET02280
		SET02290
		SET02300
390	M = NXTCHR(CARD,COL)	SET02310
	IF (M.EQ.'BLNK') GO TO 105	SET02320
	READ(PRUNIT,380)COMENT	SET02330
	REWIND PRUNIT	SET02340
	GO TO 105	SET02350
		SET02360
		SET02370

FILE: SET19

```
C      MED1
C
400  M = NXTCHR(CARD,COL)
      READ(RRUNIT,380) MED1
      REWIND RRUNIT
      GO TO 105
C
C      MED2
C
410  M = NXTCHR(CARD,COL)
      READ(RRUNIT,380) MED2
      REWIND RRUNIT
      GO TO 105
C
C      CONVERT CARD
C
500  M=NXTCHR(CARD,COL)
      IF(M.EQ.BLNK) GO TO 105
      IF(M.EQ.FF) GO TO 510
530  WRITE(NPRT,520)
520  FORMAT(' ERROR ON CONVERT CARD')
      GO TO 105
510  J=FNDD12(CARD,COL,EQUOM)

      M=NUMBER(CARD,COL,GTNOF,ZERO)
      COL=COL-1
      GO TO 500
C
C      *END*
C
420  CONTINUE
      DO 430 I=1,256
      IF(TRNS1(I).EQ.00) GO TO 450
430  CONTINUE
      GO TO 440
450  WRITE(NPRT,451)
      WRITE(NPRT,451)
451  FORMAT(//,10X,'DEFAULT TRANSFORMATION USED')
      CALL GTRNS
C
440  CONTINUE
C
      IF(GOOF.EQ.1) GO TO 333
      DO 460 I=1,11
      DO 460 J=1,19
      IF(TY(I,J).EQ.-100) GOOF=1
      IF(GOOF.EQ.1) GO TO 333
460  CONTINUE
461  CONTINUE
C
      WRITE(NPRT,1000)
      WRITE(NPRT,1100) GTRDU,GTRDF
      WRITE(NPRT,1200) GTWRU,GTWRF
      WRITE(NPRT,1300) GTNOF
      IF (MSKKEY .EQ. 1) WRITE(NPRT,1030)
      IF (MSKKEY .EQ. 2) WRITE(NPRT,1031)
      IF (MSKKEY .EQ. 3) WRITE(NPRT,1032)
1000  FORMAT(//, 'USER HAS REQUESTED THE FOLLOWING OPTIONS ://')
1030  FORMAT(' TRANSITION YEAR MASK')
1031  FORMAT(' PHASE THREE MASK')
1032  FORMAT(' INPUT MASK')
1100  FORMAT(' READ UNIT = ',I3,' PEAD FILE = ',I3)
1200  FORMAT(' WRITE UNIT = ',I3,' WRITE FILE = ',I3)
1300  FORMAT(' THE NUMBER OF TAPE FILES TO BE DUMPED TO DOT FILES = ',
      * I4)
C
630  WRITE(NPRT,630)
      FORMAT(//, ' THE CROP CODE TO SYMBOL TRANSFORMATION')
      FIRST=1
      SYM=TRNS1(1)
      DO 631 I=1,256
      LAST=I-1
      IF (TRNS1(I).NE.SYM) WRITE(NPRT,632) SYM,FIRST, LAST
      IF (TRNS1(I).NE.SYM) FIRST=I
      IF (TRNS1(I).NE.SYM) SYM=TRNS1(I)
      IF (I.EQ.256) WRITE(NPRT,632) SYM,FIRST,I
631  CONTINUE
632  FORMAT(1H,5X,A1,' = ',I3,' , ',I3)
      WRITE(NPRT,640)
```

SET02380  
SET02390  
SET02400  
SET02410  
SET02420  
SET02430  
SET02440  
SET02450  
SET02460  
SET02470  
SET02480  
SET02490  
SET02500  
SET02510  
SET02520  
SET02530  
SET02540  
SET02550  
SET02560  
SET02570  
SET02580  
SET02590  
SET02600  
SET02610  
SET02620  
SET02630  
SET02640  
SET02650  
SET02660  
SET02670  
SET02680  
SET02690  
SET02700  
SET02710  
SET02720  
SET02730  
SET02740  
SET02750  
SET02760  
SET02770  
SET02780  
SET02790  
SET02800  
SET02810  
SET02820  
SET02830  
SET02840  
SET02850  
SET02860  
SET02870  
SET02880  
SET02890  
SET02900  
SET02910  
SET02920  
SET02930  
SET02940  
SET02950  
SET02960  
SET02970  
SET02980  
SET02990  
SET03000  
SET03010  
SET03020  
SET03030  
SET03040  
SET03050  
SET03060  
SET03070  
SET03080  
SET03090  
SET03100  
SET03110  
SET03120  
SET03130  
SET03140  
SET03150  
SET03160

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: SET19

```
640  FORMAT(//, ' THE MASK')  
      DO 641 I=1,11  
      WRITE(NPWT,642) (TY(I,J),J=1,19)  
641  CONTINUE  
642  FORMAT(1H, '5X,19I5)  
C    RETURN  
      END
```

SET03170  
SET03180  
SET03190  
SET03200  
SET03210  
SET03220  
SET03230  
SET03240

## 22. GTTCN PROCESSOR

FILE: GTTCN

C  
C  
C

```
      WRITTEN BY C W AHLERS  
      GROUND TRUTH TAPE CONVERSION ROUTINE  
      SUBROUTINE GTTCN(AWAY, TOP)  
      IMPLICIT INTEGER (A-Z)  
      DIMENSION AWAY(1)  
      CALL SET17  
      CALL TCN(AWAY, TOP)  
      RETURN  
      END
```

```
GTTCN010  
GTTCN020  
GTTCN030  
GTTCN040  
GTTCN050  
GTTCN060  
GTTCN070  
GTTCN080  
GTTCN090  
GTTCN100  
GTTCN110
```



FILE: GTCRPL

```
C      WRITTEN BY C W AHLERS
SUBROUTINE GTCRPL(CROP,MT,NC)
IMPLICIT INTEGER (A-Z)
DIMENSION MT(K)
COMMON /GTRP/NROP,NPRT,PRTKEY,VLB(6),GTRDU,GTRDF
      ,GTRDU,GTRDF,GTRNOF
C 901 WRITE(NPRT,901)
      FORMAT(1H,10X,'GTCRPL')
      NC=N
      DO 10 I=1,6
      IF (VLR(I).LT.1) GO TO 10
      CC=MT(I)
      N=N
      DO 20 J=1,6
      IF (VLR(J).LT.1) GO TO 20
      IF (CC.EQ.MT(J)) N=N+1
20    CONTINUE
      IF (N.LE.NC) GO TO 10
      NC=N
      CROP=CC
10    IF (NC.GE.3) RETURN
      CONTINUE
      RETURN
      END
```

GTC00010  
GTC00020  
GTC00030  
GTC00040  
GTC00050  
GTC00060  
GTC00070  
GTC00080  
GTC00090  
GTC00100  
GTC00110  
GTC00120  
GTC00130  
GTC00140  
GTC00150  
GTC00160  
GTC00170  
GTC00180  
GTC00190  
GTC00200  
GTC00210  
GTC00220  
GTC00230  
GTC00240

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: GTUNPK

	SUBROUTINE GTUNPK(IDATA,N1,N2,OS,NRPDS,LENGTH)	GTU00010
	IMPLICIT INTEGER (A-Z)	GTU00020
	DIMENSION IDATA(1)	GTU00030
C	WRITE(3,900)	GTU00040
	WRITE(6,900)	GTU00050
900	FORMAT(1H,10X,'GTUNPK')	GTU00060
	DO 10 I=1,NRPDS	GTU00070
	DO 20 J=N1,N2	GTU00080
	JJ=J+OS	GTU00090
	IF(JJ.GT.3040) WRITE(6,901) JJ	GTU00100
	IF(JJ.GT.3040) STOP	GTU00110
901	FORMAT(1H,10X,'JJ=',110)	GTU00120
	I=IDATA(JJ)	GTU00130
	IF(I1.GT.128) IDATA(JJ)=IDATA(JJ)-128	GTU00140
	IF(I1.LE.128) IDATA(JJ)=128+IDATA(JJ)	GTU00150
20	CONTINUE	GTU00160
	OS=OS+LENGTH	GTU00170
10	CONTINUE	GTU00180
	RETURN	GTU00190
	END	GTU00200

FILE: LINLAB

```
SUBROUTINE LINLAB (/IDATA/,MAXREC)
IMPLICIT INTEGER (A-Z)
DIMENSION MT(5),IDATA(3060)
C 901 WRITE(3,901)
      FORMAT(1H,10X,'LINLAB')
      OS=72
      DO 30 PIX=1,196
      OS=OS+2
      M=0
      DO 40 LL=1,3
      DO 50 SS=1,2
      M=M+1
      SSS=OS+SS-2
      IF (SSS.GT.3060) WRITE(6,903) SSS
      IF (SSS.GT.3060) STOP
      C 907 WRITE(3,907) SSS
      FORMAT(1H,10X,'SSS=',I10)
      IF (M.GT.6) WRITE(6,911) M
      IF (M.GT.6) STOP
      911 FORMAT(1H,10X,'M=',I5)
      MT(M)=IDATA(SSS)
      50 CONTINUE
      OS=OS+MAXREC
      40 CONTINUE
      CALL GTCRPL(CROP,MT,NCROP)
      C PP=PIX*72
      PP=PIX
      IDATA(PP)=CROP
      OS=72+2*PIX
      30 CONTINUE
      DO 60 I=260,3060
      IDATA(I)=0
      C 60 WRITE(6,902) (IDATA(K),K=1,196)
      902 FORMAT(1H,2516)
      RETURN
      END
```

L N00010  
L N00020  
L N00030  
L N00040  
L N00050  
L N00060  
L N00070  
L N00080  
L N00090  
L N00100  
L N00110  
L N00120  
L N00130  
L N00140  
L N00150  
L N00160  
L N00170  
L N00180  
L N00190  
L N00200  
L N00210  
L N00220  
L N00230  
L N00240  
L N00250  
L N00260  
L N00270  
L N00280  
L N00290  
L N00300  
L N00310  
L N00320  
L N00330  
L N00340  
L N00350  
L N00360

22-4  
508

ORIGINAL PAGE IS  
OF POOR QUALITY.

