

DORMAN Computer Program (Study 2.5) Final Report Volume IV DORMAN Program Listing, UNIVAC 1108 Version

Prepared by DATA PROCESSING SUBDIVISION
Information Processing Division

11 September 1973

(NASA-CR-137358) DORMAN COMPUTER PROGRAM
(STUDY 2.5). VOLUME 4: DORMAN PROGRAM
LISTING, UNIVAC 1108 VERSION Final
Report (Aerospace Corp., El Segundo,
Calif.) 97 p HC \$8.00

CSCL 09B G3/08

N74-19830

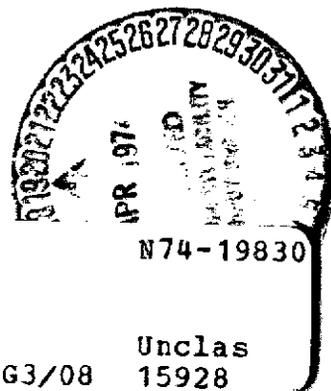
Unclas
15928

Prepared for OFFICE OF MANNED SPACE FLIGHT
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D. C.

Contract No. NASW-2472



Engineering Science Operations
THE AEROSPACE CORPORATION



Aerospace Report No.
ATR-74(7335)-1, Vol. IV

DORMAN COMPUTER PROGRAM (STUDY 2.5) FINAL REPORT
Volume IV. DORMAN Program Listing, UNIVAC 1108 Version

Prepared by
Data Processing Subdivision
Information Processing Division

11 September 1973

Engineering Science Operations
THE AEROSPACE CORPORATION
El Segundo, California

Prepared for
OFFICE OF MANNED SPACE FLIGHT
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D. C.

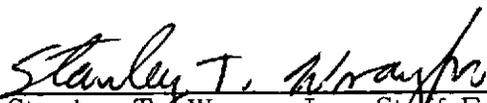
Contract No. NASW-2472

;

DORMAN COMPUTER PROGRAM (STUDY 2.5) FINAL REPORT

Volume IV: DORMAN Program Listing, UNIVAC 1108 Version

Prepared by



Stanley T. Wray, Jr., Staff Engineer
Advanced Application Staff
Data Processing Subdivision

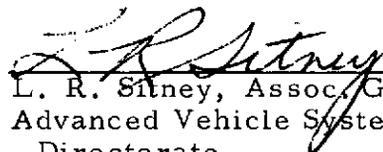


L. Sashkin, Director
Data Processing Subdivision
Information Processing Division

Approved by



Robert R. Wolfe
NASA Study Director
Advanced Vehicle Systems
Directorate
Systems Planning Division



L. R. Silney, Assoc. Group Director
Advanced Vehicle Systems
Directorate
Systems Planning Division

PRECEDING PAGE BLANK NOT FILMED

FOREWORD

Study 2.5, DORCA Applications, has been directed at development of a data bank management computer program identified as DORMAN. The size of the DORCA data files and the manipulations required on that data to support analyses with the DORCA program necessitates automated data techniques to replace time consuming manual input generation. The DORCA program (Dynamic Operations Requirements and Cost Analysis) was developed by The Aerospace Corporation for use by NASA in planning future space programs. Both programs are designed for implementation on the UNIVAC 1108 computing system at the NASA Computing Facility, Slidell, Louisiana.

This volume contains a listing of the UNIVAC 1108 version of the DORMAN program. The code printed herein has been compiled, loaded, and executed successfully on the EXEC 8 system for the UNIVAC 1108 at Slidell, Louisiana. This was accomplished during the month of September (1973) using the segment map contained in Volume II of this report.

In addition to this volume, the following documentation has been prepared.

| | |
|------------|-------------------------------------|
| Volume I | Executive Summary |
| Volume II | User's Guide and Programmer's Guide |
| Volume III | Original Data Bank Listing |

Study 2.5, DORCA Applications, is one of several study tasks conducted under NASA Contract NASW-2472 in FY 1973. The NASA Study Director was Mr. V. N. Huff, NASA Headquarters, Code MTE.

By agreement with Mr. Huff, the DORMAN program will be delivered directly to the NASA Computing Facility.

```

*ELT,I DORMAN.BLOCK
BLOCK DATA
COMMON /BFRS/ ISYN(2,20),ISYN1,IT1,IT2,IT3,FULL,CN1,CN2,LIN1,LIN2
* ,NWBUF,NB1,NB2,ICN1,ICN2
COMMON /BFRS/ BUF1(14,50),BUF2(14,50),AAA(700)
INTEGER CN1,CN2,BUF1,BUF2
LOGICAL FULL
COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14)
INTEGER ERFLAG,FERR,ACTION,PRTFIL
EQUIVALENCE (PRTFIL,FERR)
COMMON /NAMES/ IVER,DNAME(2),MODNAM(2),BNAME(2)
INTEGER DNAME,MODNAM,BNAME
COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2
INTEGER BASIC,MTAPE,FINAL,S1,S2
COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH
INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH
COMMON/VCARD/ILBL(14)
COMMON /WORK/IIUNT(3),IIRW(3),IUTBL(3,19),IACT(8,3),NFILES
DATA (ILBL(I),I=1,14) /
* 6H$DORMA, 6HN DATA, 6H BANK , 6H , 6HVERSIO, 6HN ,
* 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
DATA FINAL,BASIC/14,1/,PRTFIL/6 /
DATA USES, BLANK, END / 6H USES , 6H , 6H$END 0 /
DATA TABLE/6HF TABL/,DECK/6HF DECK/,DDECK/6H$DECK /,FILE/6HF FILE/BLOCK
DATA BATCH/0/
DATA IUTBL/
* 0,0,0, 1,2,0, 2,2,2, 3,2,2,
* 4,2,1, 5,3,0, 6,2,1,
*11,2,2, 12,2,2, 13,2,2, 14,2,2,
*21,2,2, 22,2,2, 23,2,2, 24,2,2,
*,25,2,2, 26,2,2, 27,2,2 , 20,2,2
* /
DATA NFILES/19/
DATA IVER /4HTEMP/
DATA IACT/24*0/
END

```

```

BLOCK 3
BLOCK 5
BFRS1 2
BFRS1 3
BFRS1 4
BFRS1 5
BFRS1 6
MISC 2
MISC 3
MISC 4
NAMES 2
NAMES 3
FILES 2
FILES 3
REST 2
REST 3
BLOCK 11
WORK 2
BLOCK 13
BLOCK 14
BLOCK 15
BLOCK 16
BLOCK 17
BLOCK 18
BLOCK 19
BLOCK 20
BLOCK 21
BLOCK 22
BLOCK 23
BLOCK 24
BLOCK 25
BLOCK 26
BLOCK 27
BLOCK 28
GWT25 2
BLOCK 29

```

#ELT,I DORMAN.DORMAN

```
DIMENSION IU(3),IR(3)
INTEGER  CONTEN,GENEOL
INTEGER  CREATE,USE,OPTION,SAVE,ADD,DELETE,EDIT
INTEGER  CONVER,REPLAC,LIST,DONE
COMMON  /MISC/ERFLAG,FERR,KARD(14),ACTION(14)
INTEGER  ERFLAG,FERR,ACTION,PRTFIL
EQUIVALENCE (PRTFIL,FERR)
COMMON  /FILES/ BASIC,MTAPE,FINAL,S1,S2
INTEGER  BASIC,MTAPE,FINAL,S1,S2
COMMON  /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH
INTEGER  TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH
DATA  INPUT/5/,CREATE/6HCREATE/,OPTION/6HOPTION/,SAVE/4HSAVE/
DATA  ADD/3HADD/,DELETE/6HDELETE/,EDIT/4HEDIT/
DATA  CONVER/6HCONVER/,REPLAC/6HREPLAC/,LIST/4HLIST/,DONE/4HDONE/
DATA  CONTEN/6HCONTEN/,GENEOL/6HGENEOL/
DATA  USE/3HUSE/
DATA  IU/5,0,0/
DATA  IR/2,0,0/
DATA  IFLAG/0/
```

DORMAN 3
DORMAN13
DORMAN14
DORMAN15
DORMAN16
MISC 2
MISC 3
MISC 4
FILES 2
FILES 3
REST 2
REST 3
DORMAN20
DORMAN21
DORMAN22
DORMAN23
DORMAN24
DORMAN25
DORMAN26
DORMAN27
DORMAN28
DORMAN29
DORMAN30
DORMAN31
DORMAN32
DORMAN33
DORMAN34
DORMAN35
DORMAN36
DORMAN37
DORMAN38
DORMAN39
DORMAN40
DORMAN41
DORMAN42
DORMAN43

-2-

```
C
C  MAIN PROGRAM
C
C  PROGRAMMER - S. WRAY
C
C
C  RESERVED FILES
C
C      TAPE 1 - INPUT DATA BANK
C      TAPE 4 - OUTPUT DATA BANK
C      TAPE 12 - DORCA DATA DECK
C      TAPE 13 - MOD DECKS -INPUT
C      TAPE 14 - BASIC DECK
C      TAPE 21 - INPUT BASIC DECKS
C      TAPE 24 - MOD DECKS - OUTPUT
C
```

| | |
|--|----------|
| WRITE(PRTFIL,5) | DORMAN48 |
| 5 FORMAT(21H1 -- BEGIN DORMAN --//34H DO YOU WISH TO CREATE A DATA | DORMAN49 |
| 1 FILE/22H OR TO USE SUCH A FILE/20H ENTER CREATE OR USE) | DORMAN50 |
| 10 WRITE(PRTFIL,15) | DORMAN51 |
| 15 FORMAT(24H)ENTER OPTION REQUEST -) | DORMAN52 |
| CALL INTBUF(IU,IR) | DORMAN53 |
| REWIND BASIC | DORMAN54 |
| CALL INCD (KARD,INPUT) | DORMAN55 |
| IF(IFLAG.NE.0) GO TO 17 | DORMAN56 |
| IF(KARD(1).EQ.CREATE) GO TO 20 | DORMAN57 |
| 17 IF(KARD(1).EQ.1HU) GO TO 30 | DORMAN58 |
| IF(KARD(1).EQ.USE) GO TO 30 | DORMAN59 |
| IF(IFLAG.EQ.0) GO TO 18 | DORMAN60 |
| IF(KARD(1).EQ.1HO) GO TO 25 | DORMAN61 |
| IF(KARD(1).EQ.1HS) GO TO 40 | DORMAN62 |
| IF(KARD(1).EQ.1HA) GO TO 35 | DORMAN63 |
| IF(KARD(1).EQ.1HD) GO TO 45 | DORMAN64 |
| IF(KARD(1).EQ.1HE) GO TO 50 | DORMAN65 |
| IF(KARD(1).EQ.1HC) GO TO 55 | DORMAN66 |
| IF(KARD(1).EQ.1HR) GO TO 60 | DORMAN67 |
| IF(KARD(1).EQ.1HL) GO TO 65 | DORMAN68 |
| IF(KARD(1).EQ.OPTION) GO TO 25 | DORMAN69 |
| IF(KARD(1).EQ.SAVE) GO TO 40 | DORMAN70 |
| IF(KARD(1).EQ.ADD) GO TO 35 | DORMAN71 |
| IF(KARD(1).EQ.DELETE) GO TO 45 | DORMAN72 |
| IF(KARD(1).EQ.EDIT) GO TO 50 | DORMAN73 |
| IF(KARD(1).EQ.CONVER) GO TO 55 | DORMAN74 |
| IF(KARD(1).EQ.REPLAC) GO TO 60 | DORMAN75 |
| IF(KARD(1).EQ.LIST) GO TO 65 | DORMAN76 |
| IF(KARD(1).EQ.DONE) GO TO 70 | DORMAN77 |
| IF(KARD(1).EQ.SHATCH) GO TO 80 | DORMAN78 |
| 18 CONTINUE | DORMAN79 |
| WRITE (PRTFIL,16) | DORMAN80 |
| 16 FORMAT(40H COMMAND NOT UNDERSTOOD - PLEASE RETRY) | DORMAN81 |
| IF(ERFLAG.NE.0) CALL TERM | DORMAN82 |
| ERFLAG=1 | DORMAN83 |

```

        GO TO 100
65 CONTINUE
    CALL LISTER
    GO TO 100
70 CONTINUE
    CALL TERM
    GO TO 100
20 CONTINUE
    BASIC = 2
    CALL INCR
    IFLAG = 1
    GO TO 100
25 CONTINUE
    CALL OPT
    GO TO 100
30 CONTINUE
    CALL USER (IFLAG)
    IFLAG = 1
    GO TO 100
40 CONTINUE
    CALL SAVER
    GO TO 100
35 CONTINUE
    CALL ADDER
    GO TO 100
45 CONTINUE
    CALL DELET
    GO TO 100
50 CONTINUE
    CALL EDITER
    GO TO 100
55 CONTINUE
    CALL CONV
    GO TO 100
60 CONTINUE
    CALL REPL

```

```

DORMAN84
UORMAN85
DORMAN86
DORMAN87
DORMAN88
DORMAN89
DORMAN90
UORMAN91
DORMAN92
DORMAN93
UORMAN94
DORMAN95
DORMAN96
DORMAN97
DORMAN98
UORMAN99
DORMA100
DORMA101
DORMA102
DORMA103
DORMA104
DORMA105
DORMA106
DORMA107
DORMA108
DORMA109
DORMA110
DORMA111
DORMA112
DORMA113
DORMA114
DORMA115
DORMA116
DORMA117
DORMA118
DORMA119

```

| | |
|---|----------|
| GO TO 100 | DORMA120 |
| 80 CONTINUE | DORMA121 |
| BATCH = 5 | DORMA122 |
| GO TO 100 | DORMA123 |
| 100 ERFLAG=0 | DORMA124 |
| GO TO 10 | DORMA127 |
| END | DORMA128 |
| *ELT,I DORMAN,ADD | ADD 3 |
| SUBROUTINE ADD(IN1,IN2,OUT) | ADD 5 |
| C ADD A DECK (WHICH RESIDES ON FILE IN2) TO THE DATA FILE (IN1) AND | ADD 6 |
| C PUT EXPANDED DATA ON FILE OUT. | ADD 7 |
| C BY B.J. GOLD | ADD 8 |
| C | ADD 9 |
| COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC 2 |
| INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC 3 |
| EQUIVALENCE (PRTFIL,FERR) | MISC 4 |
| INTEGER OUT | ADD 11 |
| INTEGER TYPE, FLAG, TEMP(14) | ADD 12 |
| COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST 2 |
| INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST 3 |
| C | ADD 14 |
| C INITIALIZE FILES. READ FIRST CARD OF NEW DECK AND VERSION CARD FROM | ADD 15 |
| C DATA FILE. | ADD 16 |
| C | ADD 17 |
| FLAG = 1 | ADD 18 |
| REWIND IN1 | ADD 19 |
| REWIND OUT | ADD 20 |
| ERFLAG = 0 | ADD 21 |
| CALL INGD(KARD,IN2) | ADD 22 |
| KOUNT2 = 1 | ADD 23 |
| IF (ERFLAG.NE.0) GO TO 100 | ADD 24 |
| TYPE = 5HBASIC | ADD 25 |
| IF (KARD(4).EQ.USES) TYPE = 3HMOD | ADD 26 |
| IF (KARD(1).NE.DDECK) GO TO 110 | ADD 27 |
| CALL EXTRAN (KARD(7),KARD(9)) | ADD 28 |
| WRITE (PRTFIL,10) KARD(2), KARD(3) | ADD 29 |

| | | | |
|----|---|-----|----|
| 10 | FORMAT(16H0ADD DECK NAMED ,2A6,13H TO DATA FILE) | ADD | 30 |
| | CALL OUTCD (KARD,PRTFIL) | ADD | 31 |
| | IF ((KARD(2).EQ.KARD(5)).AND.(KARD(3).EQ.KARD(6))) GO TO 150 | ADD | 32 |
| | CALL INCD(TEMP,IN1) | ADD | 33 |
| | IF (ERFLAG.NE.0) GO TO 120 | ADD | 34 |
| | KOUNT1 = 1 | ADD | 35 |
| | CALL OUTCD(TEMP,OUT) | ADD | 36 |
| C | | ADD | 37 |
| C | INSERT NEW ENTRY INTO TABLE OF CONTENTS AT BEGINNING IF IT IS A MOD | ADD | 38 |
| C | DECK, AT END IF A BASIC DECK. | ADD | 39 |
| C | VERIFY THAT NEW DECK NAME IS NOT ALREADY IN TABLE OF CONTENTS. | ADD | 40 |
| C | IF NEW DECK IS A MOD DECK, VERIFY THAT THE DECK IT USES IS ALREADY | ADD | 41 |
| C | ON THE DATA FILE. | ADD | 42 |
| C | | ADD | 43 |
| | FLAG = 2 | ADD | 44 |
| | IF (TYPE.EQ.3HMOD) CALL OUTCD(KARD,OUT) | ADD | 45 |
| | NEED = 0 | ADD | 46 |
| | LASTM = 0 | ADD | 47 |
| | NTC = 0 | ADD | 48 |
| 30 | CALL INCD(TEMP,IN1) | ADD | 49 |
| | IF (ERFLAG.NE.0) GO TO 120 | ADD | 50 |
| | KOUNT1 = KOUNT1+1 | ADD | 51 |
| | IF (TEMP(1).EQ.END) GO TO 40 | ADD | 52 |
| | CALL OUTCD(TEMP,OUT) | ADD | 53 |
| | NTC = NTC+1 | ADD | 54 |
| | IF (TEMP(4).EQ.USES) LASTM = NTC | ADD | 55 |
| | IF ((KARD(2).EQ.TEMP(2)).AND.(KARD(3).EQ.TEMP(3))) GO TO 130 | ADD | 56 |
| | IF ((KARD(5).EQ.TEMP(2)).AND.(KARD(6).EQ.TEMP(3))) NEED = NTC | ADD | 57 |
| | GO TO 30 | ADD | 58 |
| 40 | IF (TYPE.EQ.5HBASIC) CALL OUTCD(KARD,OUT) | ADD | 59 |
| | CALL OUTCD(TEMP,OUT) | ADD | 60 |
| | IF ((NEED.EQ.0).AND.(TYPE.EQ.3HMOD)) GO TO 140 | ADD | 61 |
| C | | ADD | 62 |
| C | COPY ALL MOD DECKS FROM IN1 TO OUT | ADD | 63 |
| C | | ADD | 64 |
| | FLAG = 3 | ADD | 65 |

| | | |
|---|-----|-----|
| NDECK = 0 | ADD | 66 |
| 50 CALL INCD(TEMP,IN1) | ADD | 67 |
| IF (ERFLAG.NE.0) GO TO 120 | ADD | 68 |
| KOUNT1 = KOUNT1 + 1 | ADD | 69 |
| CALL OUTCD(TEMP,OUT) | ADD | 70 |
| IF (TEMP(1).NE.END) GO TO 50 | ADD | 71 |
| NDECK = NDECK+1 | ADD | 72 |
| IF (NDECK.LT.LASTM) GO TO 50 | ADD | 73 |
| C | ADD | 74 |
| C TRANSFER NEW DECK ONTO OUT FILE | ADD | 75 |
| C | ADD | 76 |
| FLAG = 4 | ADD | 77 |
| CALL OUTCD(KARD,OUT) | ADD | 78 |
| 60 CALL INCD(KARD,IN2) | ADD | 79 |
| IF (ERFLAG.NE.0) GO TO 100 | ADD | 80 |
| KOUNT2 = KOUNT2+1 | ADD | 81 |
| CALL OUTCD(KARD,OUT) | ADD | 82 |
| IF (KARD(1).NE.END) GO TO 60 | ADD | 83 |
| C | ADD | 84 |
| C COPY REST OF DATA FILE, INCLUDING \$END OF FILE CARD, ONTO OUT FILE | ADD | 85 |
| C | ADD | 86 |
| FLAG = 5 | ADD | 87 |
| 80 CALL INCD(TEMP,IN1) | ADD | 88 |
| IF (ERFLAG.NE.0) GO TO 120 | ADD | 89 |
| KOUNT1 = KOUNT1+1 | ADD | 90 |
| CALL OUTCD(TEMP,OUT) | ADD | 91 |
| IF (TEMP(1).NE.END) GO TO 80 | ADD | 92 |
| IF (TEMP(2).NE.6HF FILE) GO TO 80 | ADD | 93 |
| RETURN | ADD | 94 |
| C | ADD | 95 |
| C ERROR STOPS | ADD | 96 |
| C | ADD | 97 |
| 100 WRITE (FERR,105) | ADD | 98 |
| 105 FORMAT(23H\$ERROR READING FILE IN2) | ADD | 99 |
| GO TO 200 | ADD | 100 |
| 110 WRITE (FERR,115) | ADD | 101 |