FILE SET17

```

C      WRITER BY C. W. AHLERS
SUBROUTINE SET17
IMPLICIT INTEGER (A-Z)
DIMENSION CODE(10),CARD(62),EQUOM(3),ACARD(20)
DIMENSION SLASH(2)
DATA SLASH /1,0//
DATA CODE /1,AH,1,HEAD,WHIT,
* OPTI,DATE,COMMS,MED1,MED2,END,CONV//
DATA EQUOM /2,1,1,1,1//
DATA VVVV /2,CLNK,1,1,U,1,1,FF,1,1,00,1,0,P,1,1//
C
C
C      INCLUDE COMMON1.LIST
C      INCLUDE COMMON4.LIST
C      INCLUDE COMMON5.LIST
C      INCLUDE COMMON10.LIST
COMMON/INFORM/NOCLS2,NUSUR2,NOFET2,VAR52,TOTVT2,NOFLD2,
* AVAR2,COVAR2,CLS102,SUHRN2,SUHRD52,FLUSV2,VERTX2,
* FE1VC2(30),SUHVC2(75),SUHPTX(75),CLSV2(60),
* KEPTS(60),NOGHP,GHPNAM(60),GRPDER(61),
* GRPCHK(61),GROUPS(124)
DIMENSION MED1(15),MED2(15),DATE(3),COMENT(15)
EQUIVALENCE (MED1(1),HEAD(4)),(DATE(1),HEAD(22)),
(MED2(1),HEAD(30)),(COMENT(1),HEAD(48))
2 COMMON/GLOBAL/HEAD(63),MARTAP,DATEP,SAVTAP,RMFILE,RMKEY,
* HISTFL,HISKEY,INFORM,ENLPTP,ERRKEY,MAPUNT,NOFILF,
* DRHMAD,DRHMOS,PAGSIZ,DATEFL,STAFIL,ASAV,ASAVFL
* .NHSTUN,NHSTFL,SCHEUN,MAPFIL
* .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* .CDUNT,PRUNT,HANDIO
COMMON /GTRK/NRDR,NPRT,PRTKY,VLR(6),GTRDU,GTRDF,
* GTRHU,GTRHF,GTRNF
COMMON /TAPHO/IUNIT,IFIRST,FSCAN,SAMEND,SAMINC,READY,NSCAN,
* LIAC,IO(200),OSL,LHUF(30),JREC(30),IBYTE(30),NHUFS,FILENO,LINFNO
* LININC,NSAMP,NOCHAN,IFORMT
CSEND
7FRD = 0
NPRT=PRUNT
NRDR=CDUNT
GTRDU=1
GTRHU=12
GTRHF=1
GTRNF=1
GTRDF=1
NOCHAN = 0
PRTKY = 0
NPRT=10
GTRNF=1
NOCHAN=1
IFORMT=1
C
C      WRITE(NPRT,100)
100 FORMAT(/11A,'INPUT SUMMARY'//)
C
C      SET UP REHEAD BUFFER
C
C      PRINT = 30
CALL REHEAD(NIUNIT,RO)
C
C      PUT CARD IN BUFFER
105 READ(NRDR,103)(ACARD(I),I=1,20)
103 FORMAT(20A4)
WRITE(NPRT,103)(ACARD(I),I=1,20)
REWIND NPRT
C
C      READ(NRDR,110)CODE1,CARD
REWIND NIUNIT
CODE = 0
WRITE(NPRT,120)CODE1,CARD
120 FORMAT(1X,A4,02,02A1)
110 FORMAT(A4,02,02A1)
DO 130 I=1,PRINT
IF (CODE1.EQ.CODE(I)) GO TO (150,160,210,330,370,
* 390,400,410,420,500),I
130 CONTINUE
125 WRITE(NPRT,140)
140 FORMAT(' INVALID CONTROL CARD - IGNORED ')
GO TO 105
C
C      LABEL VECTOR

```

SF 100010  
 SF 100020  
 SF 100030  
 SF 100040  
 SF 100050  
 SF 100060  
 SF 100070  
 SF 100080  
 SF 100090  
 SF 100100  
 SF 100110  
 SF 100120  
 SF 100130  
 SF 100140  
 SF 100150  
 SF 100160  
 SF 100170  
 SF 100180  
 SF 100190  
 SF 100200  
 SF 100210  
 SF 100220  
 SF 100230  
 SF 100240  
 SF 100250  
 SF 100260  
 SF 100270  
 SF 100280  
 SF 100290  
 SF 100300  
 SF 100310  
 SF 100320  
 SF 100330  
 SF 100340  
 SF 100350  
 SF 100360  
 SF 100370  
 SF 100380  
 SF 100390  
 SF 100400  
 SF 100410  
 SF 100420  
 SF 100430  
 SF 100440  
 SF 100450  
 SF 100460  
 SF 100470  
 SF 100480  
 SF 100490  
 SF 100500  
 SF 100510  
 SF 100520  
 SF 100530  
 SF 100540  
 SF 100550  
 SF 100560  
 SF 100570  
 SF 100580  
 SF 100590  
 SF 100600  
 SF 100610  
 SF 100620  
 SF 100630  
 SF 100640  
 SF 100650  
 SF 100660  
 SF 100670  
 SF 100680  
 SF 100690  
 SF 100700  
 SF 100710  
 SF 100720  
 SF 100730  
 SF 100740  
 SF 100750  
 SF 100760  
 SF 100770  
 SF 100780  
 SF 100790

FILE: SET17

```

C
150 A = NATCHW(CARD,COL)
    IF (M.FO.V) GO TO 155
    IF (M.FO.HLNR) GO TO 105
152 WRITE(MUNIT,153)
153 FORMAT(' ERROR ON LABEL CARD')
    GO TO 105
155 J = FIND12(CARD,COL,EDUCOM)
    IF (J.NF.2) GO TO 152
    NO LAB = NUMRFK(CARD,COL.VLR,NOLAB)
    GO TO 105

```

C

READ TAPE

```

180 M = NATCHW(CARD,COL)
    IF (M.FO.HLNR) GO TO 105
    IF (M.FO.U) GO TO 190
    IF (M.FO.EF) GO TO 200
185 WRITE(MUNIT,187)
187 FORMAT(' ERROR ON HEAD TAPE CARD')
    GO TO 105
190 J = FIND12(CARD,COL,EDUCOM)
    IF (J.NF.2) GO TO 185
    M = NUMBER(CARD,COL.GTWDU,ZERO)
    COL = COL - 1
    GO TO 180
200 J = FIND12(CARD,COL,EDUCOM)
    IF (J.NF.2) GO TO 185
    M = NUMBER(CARD,COL.GTRDF,ZERO)
    COL = COL - 1
    GO TO 180

```

C

WRITE TAPE

```

210 CONTINUE
214 M = NATCHW(CARD,COL)
    IF (M.FO.HLNR) GO TO 105
    IF (M.FO.U) GO TO 230
    IF (M.FO.EF) GO TO 240
215 WRITE(MUNIT,220)
220 FORMAT(' ERROR ON WRITE TAPE CARD')
    GO TO 105
230 J = FIND12(CARD,COL,EDUCOM)
    IF (J.NF.2) GO TO 215
    M = NUMBER(CARD,COL.GTWDU,ZERO)
    COL = COL - 1
    GO TO 214
240 J = FIND12(CARD,COL,EDUCOM)
    IF (J.NF.2) GO TO 215
    M = NUMBER(CARD,COL.GTWFZ,ZERO)
    COL = COL - 1
    GO TO 214

```

C

OPTION CARD

```

330 M = NATCHW(CARD,COL)
    IF (M.FO.FLNR) GO TO 105
    IF (M.FO.M) GO TO 340
333 WRITE(MUNIT,335)
335 FORMAT(' ERROR ON OPTION CARD')
    GO TO 105
340 PRTKBY = 1
    GO TO 105

```

C

DATE CARD

```

370 M = NATCHW(CARD,COL)
    IF (M.FO.FLNR) GO TO 105
    READ(MUNIT,380) DATE
380 FORMAT(10X,15A4)
    DEFINE MUNIT
    GO TO 105

```

C

COMMENT CARD

```

390 M = NATCHW(CARD,COL)
    IF (M.FO.FLNR) GO TO 105
    READ(MUNIT,395) COMMENT
    DEFINE MUNIT

```

SF 100400  
 SF 100410  
 SF 100420  
 SF 100430  
 SF 100440  
 SF 100450  
 SF 100460  
 SF 100470  
 SF 100480  
 SF 100490  
 SF 100900  
 SF 100910  
 SF 100920  
 SF 100930  
 SF 100940  
 SF 100950  
 SF 100960  
 SF 100970  
 SF 100980  
 SF 100990  
 SF 101000  
 SF 101010  
 SF 101020  
 SF 101030  
 SF 101040  
 SF 101050  
 SF 101060  
 SF 101070  
 SF 101080  
 SF 101090  
 SF 101100  
 SF 101110  
 SF 101120  
 SF 101130  
 SF 101140  
 SF 101150  
 SF 101160  
 SF 101170  
 SF 101180  
 SF 101190  
 SF 101200  
 SF 101210  
 SF 101220  
 SF 101230  
 SF 101240  
 SF 101250  
 SF 101260  
 SF 101270  
 SF 101280  
 SF 101290  
 SF 101300  
 SF 101310  
 SF 101320  
 SF 101330  
 SF 101340  
 SF 101350  
 SF 101360  
 SF 101370  
 SF 101380  
 SF 101390  
 SF 101400  
 SF 101410  
 SF 101420  
 SF 101430  
 SF 101440  
 SF 101450  
 SF 101460  
 SF 101470  
 SF 101480  
 SF 101490  
 SF 101500  
 SF 101510  
 SF 101520  
 SF 101530  
 SF 101540  
 SF 101550  
 SF 101560  
 SF 101570  
 SF 101580

FILE: SET17

ORIGINAL PAGE IS  
OF POOR QUALITY

```
C      GO TO 105
C
C      HED1
C 400 M = NATCHK(CARD,COL)
      READ(RRUNIT,300) HED1
      REWIND RRUNIT
      GO TO 105
C
C      HED2
C 410 M = NATCHK(CARD,COL)
      READ(RRUNIT,300) HED2
      REWIND RRUNIT
      GO TO 105
C
C      CONVERT CARD
C 500 M=NATCHK(CARD,COL)
      IF(M.EQ.4LNK) GO TO 105
      IF(M.EQ.4FF) GO TO 510
      WRITE(NPRT,520)
      FORMAT(' ERROR ON CONVERT CARD')
      GO TO 105
C 510 J=FINDI2(CARD,COL,EQUCOM)
      M=NUMBER(CARD,COL,GTNOF,ZERO)
      COL=COL-1
      GO TO 500
C
C      *END*
C
C 420 CONTINUE
      IF (NOLAB.NE. 0) GO TO 440
      DO 430 I=1,6
      VLR(I) = 1
C 430 CONTINUE
      NOLAB = 6
      GO TO 440
C 450 WRITE(NPRT,153)
      NOLAB=0
      GO TO 420
C
C 440 CONTINUE
      IF(NOLAB.LT.6) GO TO 450
C
C
C      WRITE(NPRT,1000)
      WRITE(NPRT,1100) GTROU,GTROF
      WRITE(NPRT,1200) GTWRU,GTWRF
      WRITE(NPRT,1300) GTNOF
      WRITE(NPRT,1010) (VLR(I),I=1,6)
      IF (PRTKEY.EQ. 1) WRITE(NPRT,1030)
C 1000 FORMAT(// ' USER HAS REQUESTED THE FOLLOWING OPTIONS :'/)
C 1010 FORMAT(' THE LABEL VECTOR IS = ',6I3)
C 1030 FORMAT(' PRINT THE 204 DOT LABELS')
C 1100 FORMAT(' READ UNIT = ',I3,' READ FILE = ',I3)
C 1200 FORMAT(' WRITE UNIT = ',I3,' WRITE FILE = ',I3)
C 1300 FORMAT(' THE NUMBER OF FILES TO BE CONVERTED = ',I3)
C
C      RETURN
C
C      END
```

SET01590  
SET01600  
SET01610  
SET01620  
SET01630  
SET01640  
SET01650  
SET01660  
SET01670  
SET01680  
SET01690  
SET01700  
SET01710  
SET01720  
SET01730  
SET01740  
SET01750  
SET01760  
SET01770  
SET01780  
SET01790  
SET01800  
SET01810  
SET01820  
SET01830  
SET01840  
SET01850  
SET01860  
SET01870  
SET01880  
SET01890  
SET01900  
SET01910  
SET01920  
SET01930  
SET01940  
SET01950  
SET01960  
SET01970  
SET01980  
SET01990  
SET02000  
SET02010  
SET02020  
SET02030  
SET02040  
SET02050  
SET02060  
SET02070  
SET02080  
SET02090  
SET02100  
SET02110  
SET02120  
SET02130  
SET02140  
SET02150  
SET02160  
SET02170  
SET02180  
SET02190  
SET02200  
SET02210  
SET02220