| | | | |
|--------|---|-------|-----|
| 115 | FORMAT(37H0FIRST CARD ON FILE IN2 IS NOT SDECK) | ADD | 102 |
| | GO TO 200 | ADD | 103 |
| 120 | WRITE (FERR,125) | ADD | 104 |
| 125 | FORMAT(23H0ERROR READING FILE IN1) | ADD | 105 |
| | GO TO 200 | ADD | 106 |
| 130 | WRITE (FERR,135) KARD(2), KARD(3) | ADD | 107 |
| 135 | FORMAT(45H0NAME CONFLICT IN ADDING NEW DECK TO FILE IN1/ * 6H NAME ,2A6,36H ALREADY EXISTS IN TABLE OF CONTENTS) | ADD | 108 |
| | GO TO 200 | ADD | 109 |
| 140 | WRITE (FERR,145) (KARD(I), I=1,6) | ADD | 110 |
| 145 | FORMAT(52H0DECK TO BE ADDED USES DECK NOT IN TABLE OF CONTENTS/ * 5X,6A6) | ADD | 111 |
| | GO TO 200 | ADD | 112 |
| 150 | WRITE (FERR,155) | ADD | 113 |
| 155 | FORMAT(17H0DECK USES ITSELF) | ADD | 114 |
| C | | ADD | 115 |
| 200 | ERFLAG = 1 | ADD | 116 |
| C | | ADD | 117 |
| C | ADVANCE FILE IN2 TO END OF CURRENT DECK | ADD | 118 |
| C | | ADD | 119 |
| | ERFLAG = 0 | ADD | 120 |
| 210 | IF (KARD(1).EQ.END) GO TO 220 | ADD | 121 |
| | CALL INCD(KARD,IN2) | ADD | 122 |
| | KOUNT2 = KOUNT2 + 1 | ADD | 123 |
| | IF (ERFLAG.EQ.0) GO TO 210 | ADD | 124 |
| | WRITE (FERR,165) | ADD | 125 |
| 220 | ERFLAG = 1 | ADD | 126 |
| | RETURN | ADD | 127 |
| | END | ADD | 128 |
| #ELT,I | DORMAN.ADDER | ADD | 129 |
| | SUBROUTINE ADDER | ADDER | 3 |
| C | | ADDER | 5 |
| C | ADD CONTRULER | ADDER | 6 |
| C | | ADDER | 7 |
| C | | ADDER | 8 |
| C | PROGRAMMER - S. WRAY | ADDER | 9 |
| C | | ADDER | 10 |

| | | |
|--|-------|----|
| COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC | 2 |
| INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC | 3 |
| EQUIVALENCE (PRTFIL,FERR) | MISC | 4 |
| DATA IX/0/ | ADDER | 12 |
| IF(IX.EQ.0) REWIND 21 | ADDER | 13 |
| IX = 1 | ADDER | 14 |
| 1 WRITE (PRTFIL,5) | ADDER | 15 |
| 5 FORMAT(12H ENTER INPUT/12H OR BASIC/15H OR MOD DECK/9H READY | ADDER | 16 |
| * -) | ADDER | 17 |
| CALL INGD(KARD,5) | ADDER | 18 |
| IF(KARD(1).EQ.5HBASIC) GO TO 10 | ADDER | 19 |
| IF(KARD(1).EQ.6HMOD DE) GO TO 15 | ADDER | 20 |
| IF(KARD(1).NE.5HINPUT) GO TO 1 | ADDER | 21 |
| CALL ADDX(21) | ADDER | 22 |
| RETURN | ADDER | 23 |
| 10 CONTINUE | ADDER | 24 |
| REWIND 14 | ADDER | 25 |
| CALL ADDX (14) | ADDER | 26 |
| RETURN | ADDER | 27 |
| 15 CONTINUE | ADDER | 28 |
| WRITE (PRTFIL,20) | ADDER | 29 |
| 20 FORMAT(*2H IS MOD DECK GENERATED OR INPUT ON TAPE 11/9H READY -) | ADDER | 30 |
| CALL INGD (KARD,5) | ADDER | 31 |
| M1 = 13 | ADDER | 32 |
| IF(KARD(1).EQ.5HINPUT) M1 = 11 | ADDER | 33 |
| REWIND M1 | ADDER | 34 |
| CALL ADDX(M1) | ADDER | 35 |
| RETURN | ADDER | 36 |
| END | ADDER | 37 |
| #ELT,I DORMAN.ADDX | ADDX | 3 |
| SUBROUTINE ADDX (INDKF) | ADDX | 5 |
| C | ADDX | 6 |
| C CONTROLS ADD OPERATION | ADDX | 7 |
| C | ADDX | 8 |
| C PROGRAMMER - S. WRAY | ADDX | 9 |
| C | ADDX | 10 |

| | | |
|--|-------|----|
| COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST | 2 |
| INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST | 3 |
| COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2 | FILES | 2 |
| INTEGER BASIC,MTAPE,FINAL,S1,S2 | FILES | 3 |
| COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC | 2 |
| INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC | 3 |
| EQUIVALENCE (PRTFIL,FERR) | MISC | 4 |
| INTEGER OUTDK,GU,DONE,IU(3),IR(3) | ADDX | 14 |
| DATA IR/2,2,2/ | ADDX | 15 |
| DATA GU/2HGO/,DONE/4HDONE/ | ADDX | 16 |
| C | ADDX | 17 |
| C INPUT DECKS ARE ON TAPE21, TEMP OUT FILE IS TAPE22 | ADDX | 18 |
| C | ADDX | 19 |
| IC = 0 | ADDX | 20 |
| IX = 1 | ADDX | 21 |
| OUTDK = 22 | ADDX | 22 |
| 10 CONTINUE | ADDX | 23 |
| NO = 0 | ADDX | 24 |
| IU(1) = 5 | ADDX | 25 |
| IU(2) = INDKF | ADDX | 26 |
| IU(3) = OUTDK | ADDX | 27 |
| CALL INTBUF(IU,IR) | ADDX | 28 |
| CALL INCD (KARD,INDKF) | ADDX | 29 |
| IF (ERFLAG.NE.0) GO TO 1000 | ADDX | 30 |
| 12 REWIND OUTDK | ADDX | 31 |
| IF (KARD(1).EQ.DDECK) GO TO 20 | ADDX | 32 |
| WRITE (PRTFIL,11) | ADDX | 33 |
| 11 FORMAT(27H PLEASE ENTER NAME FOR DECK / 9H READY -) | ADDX | 34 |
| CALL INCD(ACTION,5) | ADDX | 35 |
| ACTION (3) = ACTION(2) | ADDX | 36 |
| ACTION (2) = ACTION(1) | ADDX | 37 |
| ACTION (1) = DDECK | ADDX | 38 |
| CALL OUTCD (ACTION,OUTDK) | ADDX | 39 |
| 20 CONTINUE | ADDX | 40 |
| IC = IC+1 | ADDX | 41 |
| NO = 1 | ADDX | 42 |

| | | | |
|------|--|------|----|
| 39 | WRITE (PRTFIL,40) KARD(2),KARD(3) | ADJX | 43 |
| 40 | FORMAT(14H DECK FOUND - ,2AS/17H ENTER OK OR SKIP/9H READY -) | ADJX | 44 |
| | CALL INCD (ACTION,5) | ADJX | 45 |
| | IF(ACTION(1).EQ.2HOK) GO TO 45 | ADJX | 46 |
| | IF(ACTION(1).NE.4HSKIP) GO TO 39 | ADJX | 47 |
| 41 | CALL INCD (KARD,INDKF) | ADJX | 48 |
| | IF(ERFLAG.NE.0) RETURN | ADJX | 49 |
| | IF(KARD(1).NE.DDECK) GO TO 41 | ADJX | 50 |
| | GO TO 20 | ADJX | 51 |
| 45 | CONTINUE | ADJX | 52 |
| | CALL OUTCD(KARD,OUTOK) | ADJX | 53 |
| | IC = IC + 1 | ADJX | 54 |
| 21 | CALL INCD (KARD,INDKF) | ADJX | 55 |
| | ACTION(2)=KARD(2) | ADJX | 56 |
| | IF(ERFLAG.NE.0) RETURN | ADJX | 57 |
| | IF(KARD(1).NE.END) GO TO 45 | ADJX | 58 |
| 23 | CONTINUE | ADJX | 59 |
| | DO 22 I = 1,14 | ADJX | 60 |
| 22 | KARD(I) = BLANK | ADJX | 61 |
| | KARD(1) = END | ADJX | 62 |
| | KARD(2) = DECK | ADJX | 63 |
| | CALL OUTCD(KARD,OUTOK) | ADJX | 64 |
| | KARD(2) = FILE | ADJX | 65 |
| | IX = 2 | ADJX | 66 |
| | IF(ACTION(2).EQ.DECK) GO TO 30 | ADJX | 67 |
| 1000 | WRITE (PRTFIL,1001) | ADJX | 68 |
| 1001 | FORMAT(51H EOF FOUND - IF ACCEPTED, ENTER GO, OTHERWISE DONE /9H RAJDX | ADJX | 69 |
| | *EADY -) | ADJX | 70 |
| | CALL INCD (ACTION,5) | ADJX | 71 |
| | IF(ACTION(1).EQ.DONE) RETURN | ADJX | 72 |
| | IF(ACTION(1).NE.GO) GO TO 1000 | ADJX | 73 |
| | ERFLAG = 0 | ADJX | 74 |
| 30 | CONTINUE | ADJX | 75 |
| | WRITE (PRTFIL,31) IC | ADJX | 76 |
| 31 | FORMAT(I6,20H CARUS FOUND IN DECK) | ADJX | 77 |
| | IC = 0 | ADJX | 78 |

| | | |
|--|-------|----|
| S1 = BASIC + 1 | ADDX | 79 |
| IF(S1.GT.3) S1 = 2 | ADDX | 80 |
| REWIND BASIC | ADDX | 81 |
| REWIND OUTOK | ADDX | 82 |
| REWIND S1 | ADDX | 83 |
| IU(1) = BASIC | ADDX | 84 |
| IU(2) = OUTOK | ADDX | 85 |
| IU(3) = S1 | ADDX | 86 |
| CALL INTBUF(IU,IR) | ADDX | 87 |
| CALL ADD(BASIC,OUTOK,S1) | ADDX | 88 |
| IF(ERFLAG.NE.0) RETURN | ADDX | 89 |
| BASIC = S1 | ADDX | 90 |
| RETURN | ADDX | 91 |
| END | ADDX | 92 |
| *ELT,I DORMAN,ASSGN | ASSGN | 4 |
| SUBROUTINE ASSGN (I) | ASSGN | 6 |
| C | ASSGN | 7 |
| C ISSUE ASG CARD FOR FILE I | ASSGN | 8 |
| C | ASSGN | 9 |
| C PROGRAMMER - S. WRAY | ASSGN | 10 |
| C | ASSGN | 11 |
| COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC | 2 |
| INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC | 3 |
| EQUIVALENCE (PRTFIL,FERR) | MISC | 4 |
| WRITE (PRTFIL,5) 1 | ASSGN | 13 |
| 5 FORMAT(28H WHAT IS YOUR NAME FOR TAPE ,13/9H READY -) | ASSGN | 14 |
| CALL INGD (ACTION,5) | ASSGN | 15 |
| IF(ERFLAG.NE.0) RETURN | ASSGN | 16 |
| C | ASSGN | 17 |
| C ISSUE ASG AND USE CARDS | ASSGN | 18 |
| C | ASSGN | 19 |
| RETURN | ASSGN | 20 |
| END | ASSGN | 21 |
| *ELT,I DORMAN,CKSYN | CKSYN | 3 |
| SUBROUTINE CKSYN (ICD1,ICD2,N1,N2) | CKSYN | 5 |
| C | CKSYN | 6 |