FILE: TCN

```
C      WRITTEN BY C W AMLEPS
      SUBROUTINE TCN(ARRAY, TOP)
      IMPLICIT INTEGER (A-Z)
      DIMENSION ARRAY(1)
      LOGICAL *1 VRL(2400), IDL(800)
      DIMENSION IDATA1(3060)
      DIMENSION IPUF(765)
      DIMENSION FETVEC(30)
      DATA FETVEC/30*0/
      DIMENSION IDATA(3060)
      DIMENSION BLOCK(6)
      COMMON /TAPERD/ IUNIT, IFRST, FSCAN, SAMEND, SAMINC, READY, NSCAN,
      * LINC, ID(200), DSL, LBUF(30), JREC(30), IBYTE(30), NBUFS, FILENO, LINEND
      * LININC, NSAMP, NOCHAN, IFORMT
      COMMON /PRTAP/ ICOUNT, FORMT, UNIT, VARBL(600), IREM
      EQUIVALENCE (VRL, VARBL)
      EQUIVALENCE (ID, IDL)
C      LOGICAL *1 DY(4), MN(4), YR(4), ST(4)
C      EQUIVALENCE (DAY, DY(1)), (MON, MN(1)), (YEAR, YR(1)), (SITE, ST(1))
C      EQUIVALENCE (DAY, DY), (MON, MN), (YEAR, YR), (SITE, ST)
      COMMON /GTRK/NKDR, NPRT, PRTKEY, VLB(6), GTRDU, GTRDF,
      * GTWRU, GTWRF, GTNOF
      EQUIVALENCE (IDATA(1), ARRAY(1))
C      EQUIVALENCE (IDATA(1), ARRAY(3061))
      UNIT=GTWRU
      IUNIT=GTRDU
      FORMT=TFURMT
      NEOF=0
      FTLF=0
      NOCHAN=1
      NOFFAT=NOCHAN
      FETVEC(1)=1
      ZERO=0
      OUTPX=196
      INPX=392
      OSG=72
      ONE=1
      GTNRPO=3
      BLOCK(1)=1
      BLOCK(2)=351
      BLOCK(3)=1
      BLOCK(4)=1
      BLOCK(5)=392
      BLOCK(6)=1
      10 CONTINUE
      PEWIND GTRDU
      PEWIND GTWRU
      RDF=GTRDF-1
      WRF=GTWRF-1
      CALL FCFMFL(GTWRU, WRF, ISTATW)
C      10 CONTINUE
      IPEC=0
      ICOUNT=0
      NPFC=0
      FTLF=FILE+1
      DAY=0
      MON=0
      YEAR=0
      SITE=0
      CALL TAPHDR(GTRDU, RDF)
      DAY=ID(25)
      MON=ID(26)
      YEAR=ID(27)
      SITE=ID(28)
      VRL(73)=IDL(100)
      VRL(74)=IDL(104)
      VRL(75)=IDL(108)
      VRL(79)=IDL(111)
      VRL(80)=IDL(112)
      CALL WRTHFD(NOFFAT, FETVEC, OUTPX, FORMT, GTWRU)
      GTREC=540
      WRITE(NPRT, 601) GTRDF, GTWRF
601  FORMAT(//, ' FILE ', I4, 'SX, BEING CONVERTED TO FILE ', I4)
      WRITE(NPRT, 306) SITE, DAY, MON, YEAR
306  FORMAT(' SITE = ', I5, 'SX, 'DAY=' , I5, 'SX, 'MONTH=' , I5, 'SX,
      * 'YEAR=' , I5)
      IF(PRTKEY.EQ.1) WRITE(NPRT, 600)
      CALL FLDINT(BLOCK, FETVEC, NOFFAT)
      TCN0010
      TCN0020
      TCN0030
      TCN0040
      TCN0050
      TCN0060
      TCN0070
      TCN0080
      TCN0090
      TCN0100
      TCN0110
      TCN0120
      TCN0130
      TCN0140
      TCN0150
      TCN0160
      TCN0170
      TCN0180
      TCN0190
      TCN0200
      TCN0210
      TCN0220
      TCN0230
      TCN0240
      TCN0250
      TCN0260
      TCN0270
      TCN0280
      TCN0290
      TCN0300
      TCN0310
      TCN0320
      TCN0330
      TCN0340
      TCN0350
      TCN0360
      TCN0370
      TCN0380
      TCN0390
      TCN0400
      TCN0410
      TCN0420
      TCN0430
      TCN0440
      TCN0450
      TCN0460
      TCN0470
      TCN0480
      TCN0490
      TCN0500
      TCN0510
      TCN0520
      TCN0530
      TCN0540
      TCN0550
      TCN0560
      TCN0570
      TCN0580
      TCN0590
      TCN0600
      TCN0610
      TCN0620
      TCN0630
      TCN0640
      TCN0650
      TCN0660
      TCN0670
      TCN0680
      TCN0690
      TCN0700
      TCN0710
      TCN0720
      TCN0730
      TCN0740
      TCN0750
      TCN0760
      TCN0770
      TCN0780
      TCN0790
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: TCN

```
600 FORMAT(' THE 209 DOT LABELS')
70 CONTINUE
DO A01 J=1,3
  IREC=IREC+1
  IF (IREC.GT. BLOCK(2)) GO TO 50
  CALL LINERD(IDATA,ENDTAP)
C IF (ENDTAP.EQ.-1) GO TO 50
  DO A00 I =1,GTREC
    JJ=I+GTREC*(J-1)+72
    IDATA(JJ)=IDATA(I)
  A00 CONTINUE
  A01 CONTINUE
  NEOF=0
  NPEC=NREC+1
  OSGG=OSG
  CALL GTUNPK(IDATA,ONE,OUTPX,OSGG,GTNRPD,GTREC)
  G3=GTREC*3
C IF (NPEC.LE.2) WRITE(NPRT,850) (IDATA(K),K=1,63)
  850 FORMAT(1H ,15I4)
  CALL LINLAR(IDATA,GTREC)
  N10=MOD(NREC,10)
C IF (NPEC.EQ.1) WRITE(NPRT,850) (IDATA(K),K=1,GTREC)
  CALL WRTLN(IDATA,ZERO)
  IF (PRTKEY.EQ.1.AND.N10.EQ.0) WRITE(NPRT,839) (IDATA(I),I=10,190
  * .10)
  839 FORMAT(1H ,5X,19I5)
  GO TO 20
  50 CONTINUE
  ENDFILE GTWRU
  WRITE(NPRT,700) NREC
  700 FORMAT(' NUMBER OF SCAN LINES WRITTEN=',I5)
  WRITE(NPRT,750) GTRDF,GTWRF
  750 FORMAT(' FILE ',I5,5X,' WAS CONVERTED ONTO FILE ',I5)
  GTRDF=GTWRF+1
  RDF=RDF+1
  GTWRF=GTWRF+1
  IF (FILE.LT.GTNOF) GO TO 10
  ENDFILE GTWRU
  REWIND GTWRU
  REWIND GTWRU
  WRITE(NPRT,250)
  250 FORMAT(' PROGRAM GTTCN RUN COMPLETED')
  RETURN
  END
```

TCN00800  
TCN00810  
TCN00820  
TCN00830  
TCN00840  
TCN00850  
TCN00860  
TCN00870  
TCN00880  
TCN00890  
TCN00900  
TCN00910  
TCN00920  
TCN00930  
TCN00940  
TCN00950  
TCN00960  
TCN00970  
TCN00980  
TCN00990  
TCN01000  
TCN01010  
TCN01020  
TCN01030  
TCN01040  
TCN01050  
TCN01060  
TCN01070  
TCN01080  
TCN01090  
TCN01100  
TCN01110  
TCN01120  
TCN01130  
TCN01140  
TCN01150  
TCN01160  
TCN01170  
TCN01180  
TCN01190  
TCN01200  
TCN01210  
TCN01220  
TCN01230

ORIGINAL PAGE IS  
OF POOR QUALITY



## 23. TESTSP PROCESSOR

### FILE TESTSP

```

SUBROUTINE TESTSP (ARRAY, TOP)
C*
C*.....
C* THIS PROGRAM PERFORMS A MODIFIED VERSION OF THE CLUSTERING
C* ALGORITHM (ISODATA) ORIGINALLY DEVELOPED BY BALL AND HALL OF
C* STANFORD RESEARCH INSTITUTE. THE ALGORITHM HAS BEEN MODIFIED
C* ON THE RECOMMENDATIONS OF ED KAN (LEC).
C*
C* THE PROGRAM EXPECTS MULTISPECTRAL SCANNER DATA
C* IN EITHER THE LAPSYS 22 OR THE UNIVERSAL
C* FORMAT. THE DATA TAPE SHOULD BE ASSIGNED TO FORTRAN UNIT 3.
C*.....
C*
C* IMPLICIT INTEGER (A-X)
C* INCLUDE COMBK5.LIST
C* INCLUDE COMNT5.LIST
C* INCLUDE COMBK6.LIST
C* INCLUDE CMRK16.LIST
C* COMMON/PASS/STOP, LNCAT, NMIN, KRN, STDMAX, DLMIN, SEP,
C*   MAP, SPTRIG, IRD, KPTS, NOPTS, PUNCH,
C*   ICHN, CHNTHS, ICHAIN(62), NWDS, IREGIN, BEGIN1,
C*   BEGIN2, BEGIN3, CLSNAM, NOFLD, IPT, TOTWRD, TOTPTS,
C*   NCLASS, NOCLS, TOTSUR, TOTFLD, TOTVRT, NOCL, NVRT
C*   ,NXTCLS, NOFEAT, MAXCLS, FFTVEC(30), SYMMTX(62)
C*   , VARSIZ, STATKY, ISOKEY, MAPFMT, MAPKEY, SEQUEN(20), PERCEN, SIMERP
C*   , IORDER, INJUNIT, INFILE, INITM, PMIN, SUBVEC(62), NOSUB2, CHNVC(30)
C*   , NOCHAN, ERCOMP, NOSEQ, MEANDU, MEANDU,
C*   , SYMDO, SYMDU, ITRIGO, ITRIGU, DOFLAG,
C*   , DUFLAG, DODU, STDOTS(60), NSDOTS, SUNCOR(30), LLNCAT,
C*   , DVERT(250, 2), DRECT(160, 2), DVPNT(11, 2), IDCNT(2), NDOU(2)
C*   , MXFET1, MAXPOP
C* REAL SUNCOR
C*
C* COMMON BLOCK 'PASS' IS USED ONLY BY THE ISOCLS PROCESSOR.
C*
C* ISOCLS USES THE RANDOM ACCESS DRUM FILE AS FOUR DISTINCT FILES.
C* SEE DEFINITIONS OF IREGIN, BEGIN1, BEGIN2, BEGIN3 BELOW
C*
C* DEFINITIONS
C*
C*   ISTOP - MAX. NO. OF ITERATIONS FOR THE CLUSTERING PROCEDURE
C*           SET IN SETUP7 ROUTINE. (USER INPUT)
C*   LNCAT - CURRENT NO. OF CLUSTERS. SET INITIALLY IN RDFILE OR ISTOP
C*           ISOCLS. THEN ONLY IN ISODAT.
C*   NMIN  - MIN. NO. OF POINTS TO ALLOW PER CLUSTER
C*           SET IN SETUP7 ROUTINE. (USER INPUT)
C*   KRN   - PRINT CLUSTER SUMMARY EVERY 'KRN' ITERATION(S)
C*           SET IN SETUP7 ROUTINE. (USER INPUT)
C*   STDMAX - STANDARD DEVIATION FOR SPLITTING CLUSTERS
C*           SET IN SETUP7 ROUTINE. (USER INPUT)
C*   DLMIN - MIN. DISTANCE BETWEEN CLUSTERS FOR COMBINING.
C*   SEP   - DISTANCE TO SEPARATE CLUSTERS. SET EITHER IN SETUP7,
C*           BY USER INPUT, OR IN ID
C*           BY USER INPUT, OR IN ISODAT.
C*   MAP   - PRINT A CLUSTER MAP EVERY 'MAP' ITERATION(S) - SETUP7
C*   SPTRIG - TRIGGER TELLING WHETHER OR NOT 'SEP' WAS INPUT. - SETUP7
C*   IRD   - NO. OF RECORDS TO READ FROM DATA FILE. COMPUTED IN
C*           ISOCLS
C*   NOPTS - NO. OF POINTS IN EACH RECORD. COMPUTER IN ISOCLS
C*   CONTINUE
C*   KPTS  - NO. OF POINTS IN LAST RECORD. COMPUTER IN ISOCLS
C*   PUNCH - TRIGGER TELLING WHETHER OR NOT TO PUNCH THE MODULE
C*           STAT DFCK. - SETUP7
C*   ICHN  - TRIGGER TELLING WHETHER OR NOT CHAINING IS TO BE DONE
C*   CHNTHS - MIN. DISTANCE BETWEEN CLUSTERS FOR CHAINING - SETUP7
C*   ICHAIN - ARRAY CONTAINING CHAINED CLUSTER NUMBERS. SET IN
C*           'CHAIN' ROUTINE.
C*   NWDS  - TOTAL NO. OF WORDS AVAILABLE FOR DRUM STORAGE OF
C*           IMAGE DATA TO BE CLUSTERED - SET IN ISOCLS
C*   IREGIN - BEGINNING DRUM FILE ADDRESS FOR INPUT INITIAL CLUSTER
C*           CENTERS - SET IN ISOCLS
C*   REGINS - BEGINNING DRUM FILE ADDRESS FOR TEMPORARY STORAGE OF
C*           CLASS STATISTICS - SET IN ISOCLS ROUTINE
C*   BEGIN1 - BEGINNING DRUM FILE ADDRESS FOR IMAGE DATA

```

FILE TESTSP

```

C*      BEGIN2 - BEGINNING DRUM FILE ADDRESS FOR 'IPLACE' (CLUSTER TO TES00770
C*      WHICH CORRESPONDING POINT BELONGS.) TES00780
C*      CLSNAM - NAME OF CLASS CURRENTLY BEING PROCESSED. - RDDATA TES00790
C*      NOFLD  - NO. OF FIELDS INPUT FOR THIS CLASS - RDDATA TES00800
C*      IPT    - NO. OF WORDS OF STORAGE USED IN 'ARRAY' FOR FIELD AND TES00810
C*      CLASS INFORMATION FOR THIS CLASS. - RDDATA TES00820
C*      TOTWRD - TOTAL WORDS WRITTEN ON DRUM FILE BEGINNING AT ADDRESS TES00830
C*      REGIN1 - RDDATA TES00840
C*      TOTPTS - TOTAL POINTS TO BE CLUSTERED FOR CURRENT CLASS - RDDATA TES00850
C*      NCLASS - NO. OF CLASSES TO BE CLUSTERED FOR CURRENT CALL TO TES00860
C*      ISOCLS - USER INPUT - SETUP7. TES00870
C*      NOCLS  - CURRENT CLASS NO. - ISOCLS TES00880
C*      TOTSUB - TOTAL CLUSTERS FOR THIS CALL TO ISOCLS TES00890
C*      TOTFLD - TOTAL FIELDS FOR ALL CLASSES - ISOCLS TES00900
C*      TOTVRT - TOTAL VERTICES FOR ALL FIELDS - ISOCLS TES00910
C*      NOCL   - NO. OF CLASSES SINCE LAST CALL TO SETUP - RDDATA TES00920
IEY033I COMMENTS DELETED *****
COMMON/GLOBAL/HEAD(63),MAPTAP,DATEPE,SAVTAP,BMFILE,BMKEY, TES01200
* HISFIL,HISKEY,TRFORM,ERIP,T,ERPKEY,MAPUNT,NOFILE, TES01210
* DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL TES01220
* ,NHSTUN,NHSTFI,SCTRUN,MAPFIL TES01230
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, TES01240
* CRDUNT,PRUNT,RANDIO TES01250
COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP TES01260
CSEND TES01270
DIMENSION KVAR(11500) TES01280
KVARDM = 11500 TES01290
DIMENSION ARRAY(1) TES01300
DIMENSION COVAR(465) TES01310
DIMENSION NN(60) TES01320
DATA SYMDA /* /*,SYMDB /* /* /* TES01330
MAXPOP=62 TES01340
MXFET1=30 TES01350
IBEGIN=DRUMAD TES01360
C* TES01370
C* RESERVE ENOUGH DRUM STORAGE FOR MAXIMUM INITIAL MEANS TES01380
C* TES01390
C* BEGIN3=IBEGIN + MAXPOP*MXFET1 + MXFET1 + 2 TES01400
C* TES01410
C* CALL SETUP TO READ CARD INPUT AND INITIALIZE DEFAULT VALUES TES01420
C* TES01430
ITIME=1 TES01440
NOCLS = 0 TES01450
TOTFLD = 0 TES01460
TOTVRT = 0 TES01470
TOTSUB = 0 TES01480
CORBAS=1 TES01490
ITRIGU = 0 TES01500
ITPIGO=0 TES01510
SYMDO = SYMDA TES01520
SYMDU = SYMDR TES01530
MEANDU = 0 TES01540
MEANDU = 255 TES01550
1 CALL SETUP7(ARRAY(CORBAS),TOP,ITIME) TES01560
IDUM = MAXCLS TES01570
IF (ITIME.GT.1)GO TO 2 TES01580
VARSIZ=NOFEAT*(NOFEAT+1)/2 TES01590
REGIN1 = REGIN3 + NCLASS*MAXPOP*(VARSIZ + NOFEAT + 1) TES01600
NWDS=DRMWDS-(BEGIN1-DRUMAD) TES01610
2 ITIME=ITIME+1 TES01620
NOCL=0 TES01630
C* TES01640
C* CALL RDDATA TO COORDINATE READING OF DATA TES01650
C* TES01660
5 MAXDIM = TOP-CORBAS TES01670
FD1=CORBAS TES01680
CALL RDDPAT(FD1,MAXDIM,KVAR,IVARDM,LAST) TES01690
MAXCLS = IDUM + DODU TES01700
WRITE(4,210) NDOU(1),NDOU(2) TES01710
210 FORMAT(1X,/' ' D0/DU CLUSTER POP FOR THIS CLASS ',217) TES01720
REGIN2 = BEGIN1 + (TOTWRD/4) + 2 TES01730
N1 = FD1 + IPT TES01740
MEANS1=N1 + MAXCLS TES01750
STDEV1=MEANS1 + MAXCLS*NOFEAT TES01760
ITOP = STDEV1 + MAXCLS*NOFEAT TES01770
MAXDIM=TOP-ITOP TES01780

```

23-2  
515

OF POOR QUALITY

FILE TESTSP

```

NOPTS = MAXDIM/(NOFEAT+1)
NOPTS = (NOPTS/4)*4
C PART OF PACKING CHG SEPT 1978
IDAT1 = TTOP
IF (NSDOTS.EQ.0) GO TO 4
DOTDMF = NOCHAN
TYPST = 1
CALL RDDOTS (ARRAY (MEANS1), STDOTS, NSDOTS,
* TYPST, DOTDMF, DOTDMC, DOTDUM, COVAR,
* NOCHAN, CHNVC, DTD, COVAR,
* DOTDM, DOTDM, DOTDM, DOTDM, DOTDM, DOTDM, KVAR)
LNCAT = NSDOTS
DO 500 I = 1, NSDOTS
DO 500 K = 1, NOFEAT
III = (I-1)*NOFEAT + K
II = III + MEANS1 - 1
500 ARRAY (II) = KVAR (III)
IF (NOCHAN.EQ.NOFEAT) GO TO 8
WRITE (6,110)
110 FORMAT (1H, 'NO CHANNELS FOR STARTING NOT EQUAL THAT FOR CLUSTER')
GO TO 9
4 CONTINUE
IF (ISOKEY.EQ.1) GO TO 7
C* SURVEC= SURCLASSES FROM STATISTICS FILE FOR INTIAL MEANS.
C* NOSUB2= NUMBER OF INITIAL MEANS.
C* CHNVEC= NUMBER OF CHANNELS FROM STATISTICS FILE. NOCHAN MUST EQUAL
IF (INITM.EQ.1) GO TO 6
LNCAT=1
GO TO 8
6 LNCAT=NOSUB2
CALL GETST (INUNIT, INFILE, ARRAY (MEANS1), DUM, NOSUB2, SUBVEC, NOCHAN
* , CHNVC, ARRAY (TTOP), COVAR, 0)
LNCAT = NOSUB2
GO TO 8
7 CONTINUE
IF (ISOKEY.EQ.1) CALL RDFILE (ARRAY (MEANS1), ARRAY (TTOP))
8 CONTINUE
IF (NOPTS.GT.0) GOTO 10
WRITE (6,100) MAXDIM
100 FORMAT (' DIMENSION LIMITS EXCEEDED IN ISOCLS BY', I6,
* ' REDUCE CHANNELS OR MAX. CLUSTERS')
9 CALL CMERR
10 CONTINUE
IRD=TOTPTS/NOPTS
IF (MOD (TOTPTS, NOPTS).EQ.0) GO TO 20
KPTS=MOD (TOTPTS, NOPTS)
IRD=IRD+1
IF (IRD.EQ.1) NOPTS=KPTS
GO TO 25
20 KPTS=NOPTS
25 CONTINUE
C*
C* CALL ISODAT TO PERFORM CLUSTERING
C*
A1=1
A2=A1+ MAXCLS*NOFEAT
CLD1=A2 + MAXCLS*NOFEAT
KPLCE = NOPTS*NOFEAT + IDAT1
CALL ISOPAT (IDAT1, ARRAY (KPLCE), ARRAY (MEANS1), ARRAY (N1),
* ARRAY (STDEV1), KVAR (CLD1), ARRAY (FD1), KVAR (A1),
* KVAR (A2))
C*
C* CHAIN CLUSTERS WHOSE DISTANCES ARE LESS THAN DLMIN
C*
LNCAT=LNCAT+DODU
IF (ICHN.GT.0) CALL CHAIN (KVAR (CLD1))
C*
C* PRINT FINAL RESULTS
C*
CALL PRINT (-1, ARRAY (KPLCE), ARRAY (MEANS1), ARRAY (STDEV1),
* KVAR (CLD1), ARRAY (FD1), ARRAY (N1))
C*
C* CREATE MAP OUTPUT TAPE FOR PMIS DAS IF DESIRED
C*
IF (MAPFMT.GT.0) CALL DSTAPE (ARRAY (KPLCE), KVAR (1), ARRAY (MEANS1),
* ARRAY (FD1))

```

TES01790  
 TES01800  
 TES01810  
 TES01820  
 TES01830  
 TES01840  
 TES01850  
 TES01860  
 TES01870  
 TES01880  
 TES01890  
 TES01900  
 TES01910  
 TES01920  
 TES01930  
 TES01940  
 TES01950  
 TES01960  
 TES01970  
 TES01980  
 TES01990  
 TES02000  
 TES02010  
 TES02020  
 TES02030  
 TES02040  
 TES02050  
 TES02060  
 TES02070  
 TES02080  
 TES02090  
 TES02100  
 TES02110  
 TES02120  
 TES02130  
 TES02140  
 TES02150  
 TES02160  
 TES02170  
 TES02180  
 TES02190  
 TES02200  
 TES02210  
 TES02220  
 TES02230  
 TES02240  
 TES02250  
 TES02260  
 TES02270  
 TES02280  
 TES02290  
 TES02300  
 TES02310  
 TES02320  
 TES02330  
 TES02340  
 TES02350  
 TES02360  
 TES02370  
 TES02380  
 TES02390  
 TES02400  
 TES02410  
 TES02420  
 TES02430  
 TES02440  
 TES02450  
 TES02460  
 TES02470  
 TES02480  
 TES02490  
 TES02500  
 TES02510  
 TES02520  
 TES02530  
 TES02540

FILE TESTSP