| | | |
|--------|--|----------|
| C | ICD1 IS CARD IN BUFFER1 | CKSYN 7 |
| C | ICD2 IS CARD IN BUFFER2 | CKSYN 8 |
| C | N1 IS INDEX TO PROPER SYNC CARD | CKSYN 9 |
| C | N2 = 0, DO NOT INCREMENT N1 | CKSYN 10 |
| C | N2 = 1, OK TO INCREMENT N1 | CKSYN 11 |
| | COMMON /BFRS/ | CKSYN 12 |
| | * ISYN(2,20) | CKSYN 13 |
| C | | CKSYN 14 |
| C | IF ICD1 OR ICD2 IS GE TO SYNC CARD THEN | CKSYN 15 |
| C | RESET TO SYNC CARD VALUES | CKSYN 16 |
| C | | CKSYN 17 |
| C | PROGRAMMER: VOIT | CKSYN 18 |
| C | | CKSYN 19 |
| | ISYN1 = N1 | CKSYN 20 |
| | ICD1 = ISYN(1,ISYN1) | CKSYN 21 |
| | ICD2 = ISYN(2,ISYN1) | CKSYN 22 |
| | IF (N2.EQ.0) RETURN | CKSYN 23 |
| | N1 = N1+1 | CKSYN 24 |
| | RETURN | CKSYN 25 |
| | END | CKSYN 26 |
| #ELT,I | DORMAN.CLOSE | CLOSE 4 |
| | SUBROUTINE CLOSE (I) | CLOSE 5 |
| | ENDFILE I | CLOSE 7 |
| | REWIND I | CLOSE 8 |
| | RETURN | CLOSE 9 |
| | END | CLOSE 10 |
| #ELT,I | DORMAN.COUNT | COUNT 3 |
| | SUBROUTINE COUNT(NFILE) | COUNT 5 |
| | COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST 2 |
| | INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST 3 |
| | COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC 2 |
| | INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC 3 |
| | EQUIVALENCE (PRTFIL,FERR) | MISC 4 |
| C | | COUNT 8 |
| C | PROGRAMMER - S. WRAY | COUNT 9 |
| C | | COUNT 10 |

| | |
|---|----------|
| REWIND NFILE | COUNT 11 |
| ACTION (1) = 6HNO NAM | COUNT 12 |
| ACTION (2) = 6HE | COUNT 13 |
| DO 1 I=3,6 | COUNT 14 |
| 1 ACTION(I) = BLANK | COUNT 15 |
| I=1 | COUNT 16 |
| CALL INCD(KARD,NFILE) | COUNT 17 |
| IF(ERFLAG.NE.0) RETURN | COUNT 18 |
| IF(KARD(1).NE.DDECK) GO TO 5 | COUNT 19 |
| ACTION (1) = KARD (1) | COUNT 20 |
| ACTION (2) = KARD (2) | COUNT 21 |
| ACTION (3) = KARD (3) | COUNT 22 |
| ACTION (4) = KARD (4) | COUNT 23 |
| ACTION (5) = KARD (5) | COUNT 24 |
| ACTION (6) = KARD (6) | COUNT 25 |
| 5 CALL INCD (KARD,NFILE) | COUNT 26 |
| IF(ERFLAG.NE.0) GO TO 10 | COUNT 27 |
| I=I+1 | COUNT 28 |
| IF(KARD(1).NE.END) GO TO 5 | COUNT 29 |
| IF(KARD(2).NE.FILE) GO TO 5 | COUNT 30 |
| 10 WRITE (PRTFIL,15) NFILE,(ACTION(LL),LL=1,6),I | COUNT 31 |
| 15 FORMAT(5HTAPE,I3,6H WITH ,6A6,9H CONTAINS,I7,6H CARDS/) | COUNT 32 |
| RETURN | COUNT 33 |
| END | COUNT 34 |
| *ELT,I JORMAN.DELCD | DELCD 3 |
| SUBROUTINE DELCD (ISP) | DELCD 5 |
| C | DELCD 6 |
| C | DELCD 7 |
| C | DELCD 8 |
| C | DELCD 9 |
| C | DELCD 10 |
| C | DELCD 11 |
| COMMON /BFRS/ ISYN(2,20),ISYN1,IT1,IT2,IT3,FULL,CN1,CN2,LIN1,LIN2 | BFRS1 2 |
| * ,NWBUF,NB1,NB2,ICN1,ICN2 | BFRS1 3 |
| COMMON /BFRS/ BUF1(14,50),BUF2(14,50),AAA(700) | BFRS1 4 |
| INTEGER CN1,CN2,BUF1,BUF2 | BFRS1 5 |

```

LOGICAL FULL
INTEGER FIN1,FIN2
EQUIVALENCE (FOUT,IT3)
EQUIVALENCE (FIN1,IT1)
EQUIVALENCE (FIN2,IT2)
INTEGER FOUT
KK = ISP-2
KKK = LIN1 - 1
WRITE(FOUT,81) KKK,KK,

```

```

C
C
C
C

```

```

      MOVE BUFFER 1 UP
      IF SYNC IS OUT OF BUFFER

NCARDS = KK - KKK + 1
CN1 = CN1 + NCARDS
LIN1 = LIN1 + NCARDS
IF(CN1.LT.NB1) RETURN
ISP = LIN1
LIN1 = LIN1 - NCARDS
CALL RESTOR(BUF1,ISP,NB1,FIN1,LIN1)
CN1 = 1
LIN1 = ISP
RETURN

```

```

81  FORMAT (7H$DELETE,I13,I10)
END

```

```

*ELT,I DORMAN.DELET
SUBROUTINE DELET

```

```

C
C
C
C
C

```

```

      DELETES DECKS

      PROGRAMMER - S. WRAY

DIMENSION IU(3),IR(3)
COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14)
INTEGER ERFLAG,FERR,ACTION,PRTFIL
EQUIVALENCE (PRTFIL,FERR)

```

```

5FRS1  0
DELCD  13
DELCD  14
DELCD  15
DELCD  16
DELCD  17
DELCD  18
DELCD  19
DELCD  20
DELCD  21
DELCD  22
DELCD  23
DELCD  24
DELCD  25
DELCD  26
DELCD  27
DELCD  28
DELCD  29
DELCD  30
GHT13  1
DELCD  32
DELCD  33
DELCD  34
DELCD  35
DELCD  36
DELET  3
DELET  5
DELET  6
DELET  7
DELET  8
DELET  9
DELET  10
DELET  11
MISC   2
MISC   3
MISC   4

```

```

COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2
INTEGER BASIC,MTAPE,FINAL,S1,S2
DATA IR/2,2,2/
WRITE (PRTFIL,5)
5 FORMAT(27H WHAT DECK IS TO BE DELETED/9H READY - )
S1 = BASIC +1
IF(S1.GT.3) S1=2
IU(1) = 5
IU(2) = BASIC
IU(3) = S1
CALL INTBUF(IU,IR)
CALL INCU (ACTION,5)
CALL DELETE (ACTION,BASIC,S1)
IF(ERFLAG.NE.0) RETURN
BASIC=S1
RETURN
END
#ELT,I DORMAN.DELETE
SUBROUTINE DELETE(NAME,IN1,OUT)
C
C COPY DATA FROM FILE IN1 TO FILE OUT WITH NAMED DECK DELETED.
C PROGRAMMER - B. GOLD
C
COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14)
INTEGER ERFLAG,FERR,ACTION,PRTFIL
EQUIVALENCE (PRTFIL,FERR)
INTEGER OUT
INTEGER FLAG, NAME(2), TEMP(14)
EQUIVALENCE (KARD(1),TEMP(1))
COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH
INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH
C
C INITIALIZE FILES AND COPY VERSION CARD.
C
FLAG = 1
REWIND IN1

```

```

FILES 2
FILES 3
DELET 14
DELET 15
DELET 16
DELET 17
DELET 18
DELET 19
DELET 20
DELET 21
DELET 22
DELET 23
DELET 24
DELET 25
DELET 26
DELET 27
DELET 28
DELETE 3
DELETE 5
DELETE 6
DELETE 7
DELETE 8
DELETE 9
MISC 2
MISC 3
MISC 4
DELETE11
DELETE12
DELETE13
REST 2
REST 3
DELETE15
DELETE16
DELETE17
DELETE18
DELETE19

```

| | |
|--|----------|
| REWIND OUT | DELETE20 |
| ERFLAG = 0 | DELETE21 |
| WRITE (PRTFIL,10) NAME | DELETE22 |
| 10 FORMAT(19H0DELETE DECK NAMED ,2A6,15H FROM DATA FILE) | DELETE23 |
| CALL INCD(TEMP,IN1) | DELETE24 |
| IF (ERFLAG.NE.0) GO TO 120 | DELETE25 |
| KOUNT1 = 1 | DELETE26 |
| CALL OUTCD(TEMP,OUT) | DELETE27 |
| C | DELETE28 |
| C COPY TABLE OF CONTENTS. VERIFY THAT DECK TO BE DELETED IS IN TABLE | DELETE29 |
| C OF CONTENTS AND NOT REQUIRED BY ANY OTHER DECK. | DELETE30 |
| C | DELETE31 |
| FLAG = 2 | DELETE32 |
| INDEX = 0 | DELETE33 |
| NTC = 0 | DELETE34 |
| 30 CALL INCD(TEMP,IN1) | DELETE35 |
| IF (ERFLAG.NE.0) GO TO 120 | DELETE36 |
| KOUNT1 = KOUNT1+1 | DELETE37 |
| IF (TEMP(1).EQ.END) GO TO 40 | DELETE38 |
| NTC = NTC+1 | DELETE39 |
| IF ((TEMP(2).EQ.NAME(1)).AND.(TEMP(3).EQ.NAME(2))) GO TO 35 | DELETE40 |
| IF ((TEMP(5).EQ.NAME(1)).AND.(TEMP(6).EQ.NAME(2))) GO TO 140 | DELETE41 |
| CALL OUTCD(TEMP,OUT) | DELETE42 |
| GO TO 30 | DELETE43 |
| 35 INDEX = NTC | DELETE44 |
| GO TO 30 | DELETE45 |
| 40 CALL OUTCD(TEMP,OUT) | DELETE46 |
| IF (INDEX.EQ.0) GO TO 130 | DELETE47 |
| C | DELETE48 |
| C COPY ALL DECKS FROM IN1 TO OUT UNTIL REQUIRED DECK IS FOUND. | DELETE49 |
| C | DELETE50 |
| FLAG = 3 | DELETE51 |
| 50 CALL INCD(TEMP,IN1) | DELETE52 |
| IF (ERFLAG.NE.0) GO TO 120 | DELETE53 |
| KOUNT1 = KOUNT1 + 1 | DELETE54 |
| IF ((TEMP(1).EQ.DUECK).AND.(TEMP(2).EQ.NAME(1))) | DELETE55 |

| | | | | |
|---|-----|--|----------|----------|
| | * | .AND.(TEMP(3).EQ.NAME(2)) | GO TO 55 | DELETE50 |
| | | CALL OUTCD(TEMP,OUT) | | DELETE57 |
| | | GO TO 50 | | DELETE58 |
| C | | | | DELETE59 |
| C | | SPACE PAST OLD DECK | | DELETE60 |
| C | | | | DELETE61 |
| | 55 | FLAG = + | | DELETE62 |
| | | KOUNT2 = KOUNT1 | | DELETE63 |
| | 60 | CALL INCD(TEMP,IN1) | | DELETE64 |
| | | IF (ERFLAG.NE.0) GO TO 120 | | DELETE65 |
| | | KOUNT1 = KOUNT1 + 1 | | DELETE66 |
| | | IF (TEMP(1).NE.END) GO TO 60 | | DELETE67 |
| | | KOUNT2 = KOUNT1-KOUNT2+1 | | DELETE68 |
| | | WRITE (PRTFIL,70) KOUNT2 | | DELETE69 |
| | 70 | FORMAT(/I10,14H CARDS DELETED) | | DELETE70 |
| C | | | | DELETE71 |
| C | | COPY REST OF DATA FILE, INCLUDING SEND OF FILE CARD, ONTO OUT FILE | | DELETE72 |
| C | | | | DELETE73 |
| | | FLAG = 5 | | DELETE74 |
| | 80 | CALL INCD(TEMP,IN1) | | DELETE75 |
| | | IF (ERFLAG.NE.0) GO TO 120 | | DELETE76 |
| | | KOUNT1 = KOUNT1+1 | | DELETE77 |
| | | CALL OUTCD(TEMP,OUT) | | DELETE78 |
| | | IF (TEMP(1).NE.END) GO TO 80 | | DELETE79 |
| | | IF (TEMP(2).NE.6HF FILE) GO TO 80 | | DELETE80 |
| | | RETURN | | DELETE81 |
| C | | | | DELETE82 |
| C | | ERROR STOPS | | DELETE83 |
| C | | | | DELETE84 |
| | 120 | WRITE (FERR,125) | | DELETE85 |
| | 125 | FORMAT(23HBERROR READING FILE IN1) | | DELETE86 |
| | | GO TO 200 | | DELETE87 |
| | 130 | WRITE (FERR,135) NAME | | DELETE88 |
| | 135 | FORMAT(22H0DECK TO BE DELETED (,2A6,36H) IS NOT LISTED IN TABLE | | DELETE89 |
| | | *F CONTENTS) | | DELETE90 |
| | | GO TO 200 | | DELETE91 |

| | |
|--|----------|
| 140 WRITE (FERR,145) NAME, TEMP(2), TEMP(3) | DELETE92 |
| 145 FORMAT(21HODECK TO BE DELETED (2A6,22H) IS REQUIRED BY DECK 2A6) | DELETE93 |
| C | DELETE94 |
| 200 ERFLAG = 1 | DELETE95 |
| RETURN | DELETE96 |
| END | DELETE97 |
| #ELT,I DORMAN.DIFDEC | DIFDEC03 |
| SUBROUTINE DIFDEC (FIN1,FIN2,FOUT,FSYN) | DIFDEC05 |
| C | DIFDEC06 |
| FSYN WILL CONTAIN THE SYNC CARDS | DIFDEC07 |
| THE SYNC SEARCH WILL BE ABANDONED IF THE CARD | DIFDEC08 |
| NUMBER IS GREATER THAN THE SYNC CARD | DIFDEC09 |
| THEN THE SYNC CARD NUMBERS WILL BE USED | DIFDEC10 |
| C | DIFDEC11 |
| COMMON /BUFFER/MAX | DIFDEC12 |
| COMMON /BFRS/ ISYN(2,20),ISYN1,IT1,IT2,IT3,FULL,CN1,CN2,LIN1,LIN2 | BFRS1 2 |
| * ,NWBUF,NB1,NB2,ICN1,ICN2 | BFRS1 3 |
| COMMON /BFRS/ BUF1(14,50),BUF2(14,50),AAA(700) | BFRS1 4 |
| INTEGER CN1,CN2,BUF1,BUF2 | BFRS1 5 |
| LOGICAL FULL | BFRS1 6 |
| INTEGER FIN1,FIN2,FOUT | DIFDEC14 |
| INTEGER FSYN | DIFDEC15 |
| C | DIFDEC16 |
| THIS ROUTINE DETERMINES THE DIFFERENCE BETWEEN DECKS ON TAPES FIN1 | DIFDEC17 |
| AND FIN2. THE DIFFERENCE IS EXPRESSED AS A MOD DECK TO CONVERT | DIFDEC18 |
| THE DECK ON FIN1 TO THE DECK ON FIN2. THE MOD DECK IS WRITTEN | DIFDEC19 |
| ON FOUT. | DIFDEC20 |
| C | DIFDEC21 |
| DIMENSION IUNT(3),IRW(3) | DIFDEC22 |
| INTEGER COUT(14) | DIFDEC23 |
| NWBUF = 50 | DIFDEC24 |
| MAX= NWBUF | DIFDEC25 |
| IT1 = FIN1 | DIFDEC26 |
| | DIFDEC27 |
| | DIFDEC28 |
| | DIFDEC29 |

```

IT2 = FIN2
IT3 = FOUT

C
C      READ SYNC CARDS
C
IT4 = FSYN
CALL SYNCDS(IT4)

C
C      INITIALIZE UNITS
C

ISYN1 = 1
IUNT(1)=FIN1
IUNT(2)=FIN2
IUNT(3)=FOUT
IRW(1) = 0
IRW(2) = 0
IRW(3) = 1
CALL INTBUF (IUNT,IRW)
CALL FILBUF(1,NB1,NWBUF,BUF1,FIN1)
CALL FILBUF(1,NB2,NWBUF,BUF2,FIN2)
FULL = .TRUE.
CN1 = 1
CN2 = 1
LIN1 = 1
LIN2 = 1
DO 5 I=1,14
5 COUT(I)=1H
COUT(1) = BUF1(1,1)
COUT(2) = BUF2(2,1)
COUT(3) = BUF2(3,1)
COUT(4) = 6H USES
COUT(5) = BUF1(2,1)
COUT(6) = BUF1(3,1)
CALL LXTRAN (COUT(7),COUT(9))
CALL OUTCD (COUT,6)
CALL OUTCD (COUT,FOUT)

```

```

DIFDEC30
DIFDEC31
DIFDEC32
DIFDEC33
DIFDEC34
DIFDEC35
DIFDEC36
DIFDEC37
DIFDEC38
DIFDEC39
DIFDEC40
DIFDEC41
DIFDEC42
DIFDEC43
DIFDEC44
DIFDEC45
DIFDEC46
DIFDEC47
DIFDEC48
DIFDEC49
DIFDEC50
DIFDEC51
DIFDEC52
DIFDEC53
DIFDEC54
DIFDEC55
DIFDEC56
DIFDEC57
DIFDEC58
DIFDEC59
DIFDEC60
DIFDEC61
DIFDEC62
DIFDEC63
DIFDEC64
DIFDEC65

```

| | | |
|-----|--|----------|
| 10 | CN1 = CN1+1 | DIFDEC66 |
| | CN2 = CN2+1 | DIFDEC67 |
| | LIN2=LIN2+1 | DIFDEC68 |
| | LIN1=LIN1+1 | DIFDEC69 |
| 12 | CONTINUE | DIFDEC70 |
| | IF(CN1.GT.NB1) GO TO 100 | DIFDEC71 |
| | IF(CN2.GT.NB2) GO TO 150 | DIFDEC72 |
| | DO 15 I=1,14 | DIFDEC73 |
| | IF(BUF1(I,CN1).NE.BUF2(I,CN2)) GO TO 20 | DIFDEC74 |
| 15 | CONTINUE | DIFDEC75 |
| | FULL = .FALSE. | DIFDEC76 |
| | GO TO 10 | DIFDEC77 |
| 20 | CONTINUE | DIFDEC78 |
| C | | DIFDEC79 |
| C | CHECK SYNC CARDS | DIFDEC80 |
| C | PRIOR TO SEARCH FOR MATCH | DIFDEC81 |
| C | | DIFDEC82 |
| | N2=0 | DIFDEC83 |
| | CALL CKSYN(ICN1,ICN2,ISYN1,N2) | DIFDEC84 |
| 500 | CONTINUE | DIFDEC85 |
| | IF ((LIN1.LT.ICN1).AND.(LIN2.EQ.ICN2)) GO TO 600 | DIFDEC86 |
| | IF ((LIN1.EQ.ICN1).AND.(LIN2.LT.ICN2)) GO TO 620 | DIFDEC87 |
| | IF((LIN1.EQ.ICN1).AND.(LIN2.EQ.ICN2))GO TO 630 | DIFDEC88 |
| | IF (LIN1.GT.ICN1) GO TO 630 | DIFDEC89 |
| | IF (LIN2.GT.ICN2) GO TO 630 | DIFDEC90 |
| | GO TO 640 | DIFDEC91 |
| C | | DIFDEC92 |
| C | DELETE | DIFDEC93 |
| C | | DIFDEC94 |
| 600 | CALL DELCD (ICN1-1) | DIFDEC95 |
| | GO TO 10 | DIFDEC96 |
| C | | DIFDEC97 |
| C | INSERT | DIFDEC98 |
| C | | DIFDEC99 |
| 620 | CALL INSERT (ICN2-1,0) | DIFDE100 |
| | GO TO 10 | DIFDE101 |

```

C
C      GET NEXT SYNC CARDS
C
630  N2=1
      CALL CKSYN(ICN1,ICN2,ISYN1,N2)
      GO TO 500
C
C      FIND SYNC
C
640  CONTINUE
      CALL SYNC1
      GO TO 12
100  CONTINUE
      IF(BUF1(1,NB1).EQ.6H$END 0) GO TO 120
      CALL FILBUF(1,NB1,NWBUF,BUF1,FIN1)
      CN1 = 1
      GO TO 12
120  CONTINUE
      IF(CN2.GT.NB2) GO TO 200
      CALL INSERT(NB2-1,0)
      GO TO 200
150  IF(BUF2(1,NB2).EQ.6H$END 0) GO TO 170
      CALL FILBUF(1,NB2,NWBUF,BUF2,FIN2)
      CN2= 1
      GO TO 12
170  CONTINUE
      IF(CN1.GT.NB1) GO TO 200
      CALL DELCO (NB1-1)
200  CONTINUE
      WRITE(FOUT,5000)
5000 FORMAT(12H$END OF DECK/12H$END OF FILE)
      END FILE FOUT
      RETURN
      END
#ELT,I DORMAN,EDITOK
      SUBROUTINE EDITOK(IN,MOD,NOUT)