```

C*      LNCAT=LNCAT-DODU
C*      CALCULATE COVARIANCE MATRIX FOR EACH CLUSTER
C*      IF (VARSIZ*LNCAT.GT.KVARDM)GO TO 30
C*      CALL COVPAT(KVAR,IDAT1,ARRAY(KPLCE),ARRAY(MEANS1),
C*              ARRAY(N1),IBAD)
C*      CHECK FOR A CLUSTER DELETED FOR SINGULAR MATRIX
C*      IF (IBAD.NE.0)STOP=0
C*      IF (IBAD.NE.0)GO TO 25
C*
26      DO 26 II=1,LNCAT
NN(TOTSUB+II) = ARRAY(N1+II-1)
TOTSUB = TOTSUB + LNCAT
NOCLS = NOCLS + 1
TOTFLD = TOTFLD + NOFLD
TOTVRT = TOTVRT + NVRT
ARRAY(FD1+1)=IPT + FD1
ARRAY(FD1+2)=LNCAT
ARRAY(FD1+3)=NOFLD
C*
C*      WRITE STATS FOR THESE CLUSTERS ON SCRATCH FILE 18
C*      IF (NOCLS.EQ.1) ADRES=BEGIN3
C*      IN=NOFEAT*LNCAT
C*      CALL RWRITE (ADRES,ARRAY(MEANS1),IN,JSTAT)
C*      ADRES=ADRES+IN
C*      IN=VARSIZ*LNCAT
C*      CALL RWRITE (ADRES,KVAR,IN,LSTAT)
C*      ADRES=ADRES+IN
C*      WAIT FOR I/O COMPLETION
60      IF (LSTAT.EQ.1) GO TO 60
C*
C*      GO READ IN ANOTHER CLASS
C*
CORBAS=CORBAS+IPT
IF (LAST.NE.1)GO TO 5
IF (NOCLS.LT.NCLASS)GO TO 1
C*
C*      NOW READ SCRATCH FILE AND STORE ON SAVTAP FILE AND PUNCH ON
C*      CARDS IF REQUESTED.
C*
FLD1 = 1
VERTX1 = FLD1 + TOTFLD*4
CLSNM1 = VERTX1 + TOTVRT*2
NOSUR1 = CLSNM1 + NOCLS
SURNM1 = NOSUR1 + NOCLS
C
C*      RETRIEVE INFORMATION FROM 'ARRAY'
C*
CALL GETINF (ARRAY(1),KVAR(FLD1),KVAR (VERTX1),KVAR (CLSNM1),
*          KVAR (NOSUB1),KVAR (SUBNM1),NOCLS,TOTSUB)
C
SWTCH = 1
C
C*      OUTPUT STATS
C*
CALL LABMAN (SAVTAP,STAFIL,NOCLS,TOTSUB,NOFEAT,TOTFLD,TOTVRT,
*          FETVEC,KVAR (FLD1),KVAR (VERTX1),KVAR (CLSNM1),KVAR (NOSUB1),
*          KVAR (SUBNM1),NN,REGIN3,VARSIZ,PUNCH,DUMMY,STATKY,SWTCH)
RETURN
30      KV=KVARDM
WRITE (6,200)KV
CALL CMERR
200     FORMAT (' DIMENSION LIMIT OF ',I6,' FOR COVARIANCES EXCEEDED')
RETURN
END

```

TE503250  
TE503255  
TE503260  
TE503265  
TE503270  
TE503275  
TE503280  
TE503285  
TE503290  
TE503295  
TE503300  
TE503305  
TE503310  
TE503315  
TE503320  
TE503325  
TE503330  
TE503335  
TE503340  
TE503345  
TE503350  
TE503355  
TE503360  
TE503365  
TE503370  
TE503375  
TE503380  
TE503385  
TE503390  
TE503395  
TE503400  
TE503405  
TE503410  
TE503415  
TE503420  
TE503425  
TE503430  
TE503435  
TE503440  
TE503445  
TE503450  
TE503455  
TE503460  
TE503465  
TE503470  
TE503475  
TE503480  
TE503485  
TE503490  
TE503495  
TE503500  
TE503505  
TE503510  
TE503515  
TE503520  
TE503525  
TE503530  
TE503535  
TE503540  
TE503545  
TE503550  
TE503555  
TE503560  
TE503565  
TE503570  
TE503575  
TE503580  
TE503585  
TE503590  
TE503595  
TE503600  
TE503605  
TE503610  
TE503615  
TE503620  
TE503625  
TE503630  
TE503635  
TE503640  
TE503645  
TE503650  
TE503655  
TE503660  
TE503665  
TE503670  
TE503675  
TE503680  
TE503685  
TE503690  
TE503695  
TE503700  
TE503705  
TE503710  
TE503715  
TE503720  
TE503725  
TE503730  
TE503735  
TE503740  
TE503745  
TE503750  
TE503755  
TE503760  
TE503765  
TE503770  
TE503775  
TE503780  
TE503785  
TE503790  
TE503795  
TE503800  
TE503805  
TE503810  
TE503815  
TE503820  
TE503825  
TE503830  
TE503835  
TE503840  
TE503845  
TE503850  
TE503855  
TE503860  
TE503865  
TE503870  
TE503875  
TE503880  
TE503885  
TE503890  
TE503895  
TE503900  
TE503905  
TE503910  
TE503915  
TE503920  
TE503925  
TE503930  
TE503935  
TE503940  
TE503945  
TE503950  
TE503955  
TE503960  
TE503965  
TE503970  
TE503975  
TE503980  
TE503985  
TE503990  
TE503995  
TE504000  
TE504005  
TE504010  
TE504015  
TE504020  
TE504025  
TE504030  
TE504035  
TE504040  
TE504045  
TE504050  
TE504055  
TE504060  
TE504065  
TE504070  
TE504075  
TE504080  
TE504085  
TE504090  
TE504095  
TE504100  
TE504105  
TE504110  
TE504115  
TE504120  
TE504125  
TE504130  
TE504135  
TE504140  
TE504145  
TE504150  
TE504155  
TE504160  
TE504165  
TE504170  
TE504175  
TE504180  
TE504185  
TE504190  
TE504195  
TE504200  
TE504205  
TE504210  
TE504215  
TE504220  
TE504225  
TE504230



FILE COVPAT

```

      IPACK1 = IPACK1*IPACK
      DUM = IPACK1
      KK=KK+1
      COVAR(KK,ICLS)=COVAR(KK,ICLS)*DUM
40  CONTINUE
45  CONTINUE
      IRC=IRC-1
      IF (IRC.GT.0) GO TO 20
      DO 50 I=1,LNCAT
      IF (N(I).EQ.0) GO TO 50
      KK=0
      DO 50 J=1,NOFEAT
      DO 50 K=1,J
      KK=KK+1
      COVAR(KK,I)=COVAR(KK,I)/N(I) - MEANS(K,I)*MEANS(J,I)
50  CONTINUE
      IACEPT=PMIN*NOFEAT
      IF (IACEPT.LT.NOFEAT) GO TO 58
C
C
      CHECK FOR SINGULAR COVARIANCE MATRIX
      DO 51 I=1,LNCAT
      CALL CHLDET(COVAR(I,I),NOFEAT,DUM,DET)
      IF (DET.LT.TOL) GO TO 52
51  CONTINUE
      GO TO 58
C
C
      DELETE SINGULAR COVARIANCE CLUSTER
52  WRITE(6,160)I
      IF (LNCAT.EQ.1) CALL CMERR
      IRAD=1
      LNCAT=LNCAT-1
      LLNCAT=LLNCAT-1
      DO 53 II=1,LNCAT
      DO 53 III=1,NOFEAT
      MEANS(III,II)=MEANS(III,II+1)
53  CONTINUE
58  RETURN
160  FORMAT(2X,'CLUSTER',I5,' DELETED FOR SINGULARITY')
C
      IF (STATKY.NE.1) RETURN
55  WRITE(6,HEAD)
      WRITE(6,150)CLSNAM
      DO 60 I=1,LNCAT
      WRITE(6,90)I
      DO 70 LOC=1,NOFEAT,12
      ISTOP=LOC+11
      IF (ISTOP.GT.NOFEAT) ISTOP=NOFEAT
      WRITE(6,140) (CH,FETVEC(J),J=LOC,ISTOP)
      II=1
      KINC=1
      DO 60 J=LOC,NOFEAT
      K=J*(J+1)/2-II+1
      JK=K+KINC-1
      WRITE(6,100) (COVAR(M,I),M=K,JK)
      II=II+1
60  IF (KINC.LT.ISTOP.AND.KINC.LT.12) KINC=KINC+1
      WRITE(6,110)
70  CONTINUE
80  CONTINUE
      RETURN
90  FORMAT(//,' COVARIANCE MATRIX FOR CLUSTER',I4//)
100  FORMAT(/6X,12F9.2)
110  FORMAT(//)
120  FORMAT(1H)
140  FORMAT(9X,12(A3,I2,' '),3X))
150  FORMAT(/,' COVARIANCES FOR CLASS',2X,A4//)
      END

```

COV00770  
 COV00780  
 COV00790  
 COV00800  
 COV00810  
 COV00820  
 COV00830  
 COV00840  
 COV00850  
 COV00860  
 COV00870  
 COV00880  
 COV00890  
 COV00900  
 COV00910  
 COV00920  
 COV00930  
 COV00940  
 COV00950  
 COV00960  
 COV00970  
 COV00980  
 COV00990  
 COV01000  
 COV01010  
 COV01020  
 COV01030  
 COV01040  
 COV01050  
 COV01060  
 COV01070  
 COV01080  
 COV01090  
 COV01100  
 COV01110  
 COV01120  
 COV01130  
 COV01140  
 COV01150  
 COV01160  
 COV01170  
 COV01180  
 COV01190  
 COV01200  
 COV01210  
 COV01220  
 COV01230  
 COV01240  
 COV01250  
 COV01260  
 COV01270  
 COV01280  
 COV01290  
 COV01300  
 COV01310  
 COV01320  
 COV01330  
 COV01340  
 COV01350  
 COV01360  
 COV01370  
 COV01380  
 COV01390  
 COV01400  
 COV01410  
 COV01420  
 COV01430  
 COV01440



FILE: ISOPAT

```

IF (ERCOMP.NE.1) GO TO 135
ESUM=0.0
DO 132 J=1,NOFEAT
DO 133 K=1,LNCAT
132 ESUM=ESUM+N(K)*(STDEV(J,K))**2/TOTPTS
CONTINUE
ESOT=SORT(ESUM/NOFEAT)
WRITE(6,133) ESOT,PERCENT,STOMAX
133 FORMAT(1X,///' ERCOMP= ',F7.3,' PERCENT = ',F5.3,' STOMAX = ',
.F7.3/)
CCC
CALCULATE DISTANCES BETWEEN CLUSTER CENTERS
135 CALL CLDIST(CLD,STDEV,MEANS)
CCC
IF STOP EQUALS ZERO DELETE SMALL CLUSTERS
LNCAT=LLNCAT
IF (MOD(KKT,MAP)) 150,140,150
140 CALL PRINT(KKT,IPLACE,MEANS,STDEV,CLD,FLDINF,N)
GO TO 161
150 IF (MOD(KKT,KRN)) 161,160,161
160 CONTINUE
CALL PRINT(KKT,IPLACE,MEANS,STDEV,CLD,FLDINF,N)
161 CONTINUE
LNCAT=LLNCAT-DODU
IF (STOP.EQ.0) GO TO 162
CCC
FOR ITERATION N CHECK N(K) AGAINST PMIN + NOFEAT
IF (ISEQ.NE.NOSEQ) GO TO 169
ISTOP = 0
162 DO 163 K = 1,LNCAT
IF (N(K) - (PMIN + NOFEAT)) 167,163,163
163 CONTINUE
IF (.NOT.DFL) RETURN
DO 164 KK=1,LLNCAT
DO 164 KKK=1,NOFEAT
164 MEANS(KKK,KK) = MEAN(KKK,KK)
CALL PCDATA(MEANS,STDEV,N,CLD,IDAT1,IPLACE,AVP,AMN,MEANS)
DO 165 KV=1,LLNCAT
DO 165 KKK=1,NOFEAT
165 MEAN(KKK,KK) = MEANS(KKK,KK)
CALL CLDIST(CLD,STDEV,MEANS)
RETURN
167 WRITE(6,168)K,N(K),PMIN,NOFEAT
168 FORMAT(/' CLUSTER ',I3,' REMOVED FOR HAVING ONLY ',I6,' POINTS. '/
.' MIN. POINTS IS (' ,I4,' . ',I2,' )')
RETF=1
LK=K
GO TO 570
171 K=LK
DEL = .TRUE.
GO TO 162
169 CONTINUE
170 CONTINUE
CCC
ON ITERATIONS 1 THRU N-1 CHECK N(K) AGAINST NMIN
DO 180 K=1,INCAT
IF (N(K)-NMIN) 190,180,180
180 CONTINUE
IF (DEL) CALL CLDIST(CLD,STDEV,MEANS)
GO TO 220
190 IF (MOD(KKT,KRN)) 200,195,200
195 WRITE (6,210)K,N(K),NMIN
200 RETF=2
LK=K
GO TO 570
201 K=LK
DEL = .TRUE.
GO TO 170
210 FORMAT('0 CLUSTER ',I2,' REMOVED FOR HAVING ONLY ',I6,
' ELEMENTS. MIN. NO. ELEMENTS IS ',I6)
220 CONTINUE
C
C* SPLIT ITERATION

```

```

5000800
5000810
5000820
5000830
5000840
5000850
5000860
5000870
5000880
5000890
5000900
5000910
5000920
5000930
5000940
5000950
5000960
5000970
5000980
5000990
5001000
5001010
5001020
5001030
5001040
5001050
5001060
5001070
5001080
5001090
5001100
5001110
5001120
5001130
5001140
5001150
5001160
5001170
5001180
5001190
5001200
5001210
5001220
5001230
5001240
5001250
5001260
5001270
5001280
5001290
5001300
5001310
5001320
5001330
5001340
5001350
5001360
5001370
5001380
5001390
5001400
5001410
5001420
5001430
5001440
5001450
5001460
5001470
5001480
5001490
5001500
5001510
5001520
5001530
5001540
5001550
5001560
5001570
5001580

```





FILE: ISOPAT

```

      NOCLST=LNCAT-1
      L=1
406  L=L+2
      IF (L.GT.NOCLST) GO TO 480
      NOCLTR = LNCAT - 1
      KK=0
      DMIN=DL*MIN
      DO 430 I=1,NOCLTR
C
      IF (PTR(I).EQ.0) GO TO 430
      II=I+1
      DO 425 J=II,LNCAT
      IF (PTR(J).EQ.0) GO TO 425
      SDIJ = 0.0
      DO 420 JJ=1,KDIM
      SDIJ=SDIJ*((AMN(JJ,I)-AMN(JJ,J))**2/(STDEV(JJ,I)*STDEV(JJ,J)))
420  CONTINUE
      DIJ=SQRT(SDIJ)
C
      IF (DIJ.GT.DMIN) GO TO 425
      DMIN=DIJ
      KK=I
      KKK=J
425  CONTINUE
430  CONTINUE
C
      IF (KK.EQ.0) GO TO 480
      PTR(KK)=0
C
      COMBINE CLUSTERS KK AND KKK
C
      DEL=.TRUE.
      RND=.10 /FLOAT(N(KK)*N(KKK))
C
      DO 460 K=1,KDIM
460  AMN(K,KK)=(N(KK)*AMN(K,KK)+N(KKK)*AMN(K,KKK))*RND
C
      NETF=3
      LK=KKK
      GO TO 570
461  KKK=LK
      IF (KKK.EQ.(LNCAT+1)) GO TO 435
C
      MOVE POINTERS UP
C
      DO 175 K=KKK,LNCAT
175  PTR(K) = PTR(K+1)
C
435  IF (MOD(KKT,KRN))440,441,440
441  WRITE(A,490)KK,KKK,KK
440  IF (L.LT.NOCLST) GO TO 406
C
480  CONTINUE
490  FORMAT(' CLUSTERS ',I2,' AND ',I2,' HAVE BEEN COMBINED INTO CLUST
      2ER ',I2)
C
      REINITIALIZE
C
500  CONTINUE
      DO 510 J=1,MAXCLS
      SGMA(J)=0.0
      ISGMA(J)=0
      DO 510 K=1,KDIM
      AVP(K,J)=0.0
      STDEV(K,J)=0.0
      MEANS(K,J)=AMN(K,J)
      AMN(K,J)=0.0
510  CONTINUE
      KKT=KKT+1
      DEL=.FALSE.
      GO TO 10
C
530  IF (KKT.NE.2) GO TO 550
      WRITE (6,540)
540  FORMAT(' THE ORIGINAL CLUSTER WAS NOT SPLIT - EXAMINE THE INPUT VA
      LUE FOR STOMAXI/')
      KKT=1
      ISTOP=0
      GO TO 10

```

5003180  
5003190  
5003200  
5003210  
5003220  
5003230  
5003240  
5003250  
5003260  
5003270  
5003280  
5003290  
5003300  
5003310  
5003320  
5003330  
5003340  
5003350  
5003360  
5003370  
5003380  
5003390  
5003400  
5003410  
5003420  
5003430  
5003440  
5003450  
5003460  
5003470  
5003480  
5003490  
5003500  
5003510  
5003520  
5003530  
5003540  
5003550  
5003560  
5003570  
5003580  
5003590  
5003600  
5003610  
5003620  
5003630  
5003640  
5003650  
5003660  
5003670  
5003680  
5003690  
5003700  
5003710  
5003720  
5003730  
5003740  
5003750  
5003760  
5003770  
5003780  
5003790  
5003800  
5003810  
5003820  
5003830  
5003840  
5003850  
5003860  
5003870  
5003880  
5003890  
5003900  
5003910  
5003920  
5003930  
5003940  
5003950  
5003960  
5003970  
5003980  
5003990  
5004000  
5004010  
5004020  
5004030  
5004040  
5004050  
5004060  
5004070  
5004080  
5004090  
5004100  
5004110  
5004120  
5004130  
5004140  
5004150  
5004160

FILE: ISOPAT

```
550 WRITE (6,560)KKT
560 FORMAT(//, 'AFTER ',I4,' ITERATIONS ALL DATA HAS BEEN ASSIGNED TO 0
      *NF CLUSTER',I)
      KKT=I
      ISTOP=0
      GO TO 10
570 CONTINUE
C*
C* ROUTINE TO DELETE A CLUSTER
      INCAT=INCAT-1
      LLNCAT=LLNCAT-1
      IF (LK.EQ.(INCAT-1).AND.DODU.EQ.0) GO TO (171,201,461),RETF
      DO 561 J=LK,LLNCAT
      DO 552 L=1,K*IM
      AMN(L,J)=AMN(L,J-1)
      MEANS(L,J)=MEANS(L,J-1)
      MEAN(L,J)=MEAN(L,J-1)
552 STDEV(L,J)=STDEV(L,J-1)
      N(J)=N(J-1)
561 CONTINUE
      GO TO (171,201,461),RETF
      END
```

```
5003170
5003180
5003190
5003200
5003210
5003220
5003230
5003240
5003250
5003260
5003270
5003280
5003290
5003300
5003310
5003320
5003330
5003340
5003350
5003360
5003370
5003380
5003390
```



FILE: PSPPAT

```
42  KK=1
    SDIST=10.0E+20
    DO 46 J=1,LNCAT
    DIST=0.
    DO 44 K=1,NOFEAT
      LPACK(4) = LARRAY(IBASE + K)
      CSUN(K) = IPACK(1)
44  DIST=DIST+ARS(MEANS(K,J)-CSUN(K))*SUNCOR(K)
    IF (DIST - SDIST) 45,46,46
45  KK=J
    SDIST=DIST
46  CONTINUE
47  CONTINUE
    N(KK)=N(KK)+1
    IPLACE(1)=KK
    DO 48 K=1,NOFEAT
      AMN(K,KK) = AMN(K,KK) + CSUN(K)
      AVP(K,KK) = AVP(K,KK) + CSUN(K)**2
48  CONTINUE
49  CONTINUE
    GO TO 101
50  DO 100 I = 1,ICCT
      IBASE = (I - 1)*NOFEAT + IB
      KK=1
      IF (DODU.EQ.0) GO TO 52
      DO 51 K=1,NOFEAT
        LPACK(4) = LARRAY(IBASE + K)
        CDUM = IPACK(1)
        IF (CDUM.NE.MEAND0.AND.CDUM.NE.MEANDU) GO TO 52
51  CONTINUE
      IF (CDUM.EQ.MEAND0) IPLACE(1) = LNCAT + 1
      IF (CDUM.EQ.MEANDU) IPLACE(1) = LNCAT + DODU
      GO TO 100
52  CONTINUE
      KK = 1
      SDIST=10.0E+20
      DO 70 J=1,LNCAT
      DIST=0.0
      DO 55 K = 1,NOFEAT
        LPACK(4) = LARRAY(IBASE + K)
        CSUN(K) = IPACK(1)
55  DIST = DIST + ARS(MEANS(K,J) - CSUN(K))
      IF (DIST-SDIST)60,70,70
60  KK=J
      SDIST=DIST
70  CONTINUE
80  CONTINUE
      N(KK)=N(KK)+1
      IPLACE(1)=KK
      DO 90 K=1,NOFEAT
        AMN(K,KK) = AMN(K,KK) + CSUN(K)
        AVP(K,KK) = AVP(K,KK) + CSUN(K)**2
90  CONTINUE
100 CONTINUE
101 CONTINUE
      IF (IRD.EQ.0)GO TO 110
      CALL RWRITE (ADRES2,IPLACE,ICCT,ISTAT)
      ADRES2=ADRES2+ICCT
105  IF (ISTAT.EQ.1)GO TO 105
110  IRC=IRC-1
      IF (IRC.GT.0)GO TO 20
      KA = 1
115  CONTINUE
      DO 130 K=KA,LNCAT
      IF (N(K).EQ.0)GO TO 130
      RND=FLOAT(N(K))
      DO 130 J=1,NOFEAT
        AMN(J,K) = AMN(J,K) / RND
        MEANS(J,K) = AMN(J,K)
        STDEV(J,K) = SORT(AVP(J,K) / RND - AMN(J,K)*AMN(J,K))
        DUMA = STDEV(J,K)
        IF (DUMA.LT.DUM) STDEV(J,K) = DUM
130 CONTINUE
    RETURN
    END
```

PSP00800  
PSP00810  
PSP00820  
PSP00830  
PSP00840  
PSP00850  
PSP00860  
PSP00870  
PSP00880  
PSP00890  
PSP00900  
PSP00910  
PSP00920  
PSP00930  
PSP00940  
PSP00950  
PSP00960  
PSP00970  
PSP00980  
PSP00990  
PSP01000  
PSP01010  
PSP01020  
PSP01030  
PSP01040  
PSP01050  
PSP01060  
PSP01070  
PSP01080  
PSP01090  
PSP01100  
PSP01110  
PSP01120  
PSP01130  
PSP01140  
PSP01150  
PSP01160  
PSP01170  
PSP01180  
PSP01190  
PSP01200  
PSP01210  
PSP01220  
PSP01230  
PSP01240  
PSP01250  
PSP01260  
PSP01270  
PSP01280  
PSP01290  
PSP01300  
PSP01310  
PSP01320  
PSP01330  
PSP01340  
PSP01350  
PSP01360  
PSP01370  
PSP01380  
PSP01390  
PSP01400  
PSP01410  
PSP01420  
PSP01430  
PSP01440  
PSP01450  
PSP01460  
PSP01470  
PSP01480  
PSP01490  
PSP01500  
PSP01510  
PSP01520  
PSP01530  
PSP01540

ORIGINAL PAGE IS  
OF POOR QUALITY

FILE: RDDPAT