```

```

DIFDE102
DIFDE103
DIFDE104
DIFDE105
DIFDE106
DIFDE107
DIFDE108
DIFDE109
DIFDE110
DIFDE111
DIFDE112
DIFDE113
DIFDE114
DIFDE115
DIFDE116
DIFDE117
DIFDE118
DIFDE119
DIFDE120
DIFDE121
DIFDE122
DIFDE123
DIFDE124
DIFDE125
DIFDE126
DIFDE127
DIFDE128
DIFDE129
DIFDE130
DIFDE131
DIFDE132
DIFDE133
DIFDE134
DIFDE135
EDITOK 3
EDITOK 5

```

```

COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14)
INTEGER ERFLAG,FERR,ACTION,PRTFIL
EQUIVALENCE (PRTFIL,FERR)
DIMENSION INB(14), MODB(14), NTEM1(14), NTEM2(16)
DIMENSION IEM(7,9)
DIMENSION NTEM3(16)
INTEGER OEND
DATA IEM /6HEDITOK,6H - ,
16HDECK N,6HAMES N,6HOT COM,6HPATIBL,6HE ,
26HEDITOK,6H - ,
26HNON-NU,6HMERIC ,6HENTRY ,6HIN LIN,6HE NO. ,
36HEDITOK,6H - ,
36HLINE N,6HO. IS ,6HOUT OF,6H RANGE,6H ,
46HEDITOK,6H - ,
46HLINE N,6HO. IS ,6HOUT OF,6H ORDER,6H ,
56HEDITOK,6H - ,
56HMODS R,6HEMAIN ,6H- END ,6HOF IN ,6HLECK ,
66HEDITOK,6H - ,
66HCONTIN,6HUATION,6H CARD ,6HOUT OF,6H ORDER,
76HEDITOK,6H - ,
76HDELET,6HE LINE,6H NOS. ,6HOUT OF,6H ORDER,
86HEDITOK,6H - ,
86HNO DAT,6HAMATC,6HH ON 3,6HALTER ,6HCARD ,
96HEDITOK,6H - ,
96HCONTIN,6HUATION,6H CARD ,6HNOT EN,6HTERED /

```

```

MISC 2
MISC 3
MISC 4
EDITOK 7
EDITOK 8
EDITOK 9
EDITOK10
EDITOK11
EDITOK12
EDITOK13
EDITOK14
EDITOK15
EDITOK16
EDITOK17
EDITOK18
EDITOK19
EDITOK20
EDITOK21
EDITOK22
EDITOK23
EDITOK24
EDITOK25
EDITOK26
EDITOK27
EDITOK28
EDITOK29
EDITOK30
EDITOK31
EDITOK32
EDITOK33
EDITOK34
EDITOK35
EDITOK36
EDITOK37
EDITOK38
EDITOK39

```

```

C
C
C
C
C

```

```

PROGRAMMER - G. W. TIMPSON

PRESET VARIABLES

ERFLAG=0
I = - 1
MEND=0
OEND=0
IE=0
NEF=0

```

```

C
C   READ H CARD FROM MOD FILE
C
10  NTL=NT
    NT=1
    IF(MEND.EQ.1)GO TO 12
    CALL INCD(MODB,MOD)
    IF(ERFLAG.NE.0)RETURN
    IF(MODB(1).EQ.6H$END 0)GO TO 13
    IF(MODB(1).EQ.6H$INSER)GO TO 14
    IF(MODB(1).EQ.6H$ALTER)GO TO 15
    IF(MODB(1).EQ.6H$DELET) GO TO 16
12  GO TO 20
13  MEND=1
    GO TO 20
14  NT=4
    GO TO 20
15  NT=3
    GO TO 20
16  NT=2
C
C   TEST FOR FIRST TIME
C
20  IF(I.NE.-1) GO TO 22
    NSW=1
    GO TO 600
C
C   TEST FOR PREVIOUS MOD OUT OF RANGE
C
22  IF(IE.EQ.5)GO TO 700
C
C   TEST FOR END OF IN FILE
C
24  IF(CEND.EQ.0)GO TO 30
C
C   TEST FOR CONTINUATION INSERT CARD

```

```

EDITOK40
EDITOK41
EDITOK42
EDITOK43
EDITOK44
EDITOK45
EDITOK46
EDITOK47
EDITOK48
EDITOK49
EDITOK50
EDITOK51
EDITOK52
EDITOK53
EDITOK54
EDITOK55
EDITOK56
EDITOK57
EDITOK58
EDITOK59
EDITOK60
EDITOK61
EDITOK62
EDITOK63
EDITOK64
EDITOK65
EDITOK66
EDITOK67
EDITOK68
EDITOK69
EDITOK70
EDITOK71
EDITOK72
EDITOK73
EDITOK74
EDITOK75

```

| | | |
|----|--|----------|
| C | | EDIT0K76 |
| | IF(IE.NE.0)GO TO 52 | EDIT0K77 |
| | IF(NT.EQ.1)GO TO 800 | EDIT0K78 |
| | IE=5 | EDIT0K79 |
| | GO TO 700 | EDIT0K80 |
| C | | EDIT0K81 |
| C | TEST FOR END OF MOD FILE | EDIT0K82 |
| C | | EDIT0K83 |
| 30 | IF(MEND.EQ.0)GO TO 40 | EDIT0K84 |
| | NSW=5 | EDIT0K85 |
| | GO TO 500 | EDIT0K86 |
| C | | EDIT0K87 |
| C | TEST FOR CONTINUATION INSERT CARD | EDIT0K88 |
| C | | EDIT0K89 |
| 40 | IF(NT.NE.1)GO TO 60 | EDIT0K90 |
| C | | EDIT0K91 |
| C | TEST PREVIOUS MOD CARD AND ERROR CONDITION | EDIT0K92 |
| C | | EDIT0K93 |
| | IF(IE.NE.0)GO TO 52 | EDIT0K94 |
| | IF(NTL.NE.3)GO TO 800 | EDIT0K95 |
| 50 | NSW=2 | EDIT0K96 |
| | GO TO 500 | EDIT0K97 |
| C | | EDIT0K98 |
| C | PYPASS CONTINUATION CARDS | EDIT0K99 |
| C | | EDIT0100 |
| 52 | IE=9 | EDIT0101 |
| | GO TO 700 | EDIT0102 |
| C | | EDIT0103 |
| C | GET LINE/CARD NUMBERS | EDIT0104 |
| C | | EDIT0105 |
| 60 | IE=0 | EDIT0106 |
| | ENCODE(84,62,NTEM1)MOUB | EDIT0107 |
| 62 | FORMAT(14A6) | EDIT0108 |
| | DECODE(80,64,NTEM1)NTEM2 | EDIT0109 |
| 64 | FORMAT(8(A6,A4)) | EDIT0110 |
| | CALL VALUE(NTEM2(3),VN1,IER) | EDIT0111 |

```

        IF(IER.EQ.0) GO TO 68
C
C      SET ERROR FLAG - ILLEGAL LINE ENTRY
C
      66  IE=2
        GO TO 700
      68  N1=VN1
        NSW=3
C
C      TEST FOR FILE POSITION
C
      70  IF(I.LT.N1)GO TO 500
        IF(I.EQ.N1)GO TO 80
C
C      SET ERROR FLAG - MOD OUT OF ORDER
C
        IE=4
        GO TO 700
C
C      TEST TYPE OF CARD
C
      80  GO TO (82,84,110,81),NT
C
C      CONTINUATION INSERT CARD IS ILLEGAL
C
      82  IE=6
        GO TO 700
      81  NSW=6
        GO TO 500
C
C      TEST VALIDITY OF N2
C
      84  CALL VALUE(NTEM2(5),VN2,IER)
        IF (IER.EQ.-1)GO TO 86
        IF (IER.GT.0)GO TO 66
        N2=VN2

```

```

EDITD112
EDITD113
EDITD114
EDITD115
EDITD116
EDITD117
EDITD118
EDITD119
EDITD120
EDITD121
EDITD122
EDITD123
EDITD124
EDITD125
EDITD126
EDITD127
EDITD128
EDITD129
EDITD130
EDITD131
EDITD132
EDITD133
EDITD134
EDITD135
EDITD136
EDITD137
EDITD138
EDITD139
EDITD140
EDITD141
EDITD142
EDITD143
EDITD144
EDITD145
EDITD146
EDITD147

```

| | | |
|-----|--------------------------------------|----------|
| | IF(N2.LT.N1)GO TO 90 | EDITD140 |
| 85 | NSW=4 | EDITD149 |
| | GO TO 600 | EDITD150 |
| 86 | N2=N1 | EDITD151 |
| | GO TO 85 | EDITD152 |
| C | | EDITD153 |
| C | DELETE LINES OUT OF SEQUENCE | EDITD154 |
| C | | EDITD155 |
| 90 | IE=7 | EDITD156 |
| | GO TO 700 | EDITD157 |
| C | | EDITD158 |
| C | TEST FOR END OF DELETE STRING | EDITD159 |
| C | | EDITD160 |
| 100 | IF (I.LE.N2)GO TO 85 | EDITD161 |
| | GO TO 810 | EDITD162 |
| C | | EDITD163 |
| C | ALTER OPTION | EDITD164 |
| C | | EDITD165 |
| 110 | ENCODE(84,62,NTEM1) INB | EDITD166 |
| | DECODE(80,64,NTEM1) NTEM3 | EDITD167 |
| | DO112K=1,16,2 | EDITD168 |
| | IF(NTEM3(K).NE.NTEM2(5)) GO TO 112 | EDITD169 |
| | IF(NTEM3(K+1).NE.NTEM2(6)) GO TO 112 | EDITD170 |
| | GO TO 114 | EDITD171 |
| 112 | CONTINUE | EDITD172 |
| | IE=0 | EDITD173 |
| | GO TO 700 | EDITD174 |
| 114 | NTEM3(K)=NTEM2(7) | EDITD175 |
| | NTEM3(K+1)=NTEM2(8) | EDITD176 |
| | ENCODE(80,64,NTEM1) NTEM3 | EDITD177 |
| | DECODE(84,62,NTEM1) INB | EDITD178 |
| | GO TO 810 | EDITD179 |
| C | | EDITD180 |
| C | TEST FOR RANGE ERROR | EDITD181 |
| C | | EDITD182 |
| 120 | IF (DEND.EQ.0)GO TO 70 | EDITD183 |

```

        IE=3
        GO TO 700
C
C      TEST FOR NAME MATCH
C
130  IF(MODB(5).NE.INB(2))GO TO 140
      IF(MODB(6).NE.INB(3))GO TO 140
      MOJB(4)=1H
      MODB(5)=1H
      MOJB(6)=1H
      NSW=2
      GO TO 600
140  IE=1
      GO TO 700
C
C      WRITE IN RECORD ON NOUT
C
500  CALL OUTCD(INB,NOUT)
C
C      READ IN RECORD - TEST FOR $ENDOFDECK
C
600  I=I+1
      CALL INCD(INB,IN)
      IF(ERFLAG.NE.0)RETURN
      IF(INB(1).EQ.6H$END 0)OEND=1
      GO TO (130,800,120,100,24,10),NSW
C
C      PRINT ERROR MESSAGE
C
700  WRITE (FERR,702) (IEM(J,IE),J=1,7),MODB
702  FORMAT(1H ,7A6,3H - /1X,14A6)
      NEF=1
      GO TO (820,810,810,810,810,820,810,810,810),IE
C
C      TEST FOR END OF EDITING
C