```

C* THIS SUBROUTINE COORDINATES THE ROUTINES TO READ FIELDS OF DATA FROM THE IMAGE TAPE AND STORE IT ON A DRUM FILE FOR THE ISOCLE ROUTINES.
C*
C* SUBROUTINE RDDPAT(FDI, TOP, IDATA, IDIM, LAST)
C* IMPLICIT INTEGER (A-Z)
C* DIMENSION FLDINF(6), IDATA(10IM), FL(12), LSTAT(3)
C* COMMON ARRAY(10600)
C* LOGICAL*1 LARRAY(42400)
C* EQUIVALENCE (ARRAY, LARRAY)
C* LOGICAL*1 LPACK(4)
C* EQUIVALENCE (LPACK, IPACK)
C
C INCLUDE COM-KEYS.LIST
C INCLUDE COM-KEYS.LIST
COMMON/PARS/STOP, LNCAT, NMIN, KRY, STOMAX, DLMIN, SEP,
* MAP, SPTPIG, IFO, KPTS, NOPTS, PUNCH,
* ICHN, CHNTHS, ICHAIN(62), NPOS, IREGIN, REGIN1,
* REGIN2, REGIN3, CLSNAM, NOFLD, IPT, TOTWRD, TOTPTS,
* NCLASS, NOCLS, TOTSUH, TOTFLD, TOTVRT, NOCL, NVRT
* ,NXTCLS, NOFFAT, MAXCLS, FFTVEC(30), SYMATX(62)
* ,VARSI2, STATKY, ISKEY, ASFMT, MAPKEY, SEQUEN(20), PERCFN, SIMERP
* ,IORDER, INDIRT, INFILE, INIT4, PFIU, SUSVEC(62), NOSUB2, CHNVC(30)
* ,NOCHAN, FRCOMP, NOSE0, REAND0, REANDU,
* SYND0, SYNDU, ITRIG0, ITRIGU, DOFLAG,
* DUELAG, NDOU, STDOTS(60), NSDOTS, SINCOR(30), LLNCAT,
* DVERT(250, 2), DRECT(60, 2), DVPLT(11, 2), IDCNT(2), NDOU(2)
* ,MAXFT1, MAXPOP
REAL SINCOR
COMMON/GLOBAL/HEAD(63), MAPTAP, DATAP, SAVTAP, BMFILE, HMKEY,
* HISFIL, HISKEY, TIFORM, ERIPTP, E-PRKY, MAPUNT, NOFILE,
* ,OPRMAN, OPRMDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
* ,NHSTUN, NHSTFI, SCTRUN, MAPFIL
* ,DOTUNT, DOTFIL, NCHPAS, TRNSFL, RMTRFL, HISTFL, PCHUNT,
* CRDUNT, PRUNT, RANDIO
CSEND
PRUNIT = 30
DIMENSION C40(20)
EQUIVALENCE (FLDINF(1), LINSTR), (FLDINF(4), SAMSTR),
* (FLDINF(2), LINEAD), (FLDINF(5), SAMEND),
* (FLDINF(3), LININC), (FLDINF(6), SAMINC)
DATA LORN/'/'
DIMENSION LND(2), LDOU(11, 2), ID3(35), IDE(35), NDINT(11, 2)
DATA DORNAME/'OTHE'
DATA DORNAME/'UNTD'
DIMENSION FLDSAV(4, 10), VERTEX(220)
C*
C* RESERVE 2000 LOCATIONS OF 'ARRAY' FOR FIELD DEFINITION INFORMATION.
C* THE REMAINDER OF 'ARRAY' IS USED FOR I/O BUFFERS.
C*
C* CLASS AND FIELD INFORMATION STORED AS FOLLOWS
C*
C* ARRAY(1) =CLASS NAME
C* ARRAY(2) =RESERVED FOR INDEX POINTER TO NEXT CLASS NAME
C* ARRAY(3) =RESERVED FOR NO. OF CLUSTERS IN THIS CLASS
C* ARRAY(4) =NO. OF FIELDS FOR THIS CLASS
C* ARRAY(5) =FIRST FIELD NAME FOR THIS CLASS
C* (6) =NO. OF VERTICES FOR THIS FIELD (NV)
C* (7)-(7+NV*2) = ACTUAL VERTEX NUMBERS
C* (8+NV*2) =TOTAL PIXELS IN THIS FIELD
C* (9+NV*2)-(10+NV*2) = FLDINF BLOCK FOR THIS FIELD
C*
CALL TA-HDD(DATAP, DATFIL)
1 CONTINUE
RESERV=2000
ADDRESS=REGIN1
IIND=1
IND=0
NVRT=0
LAST=0
TOTWRD=0
IND=0
DOFLAG = 0
DOFLAG = 0
DOU = 0
NDOU(1) = 0
NDOU(2) = 0
2 RFINDX=RESERV+1
NHUFS=1

```

FILE: RDDPAT

```
MAXDIM=TOP-RESERV
MUSIZ= MAXDIM/(NMFES*NOFFAT) * NOFFAT
MUSIZ = (MUSIZ/4)*4
IF (-MUSIZ.GT. 100)GO TO 3
RESERV=RESERV-100
IF (RESERV.GT.30 )GO TO 2
GO TO 70
3 CONTINUE
NOFL(0)=0
IPT=1 + FDI - 1
TOTVT2=0
IF (NOCL.EQ.0)GO TO 5
4 ARRAY(IPT)=NXTCLS
IPT=IPT+4
WRITE (6,HEAD)
WRITE (6,500)NXTCLS
C*
C* READ A FIELD DESCRIPTION FROM CARDS.
C*
5 ICK = LAREAD(ARRAY(IPT),ARRAY(IPT+2),FLDIN,ARRAY(IPT+1) )
IF (ICK.NE.-3) GO TO 1000
WRITE (6,140)
READ (RUNIT,150) (CARD(I), I=1,20)
WRITE (6,150) (CARD(I), I=1,20)
150 FORMAT(20A4)
160 FORMAT(1X,20A4)
REWIND RUNIT
IDP=IDP+1
IDCNT(IDP)=0
DVPNT(1,IDP)=1
READ(30,100) DNAME
REWIND 30
IF (DNAME.EQ.DNAME) ITRIGU=1
IF (DNAME.EQ.DNAME) ITRIGU=1
IF (DNAME.EQ.DNAME) IS=2
IF (DNAME.EQ.DNAME) IS=1
INDV=1
INDP=1
GO TO 5
1000 IF (ICK.LE.0.OR.IDP.LE.0) GO TO 1030
IF (IDCNT(IDP).LT.10) GO TO 1025
WRITE (6,170)
170 FORMAT(// * TOO MANY DO OR DU FIELDS THESE IGNORED*)
GO TO 5
1025 CONTINUE
READ (RUNIT,150) (CARD(I), I=1,20)
WRITE (6,150) (CARD(I), I=1,20)
REWIND RUNIT
OVERT(INDV,IDP) = ARRAY(IPT + 1)
IDLIM = OVERT(INDV,IDP)*2
DO 1010 I=1,IDLIM
INDV=INDV+1
VERTEX(TOTVT2+1)=ARRAY(IPT+1+1)
1010 OVERT(INDV,IDP) = ARRAY(IPT + I + 1)
INDV = INDV + 1
TOTVT2=TOTVT2+IDLIM
DO 1020 I=1,6
DRECT(INDP,IDP)=FLDINF(I)
INDP=INDP+1
1020 IDCNT(IDP)=IDCNT(IDP)+1
IDUP=IDCNT(IDP)+1
DVPNT(IDUP,IDP)=INDV
GO TO 5
C
1030 FINISHED WITH DO/DU FIELD PROCESSING
CONTINUE
IDP=0
IDPR=IT-IGO+ITRIGU
IF (ICK.LT.0)GO TO 20
IF (ICK.EQ.0)GO TO 30
IF (NOCL.GT.0)GO TO 6
WRITE (6,400)
CALL CHRP
6 CONTINUE
NV=ARRAY(IPT+1)
NVRT=NVRT+NV
NOFL=NOFL+1
NSAMP=(SAMEND-SAMSTR)/SAMINC+1
FLOSAM=0
I=IPT+2
RDD00800
RDD00810
RDD00820
RDD00830
RDD00840
RDD00850
RDD00860
RDD00870
RDD00880
RDD00890
RDD00900
RDD00910
RDD00920
RDD00930
RDD00940
RDD00950
RDD00960
RDD00970
RDD00980
RDD00990
RDD01000
RDD01010
RDD01020
RDD01030
RDD01040
RDD01050
RDD01060
RDD01070
RDD01080
RDD01090
RDD01100
RDD01110
RDD01120
RDD01130
RDD01140
RDD01150
RDD01160
RDD01170
RDD01180
RDD01190
RDD01200
RDD01210
RDD01220
RDD01230
RDD01240
RDD01250
RDD01260
RDD01270
RDD01280
RDD01290
RDD01300
RDD01310
RDD01320
RDD01330
RDD01340
RDD01350
RDD01360
RDD01370
RDD01380
RDD01390
RDD01400
RDD01410
RDD01420
RDD01430
RDD01440
RDD01450
RDD01460
RDD01470
RDD01480
RDD01490
RDD01500
RDD01510
RDD01520
RDD01530
RDD01540
RDD01550
RDD01560
RDD01570
RDD01580
```

FILE: RDDPAT

```

      NO=NV-1
      NO=NO-5
      IF (NO.GT.5) NO=5
      IF=IH+NO*2 - 1
      WRITE (6,600) NOFLD,ARRAY(IPT),SAMINC,LININC
      * (LPRN,ARRAY(I),ARRAY(I+1),I=IB,IE,2)
      IF (NR.LE.0) GO TO 7
      IR=IF+1
      IF=IH+NR*2 - 1
      WRITE (6,650) (LPRN,ARRAY(I),ARRAY(I+1),I=IB,IE,2)
7 CONTINUE
      IF (NSAMP*NOFEAT.GT.IDIM) GO TO 90
C*
C* POSITION TAPE FOR THIS FIELD
C*
      CALL FLDINT(FLDINF,FETVEC,NOFEAT)
      FLDOSAM=0
      DO 10 LINE=LINSTR,LINEND,LININC
      LND(1)=0
      LND(2)=0
      IDRR = 2
      IDEE=1
      IF (IDRR.EQ.0) GO TO 1095
C BOTH DO AND DU TRIGGERS OFF --- SKIP AROUND
      DO 1060 I=1,INDP
      IDLIM=IDCNT(I,ND)
      DO 1050 J=1,IDLIM
      IDUM=(J-1)*6
      LDSTR=OVERT(IDUM+1,IND)
      LDEND=OVERT(IDUM+2,IND)
      LDINC=OVERT(IDUM+3,IND)
      DO 1040 IT = LDSTR,LDEND,LDINC
      IF (IT.NE.I) GO TO 1040
      LND(IND)=LND(IND)+1
      IDUM=LND(IND)
      LDUM(IDUM,IND) = I
1040 CONTINUE
1050 CONTINUE
1060 CONTINUE
      IF (LND(1).EQ.0.AND.LND(INDP).EQ.0) GO TO 1095
C NO DO OR DU FOR THIS LINE
      IF (LND(1).GT.0) IDRR=1
      IF (IDRR.EQ.2.AND.LND(2).GT.0) IDEE=2
      DO 1090 I=1,INDP,IDEE
      IDLIM=LND(I,IND)
      IF (IDLIM.EQ.0) GO TO 1090
      IDUM=0
      DPRINT(1,1) = 1
      DPRINT(1,2) = 1
      DO 1080 J=1,IDLIM
      IDP=IDPRT(I,IND)
      OVP=OVPRT(IDP,IND)
      CALL FOLINT(OVERT(OVP+1,IND),OVERT(OVP,IND),FL,LINE,SAMPS,NI)
      DPRINT(I,IND)=NI
      IF (NI.EQ.0) GO TO 1080
      DO 1070 JJ=1,NI
      DINT(JJ+IDUM,IND)=FL(JJ)
      IDUM=IDUM+NI
      DPRINT(I+1,IND) = IDUM + 1
1070 CONTINUE
1080 CONTINUE
1090 CONTINUE
1095 CALL LINEFD(I,DATA,ENDTAP)
      IF (ENDTAP.EQ.-1) GO TO 80
C*
C* FIND SAMPLE INTERSECTS FOR THIS LINE - NI=NO. OF INTERSECTS
C*
      CALL FOLINT(ARRAY(IPT+2),DV,FL,LINE,SAMPS,NI)
C*
C* STORE DATA ON THIS LINE INTO OUTPUT BUFFER
C*
      MODSS=MOD(SAMSTR,SAMINC)
      DO 60 J=1,NI,2
      IP=(FL(J)-SAMSTR)/SAMINC+1
      IF=(FL(J+1)-SAMSTR)/SAMINC+1
      IF (MODSS.NE.MOD(FL(J),SAMINC)) IP=IB+1
      IF (IP.GT.IE) GO TO 60
      IF (IDRR.EQ.0) GO TO 2055
      IF (LND(IDRR).EQ.0.AND.LND(IDRR).EQ.0) GO TO 2055
      RDD01590
      RDD01600
      RDD01610
      RDD01620
      RDD01630
      RDD01640
      RDD01650
      RDD01660
      RDD01670
      RDD01680
      RDD01690
      RDD01700
      RDD01710
      RDD01720
      RDD01730
      RDD01740
      RDD01750
      RDD01760
      RDD01770
      RDD01780
      RDD01790
      RDD01800
      RDD01810
      RDD01820
      RDD01830
      RDD01840
      RDD01850
      RDD01860
      RDD01870
      RDD01880
      RDD01890
      RDD01900
      RDD01910
      RDD01920
      RDD01930
      RDD01940
      RDD01950
      RDD01960
      RDD01970
      RDD01980
      RDD01990
      RDD02000
      RDD02010
      RDD02020
      RDD02030
      RDD02040
      RDD02050
      RDD02060
      RDD02070
      RDD02080
      RDD02090
      RDD02100
      RDD02110
      RDD02120
      RDD02130
      RDD02140
      RDD02150
      RDD02160
      RDD02170
      RDD02180
      RDD02190
      RDD02200
      RDD02210
      RDD02220
      RDD02230
      RDD02240
      RDD02250
      RDD02260
      RDD02270
      RDD02280
      RDD02290
      RDD02300
      RDD02310
      RDD02320
      RDD02330
      RDD02340
      RDD02350
      RDD02360
      RDD02370

```



FILE: RDDPAT

ORIGINAL PAGE IS  
OF POOR QUALITY

```
DO 2050 I=0,IOBR,IOEE
IDLIM=IND(INO)
IF (IDLIM.EQ.0) GOTO 2050
IDSIT=1
MEANDD=MEANDD
IF (IDPR.EQ.IDFE) GOTO 2003
IF (IND.EQ.2) IDSIT=2
IF (IND.EQ.2) MEANDD=MEANDD
GOTO 2009
2003 IF (IDPR.EQ.1.AND.IDPP.EQ.2) GOTO 2009
IF (ITPI.EQ.0) GOTO 2009
IDSIT=2
MEANDD=MEANDD
2009 CONTINUE
DO 2040 K=1,IDLIM
NDIN=NDINT(K,IND)
IF (NDIN.EQ.0) GOTO 2040
OPIN=OPINT(K,IND)
DO 2010 KK=1,NDIN
2010 OTN(KK)=DINT(OPIN+KK-.1,IND)
IDUM=0
DO 2020 K=1,NDIN,2
IDUM=IDUM+1
IDR(IDUM)=(DIN(KK)-SAMSTR)/SAMINC+1
IDF(IDUM)=(DIN(KK+1)-SAMSTR)/SAMINC+1
IF (MOD(SAM,MOD(DIN(KK),SAMINC))) IDB(IDUM)=IDR(IDUM)+1
2020 CONTINUE
DO 2030 KK=1,IDUM
IDS=IDR(KK)
IDF=IDF(KK)
IF (IDS.GT.IDF) ID=IDS
IF (IDS.LE.IDF) ID=IDF
IF (IDF.GT.IDF) ID=IDF
IF (IDS.GT.IDF) GOTO 2030
DO 2025 KKK=IDS,IDF
NDUU(IND)=NDUU(IND)+1
DO 2023 KKK=1,NOFEAT
DUMMY1=KKK+NSAMP*(KKK-1)
2023 IDATA(DUMMY1)=MEANDD
2025 CONTINUE
2030 CONTINUE
2040 CONTINUE
IF (IDSIT.EQ.1.AND.NDUU(IND).GT.0) DOFLAG=1
IF (IDSIT.EQ.2.AND.NDUU(IND).GT.0) DUFLAG=1
2050 CONTINUE
2055 DDU=DOFLAG+DUFLAG
IRASF = (RFINDX - 1)*4
IRUF4 = RUESIZ*4
NRDS4 = NRDS*4
DO 50 J=1,IF
FLDSAM=FLDSAM+1
DO 50 K=1,NOFEAT
IWRD=IWRD+1
DUMMY2=J+NSAMP*(K-1)
IPACK = IDATA(DUMMY2)
DUMMY3 = IRASF + IWRD
LARRAY(DUMMY3) = IPACK(4)
IF (IWRD.LT.IRUF4) GO TO 50
TOTWRD = TOTWRD + IWRD
IF (TOTWRD.GT.NRDS4) GO TO 35
CALL WRITE(ADDRESS,ARRAY(RFINDX),RUESIZ,LSTAT(1))
ADDRESS = ADDRESS + RUESIZ
40 IF (LSTAT(IRUF).EQ.1) GOTO 40
IWRD=0
50 CONTINUE
60 CONTINUE
10 CONTINUE
IPT = IPT + NV*2 + 2
ARRAY(IPT)=FLDSAM
DO 15 I=1,4
IPT=IPT+1
15 ARRAY(IPT)=FLDINF(I)
IPT=IPT+1
IF (IPT+30 .GT. RESERV) GO TO 70
GO TO 5

***
CLASS NAME CARD ENCOUNTERED - REFEAD PREVIOUS CARD TO GET NAME
20 NOCL=NOCL+1
RDD02380
RDD02390
RDD02400
RDD02410
RDD02420
RDD02430
RDD02440
RDD02450
RDD02460
RDD02470
RDD02480
RDD02490
RDD02500
RDD02510
RDD02520
RDD02530
RDD02540
RDD02550
RDD02560
RDD02570
RDD02580
RDD02590
RDD02600
RDD02610
RDD02620
RDD02630
RDD02640
RDD02650
RDD02660
RDD02670
RDD02680
RDD02690
RDD02700
RDD02710
RDD02720
RDD02730
RDD02740
RDD02750
RDD02760
RDD02770
RDD02780
RDD02790
RDD02800
RDD02810
RDD02820
RDD02830
RDD02840
RDD02850
RDD02860
RDD02870
RDD02880
RDD02890
RDD02900
RDD02910
RDD02920
RDD02930
RDD02940
RDD02950
RDD02960
RDD02970
RDD02980
RDD02990
RDD03000
RDD03010
RDD03020
RDD03030
RDD03040
RDD03050
RDD03060
RDD03070
RDD03080
RDD03090
RDD03100
RDD03110
RDD03120
RDD03130
RDD03140
RDD03150
RDD03160
```

FILE: R00PAT

```
IF (NOCL.GT.1) GO TO 25
READ(30,100)NXTCLS
REWIND 30
GO TO 4
25 CLSNAM=NXTCLS
READ(30,100)NXTCLS
REWIND 30
GO TO 31
C* EMPTY LAST BUFFER AND RETURN TO PROCESS DATA FOR THIS CLASS.
30 CLSNAM=NXTCLS
LAST=1
C*
31 TOTWRD=TOTWRD+IWRD
IF (TOTWRD.GT.NWDS) GOTO 35
IWRD4 = IWRD/4
IF (4*IWRD4.NE.IWRD) IWRD4 = IWRD4 + 1
CALL WRITE(A00PES,ARRAY(4*INDEX),IWRD4*LSTAT(1))
TOTPTS=TOTWRD/NOFEAT
IPT = IPT - F01 + 1
IPAT = (TOTWRD/4) + 1
IF (IPAT+TOTPTS.LE.NWDS) RETURN
WRITE(6,200)N*5
IPT = IPT - F01 + 1
RETURN
35 WRITE(6,200)NWDS
CALL CMFRD
70 WRITE(6,300)RESERV
CALL CMFRD
90 WRITE(6,400)
CALL CMFRD
90 WRITE(6,700)IDIM
CALL CMFRD
100 FORMAT(10X,A4)
140 FORMAT(/// * DESIGNATED OTHER OR UNIDENTIFIABLE FIELDS INPUT*/)
200 FORMAT(' * TOO MUCH DATA REQUESTED--PIXELS*(CHANNELS+1) CANNOT EXCEED
*0',I10)
300 FORMAT(' * STORAGE REQUIRED FOR FIELD DEFINITION INFORMATION EXCEEDS
* THE DIMENSION LIMIT OF',I5)
400 FORMAT(' * END-OF-TAPE REACHED BEFORE END OF FIELD')
500 FORMAT(//40X,' FIELDS TO BE CLUSTERED FOR CLASS',I4,A4//
* T36,'SAMPLE',T45,'LINE',T20,'FIELD NAME',T36,'INC.',
* T45,'INC.',T73,'VERTICES (SAMPLE,LINE)')
600 FORMAT(1X,I16,I3,I22,A4,T36,I4,T45,I4,T60,
* 5(I3,I4,'',I4,'')',I4))
650 FORMAT(1X,I60,5(A1,I4,'',I4,'')',I4))
700 FORMAT(' * NO. OF PIXELS TO BE UNPACKED PER SCAN EXCEEDS THE DIMENSION
* LIMIT OF',I5)
800 FORMAT(// * INPUT ERROR - A CLASSNAME CARD MUST BE INPUT BEFORE A GROUP
* OF FIELDS*/)
RETURN
END
```

R0003170  
R0003180  
R0003190  
R0003200  
R0003210  
R0003220  
R0003230  
R0003240  
R0003250  
R0003260  
R0003270  
R0003280  
R0003290  
R0003300  
R0003310  
R0003320  
R0003330  
R0003340  
R0003350  
R0003360  
R0003370  
R0003380  
R0003390  
R0003400  
R0003410  
R0003420  
R0003430  
R0003440  
R0003450  
R0003460  
R0003470  
R0003480  
R0003490  
R0003500  
R0003510  
R0003520  
R0003530  
R0003540  
R0003550  
R0003560  
R0003570  
R0003580  
R0003590  
R0003600  
R0003610  
R0003620  
R0003630  
R0003640  
R0003650  
R0003660  
R0003670