```

```

EDIT0184
EDIT0185
EDIT0186
EDIT0187
EDIT0188
EDIT0189
EDIT0190
EDIT0191
EDIT0192
EDIT0193
EDIT0194
EDIT0195
EDIT0196
EDIT0197
EDIT0198
EDIT0199
EDIT0200
EDIT0201
EDIT0202
EDIT0203
EDIT0204
EDIT0205
EDIT0206
EDIT0207
EDIT0208
EDIT0209
EDIT0210
EDIT0211
EDIT0212
EDIT0213
EDIT0214
EDIT0215
EDIT0216
EDIT0217
EDIT0218
EDIT0219

```

| | | |
|--------|--|----------|
| 800 | CALL OUTCO(MODB,NOUT) | EDIT0220 |
| 810 | IF(MEND.EQ.0)GO TO 10 | EDIT0221 |
| | IF(CEND.EQ.0)GO TO 10 | EDIT0222 |
| 820 | ERFLAG=NEF | EDIT0223 |
| | RETURN | EDIT0224 |
| | END | EDIT0225 |
| #ELT,I | DORMAN.EDITOR | EDITOR 3 |
| | SUBROUTINE EDITER | EDITOR 5 |
| C | | EDITOR 6 |
| C | BUILDS EDIT DECKS | EDITOR 7 |
| C | | EDITOR 8 |
| C | PROGRAMMER - S. WRAY | EDITOR 9 |
| C | | EDITOR10 |
| | COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC 2 |
| | INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC 3 |
| | EQUIVALENCE (PRTFIL,FERR) | MISC 4 |
| | COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2 | FILES 2 |
| | INTEGER BASIC,MTAPE,FINAL,S1,S2 | FILES 3 |
| | COMMON /NAMES/ IVER,ONAME(2),MODNAM(2),BNAME(2) | NAMES 2 |
| | INTEGER ONAME,MODNAM,BNAME | NAMES 3 |
| | COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST 2 |
| | INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST 3 |
| | INTEGER BMOD(5),CMOD(6) | EDITOR15 |
| | DIMENSION IU(3),IR(3) | EDITOR16 |
| | DATA IU,IR/0,0,0,2,2,2/ | EDITOR17 |
| | WRITE (PRTFIL,5) | EDITOR18 |
| 5 | FORMAT(21H BEGIN EDIT OPERATION/23H ENTER NAME OF NEW DECK/9H READ | EDITOR19 |
| | *Y -) | EDITOR20 |
| | CALL INCO(KARD,5) | EDITOR21 |
| | DO 10 I=1,14 | EDITOR22 |
| 10 | ACTION(I) = BLANK | EDITOR23 |
| | ACTION(1) = DDECK | EDITOR24 |
| | ACTION(2) = KARD(1) | EDITOR25 |
| | ACTION(3) = KARD(2) | EDITOR26 |
| | ACTION(4) = USES | EDITOR27 |
| | WRITE (PRTFIL,15) | EDITOR28 |

| | | |
|----|--|----------|
| 15 | FORMAT(29H ENTER NAME OF REFERENCE DECK/9H READY -) | EDITER29 |
| | CALL INCD(KARD,5) | EDITER30 |
| | FINAL = 14 | EDITER31 |
| | ACTION(5) = KARD(1) | EDITER32 |
| | ACTION(6) = KARD(2) | EDITER33 |
| | CALL EXTRAN (ACTION(6),ACTION(1)) | EDITER34 |
| | IF((KARD(1).EQ.BNAME(1)).AND.(KARD(2).EQ.BNAME(2))) GO TO 25 | EDITER35 |
| | IF(BNAME(1).EQ.0) GO TO 24 | EDITER36 |
| 19 | WRITE (PRTFIL,29) | EDITER37 |
| 20 | FORMAT(28H REFERENCE DECK IS NOT BASIC/29H DESTRUCT PERMISSION REQ | EDITER38 |
| | *UIRED/16H ENTER YES OR NO/9H READY -) | EDITER39 |
| | CALL INCD (KARD,5) | EDITER40 |
| | IF(KARD(1).EQ.2HNO) RETURN | EDITER41 |
| | IF(KARD(1).NE.3HYES) GO TO 19 | EDITER42 |
| 24 | CALL USE(ACTION(5)) | EDITER43 |
| 25 | REWIND FINAL | EDITER44 |
| | MODNAM(1) = ACTION(2) | EDITER45 |
| | MODNAM(2) = ACTION(3) | EDITER46 |
| | WRITE (PRTFIL,30) | EDITER47 |
| 30 | FORMAT(16H ENTER MOD CARDS/25H ENTER DONE WHEN FINISHED/9H READY - | EDITER48 |
| | *) | EDITER49 |
| | CALL INCD (KARD,5) | EDITER50 |
| | S1 = 22 | EDITER51 |
| | S2 = 22 | EDITER52 |
| | REWIND S1 | EDITER53 |
| | IU(1) = S1 | EDITER54 |
| | IU(2) = S1 + 1 | EDITER55 |
| | IU(3) = FINAL | EDITER56 |
| | CALL INTBUF(IU,1R) | EDITER57 |
| | CALL OUTCD (ACTION,S1) | EDITER58 |
| | DO 35 I = 1,14 | EDITER59 |
| 35 | ACTION(I) = BLANK | EDITER60 |
| | ACTION(1) = END | EDITER61 |
| | ACTION(2) = DECK | EDITER62 |
| | CALL OUTCD(ACTION,S1) | EDITER63 |
| 40 | CONTINUE | EDITER64 |

```

S1 = S2
REWIND S1
S2 = S1 + 1
IF(S2.GT.23) S2 = 22
REWIND S2
MODBP = 0
MODLP = 5000
MODBQ = 0
MODLQ = 5000
CALL INCD (ACTION,S1)
IF(ERFLAG.NE.0) RETURN
CALL OUTCD (ACTION,S2)
CALL INCD (ACTION,S1)
IF(ERFLAG.NE.0) RETURN
IJ = 1
GO TO 90
50 CONTINUE
IJ = 2
IF(KARD(1).EQ.6H00NE ) GO TO 500
IF(KARD(1).EQ.6HLIST ) GO TO 55
IF(KARD(1).EQ.6H$DELET ) GO TO 61
IF(KARD(1).EQ.6H$CHANG ) GO TO 62
IF(KARD(1).EQ.6H$INSER ) GO TO 63
IF(KARD(1).EQ.6H$ADJ ) GO TO 63
IF(KARD(1).EQ.6H$ALTER ) GO TO 62
48 CALL OUTCD(KARD,S2)
51 WRITE (PRTFIL,49)
CALL INCD(KARD,5)
GO TO 50
55 CONTINUE
IX = 1
54 DO 540 I=1,5
540 CMOD(I) = KARD(I)
545 ENCODE (30,56,BMOD) (CMOD(I),I=1,5)
56 FORMAT (5A6)
57 FORMAT (3(A6,A4))

```

```

EDITER65
EDITER66
EDITER67
EDITER68
EDITER69
EDITER70
EDITER71
EDITER72
EDITER73
EDITER74
EDITER75
EDITER76
EDITER77
EDITER78
EDITER79
EDITER80
EDITER81
EDITER82
EDITER83
EDITER84
EDITER85
EDITER86
EDITER87
EDITER88
EDITER89
EDITER90
EDITER91
EDITER92
EDITER93
EDITER94
EDITER95
EDITER96
EDITER97
EDITER98
EDITER99
EDITE100

```

```

        DECODE (3J,57,8MOD) CMOU
        CALL VALUE (CMOD(3),V,IER)
        IF(IER.GT.0) GO TO 58
        IST = V
        ISP = 1ST
        IF(CMOD(1).EQ.6H$ALTER ) GO TO 5300
        IF(CMOD(1).EQ.6H$INSER ) GO TO 5300
        CALL VALUE (CMOD(5),V,IER)
        IF(IER.GT.0) GO TO 58
        ISP = V
        IF(ISP.LT.IST) ISP = IST
5300 CONTINUE
        GO TO (53,75,105),IX
        53 CALL LGRO (FINAL,IST,ISP,PRTFIL)
        REWIND FINAL
        CALL INTBUF(IU,IR)
        GO TO 51
        59 FORMAT(36H ILLEGAL NUMERIC ENTRY - REENTER CARD)
        58 WRITE (FERR,59)
        GO TO (51,51,106),IX
        61 IY = 2
        GO TO 64
        62 IY = 1
        GO TO 64
        63 IY = 3
        64 IX = 2
        GO TO 54
        75 INB = IST
        INL = 1SP
        49 FORMAT(9H READY - )
        IF(INB.GE.MOUBP) GO TO 80
        IF(INL.GT.MOUBP) GO TO 79
        81 CALL OUTCD (ACTION,S2)
        IF(ACTION(1).EQ.END) GO TO 40
        CALL INCD(ACTION,S1)
        GO TO 81

```

```

EDITE101
EDITE102
EDITE103
EDITE104
EDITE105
EDITE106
EDITE107
EDITE108
EDITE109
EDITE110
EDITE111
EDITE112
EDITE113
EDITE114
EDITE115
EDITE116
EDITE117
EDITE118
EDITE119
EDITE120
EDITE121
EDITE122
EDITE123
EDITE124
EDITE125
EDITE126
EDITE127
EDITE128
EDITE129
EDITE130
EDITE131
EDITE132
EDITE133
EDITE134
EDITE135
EDITE136

```

| | | |
|-----|---|----------|
| 79 | CONTINUE | EDITE137 |
| | WRITE (FERR,82) | EDITE138 |
| 82 | FORMAT(40H MOD OVERLAPS EXISTING MOD - RENTER CARD) | EDITE139 |
| | GO TO 51 | EDITE140 |
| 80 | CONTINUE | EDITE141 |
| | IF(INB.GE.MODLP) GO TO 85 | EDITE142 |
| | IF(INL.GT.MODLP) GO TO 79 | EDITE143 |
| 83 | MODBP = INB | EDITE144 |
| | MODBQ = INL | EDITE145 |
| | GO TO 48 | EDITE146 |
| 90 | CONTINUE | EDITE147 |
| | IA = 0 | EDITE148 |
| | IF(ACTION(1).EQ.6H\$DELET) GO TO 91 | EDITE149 |
| | IF(ACTION(1).EQ.6H\$CHANG) GO TO 92 | EDITE150 |
| | IF(ACTION(1).EQ.6H\$INSER) GO TO 93 | EDITE151 |
| | IF(ACTION(1).EQ.6H\$ADD) GO TO 93 | EDITE152 |
| | IF(ACTION(1).EQ.6H\$ALTER) GO TO 92 | EDITE153 |
| | GO TO 106 | EDITE154 |
| 91 | IA = 2 | EDITE155 |
| | GO TO 94 | EDITE156 |
| 92 | IA = 1 | EDITE157 |
| | GO TO 94 | EDITE158 |
| 93 | IA = 3 | EDITE159 |
| 94 | CONTINUE | EDITE160 |
| | IX = 3 | EDITE161 |
| | DO 560 I=1,5 | EDITE162 |
| 560 | CMOD(I) = ACTION(I) | EDITE163 |
| | GO TO 545 | EDITE164 |
| 105 | MODLP = IST | EDITE165 |
| | MODLQ = ISP | EDITE166 |
| 106 | GO TO (50,120),IJ | EDITE167 |
| 85 | CONTINUE | EDITE168 |
| | IF(INB.EQ.MODLP) GO TO 160 | EDITE169 |
| | IF(ACTION(1).EQ.END) GO TO 83 | EDITE170 |
| 87 | CALL OUTCD(ACTION,S2) | EDITE171 |
| | CALL INCD(ACTION,S1) | EDITE172 |

```

IF(ERFLAG.NE.0) RETURN
IX = 3
IF(ACTION(1).NE.END) GO TO 90
MODLP = 5000
MODLQ = 5000
GO TO 63
120 IF(IA.EQ.0) GO TO 87
GO TO 80
160 IF(IA.EQ.1) GO TO 170
IF(IA.EQ.2) GO TO 180
IF(IY.EQ.1) GO TO 83
IF(IY.EQ.3) GO TO 83
GO TO 200
170 IF(IY.EQ.1) GO TO 83
GO TO 87
180 IF(IY.EQ.1) GO TO 83
IF(IY.EQ.3) GO TO 87
GO TO 200
200 CONTINUE
WRITE (FERR,210)
210 FORMAT(45H MOD CONFLICTS WITH PRIOR MOD - PLEASE REENTER)
GO TO 51
500 CONTINUE
510 CALL OUTCD (ACTION,S2)
IF(ACTION(1).EQ.END) GO TO 520
CALL INCD (ACTION,S1)
IF(ERFLAG.NE.0) RETURN
GO TO 510
520 CONTINUE
M1 = 13
REWIND M1
REWIND S2
IU(1) = M1
IU(2) = S2
CALL INTBUF (IU,IR)
530 CALL INCD (KARD,S2)

```

```

EDITE173
EDITE174
EDITE175
EDITE176
EDITE177
EDITE178
EDITE179
EDITE180
EDITE181
EDITE182
EDITE183
EDITE184
EDITE185
EDITE186
EDITE187
EDITE188
EDITE189
EDITE190
EDITE191
EDITE192
EDITE193
EDITE194
EDITE195
EDITE196
EDITE197
EDITE198
EDITE199
EDITE200
EDITE201
EDITE202
EDITE203
EDITE204
EDITE205
EDITE206
EDITE207
EDITE208

```

```

IF(ERFLAG.NE.0) RETURN
CALL OUTCD(KARD,M1)
IF(KARD(1).NE.END) GO TO 530
RETURN
END
#ELT,1 DORMAN.TERM
SUBROUTINE TERM
C
C      TERMINATES THE RUN
C
C      PROGRAMMER - S. WRAY
C
COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14)
INTEGER ERFLAG,FERR,ACTION,PRTFIL
EQUIVALENCE (PRTFIL,FERR)
IF(ERFLAG.NE.0) WRITE (PRTFIL,10)
10 FORMAT(15H ERROR IS FATAL)
WRITE (PRTFIL,5)
5 FORMAT(37H -- END DORMAN -- RUN TERMINATED --)
STOP
END
#ELT,1 DORMAN.A1TA6
SUBROUTINE A1TA6 (CA84,CA14)
C
C      ENTER WITH 64 WORDS EACH CONTAINING 1 CHARACTER
C      EXIT WITH 14 WORDS EACH CONTAINING 6 CHARACTERS
C
C      PROGRAMMER: VOIT
C
DIMENSION CA14(14),CA84(64)
ENCODE (84,5,CA14) CA84
5 FORMAT(14(6A1))
RETURN
END
#ELT,1 DORMAN.A6TA1
SUBROUTINE A6TA1 (CA14,CA84)

```

```

EDITE209
EDITE210
EDITE211
EDITE212
EDITE213
TERM 3
TERM 5
TERM 6
TERM 7
TERM 8
TERM 9
TERM 10
MISC 2
MISC 3
MISC 4
TERM 12
TERM 13
TERM 14
TERM 15
TERM 20
TERM 21
A1TA6 3
A1TA6 5
A1TA6 6
A1TA6 7
A1TA6 8
A1TA6 9
A1TA6 10
A1TA6 11
A1TA6 12
A1TA6 13
A1TA6 14
A1TA6 15
A1TA6 16
A6TA1 3
A6TA1 5

```

| | | | |
|--------|--|-------|----|
| C | | A6TA1 | 6 |
| C | ENTER WITH 14 WORDS EACH CONTAINING 6 CHARACTERS | A6TA1 | 7 |
| C | EXIT WITH 84 WORDS EACH CONTAINING 1 CHARACTER | A6TA1 | 8 |
| C | | A6TA1 | 9 |
| C | PROGRAMMER: VOIT | A6TA1 | 10 |
| C | | A6TA1 | 11 |
| | DIMENSION CA14(14), CA84(84) | A6TA1 | 12 |
| | DECODE (84,5,CA14) CA84 | A6TA1 | 13 |
| | 5 FORMAT(14(6A1)) | A6TA1 | 14 |
| | RETURN | A6TA1 | 15 |
| | END | A6TA1 | 16 |
| #ELT,I | DORMAN.CONV | CONV | 3 |
| | SUBROUTINE CONV | CONV | 2 |
| C | | CONV | 6 |
| C | CONVERTS DECKS | CONV | 7 |
| C | | CONV | 8 |
| C | PROGRAMMER - S. WRAY | CONV | 9 |
| C | | CONV | 10 |
| | COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2 | FILES | 2 |
| | INTEGER BASIC,MTAPE,FINAL,S1,S2 | FILES | 3 |
| | COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST | 2 |
| | INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST | 3 |
| | COMMON /NAMES/ IVER,ONAME(2),MODNAM(2),BNAME(2) | NAMES | 2 |
| | INTEGER ONAME,MODNAM,BNAME | NAMES | 3 |
| | COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC | 2 |
| | INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC | 3 |
| | EQUIVALENCE (PRTFIL,FERR) | MISC | 4 |
| | DIMENSION IU(3),IR(3) | CONV | 15 |
| | INTEGER VERS | CONV | 16 |
| | DATA VERS/6HVERSIO/ | CONV | 17 |
| | DATA IU,IR/0,0,0,2,2,2/ | CONV | 18 |
| | WRITE(PRTFIL,5) | CONV | 19 |
| | 5 FORMAT(29H ENTER NAME OF REFERENCE DECK/9H READY -) | CONV | 20 |
| | CALL INCD (KARD,5) | CONV | 21 |
| | WRITE (PRTFIL,10) | CONV | 22 |
| 10 | FORMAT(25H ENTER NAME OF FINAL DECK/9H READY -) | CONV | 23 |

| | |
|--|----------|
| CALL INCD (ACTION,5) | CONV 24 |
| IU(1) = BASIC | CONV 25 |
| CALL INIBUF(IU,IR) | CONV 26 |
| ACTION(3) = KARD(1) | CONV 27 |
| ACTION(4) = KARD (2) | CONV 28 |
| KK = 0 | CONV 29 |
| KA = 0 | CONV 30 |
| REWIND BASIC | CONV 31 |
| REWIND 22 | CONV 32 |
| REWIND 23 | CONV 33 |
| C | CONV 34 |
| C ONE OR TWO OR NONE USE PASSES | CONV 35 |
| C | CONV 36 |
| 15 CALL INCD (KARD,BASIC) | CONV 37 |
| IF(ERFLAG.NE.0) RETURN | CONV 36 |
| IF(KARD(5).EQ.VERS) GO TO 15 | CONV 39 |
| IF(KARD(1).EQ.END) GO TO 40 | CONV 40 |
| IF((ACTION(1).EQ.KARD(2)).AND.(ACTION(2).EQ.KARD(3))) KA=KARD(4) | CONV 41 |
| IF((ACTION(3).EQ.KARD(2)).AND.(ACTION(4).EQ.KARD(3))) KK=KARD(4) | CONV 42 |
| GO TO 15 | CONV 43 |
| 40 CONTINUE | GWT13 2 |
| FINAL = 23 | GWT13 3 |
| CALL USE (ACTION(3)) | CONV 74 |
| FINAL = 22 | GWT13 4 |
| CALL USE (ACTION) | CONV 76 |
| 60 CONTINUE | CONV 77 |
| REWIND 22 | CONV 78 |
| REWIND 23 | CONV 79 |
| REWIND 13 | CONV 80 |
| CALL DIFDEC (23,22,13,5) | CONV 81 |
| MODNAM (1) = ACTION (1) | CONV 82 |
| MODNAM (2) = ACTION (2) | CONV 83 |
| RETURN | CONV 84 |
| END | CONV 85 |
| #ELT,1 JORMAN.EXTRAN | EXTRAN 3 |
| SUBROUTINE EXTRAN(ITIME,IDATE) | EXTRAN 5 |

C
C
C
C
C
C
C
C

ENTER WITH 6 CHAR EACH IN ITIM, AND IDTE
EXIT WITH 2 WORDS (6 CHAR AND 2 CHAR)

PROGRAMMER: VOIT

DIMENSION ITIME(2), IDATE(2)
DIMENSION ITMP(3)
DATA ISLS /1H//
DATA INCL/1H./
CALL EXTRAN (9,ITIM,IDTE)
DECODE (6,100,ITIM) (ITMP(1),I=1,3)
100 FORMAT (3A2)
ENCODE (6,110,ITIME) ITMP(1),INCL,ITMP(2),INCL
110 FORMAT (A2,A1,A2,A1)
ITIME(2)=ITMP(3)
DECODE (6,100,IDTE) (ITMP(1),I=1,3)
ENCODE (6,110,IDATE) ITMP(1),ISLS,ITMP(2),ISLS
IDATE(2) = ITMP(3)
RETURN
END

*ELT,I DORMAN.FILBUF

SUBROUTINE FILBUF(BEGIN,END,MAX,BUF ,FILE)

C
C
C
C
C
C
C
C

CALLED BY SUBROUTINES
JIFDEC,RESTOR,AND SYNCBF

PROGRAMMER: VOIT

COMMON /MISC/ERFLAG
COMMON /MISC/FERR
INTEGER FERR
INTEGER ERFLAG
INTEGER BEGIN,END,MAX,BUF(14,20),FILE

EXTRAN 0
EXTRAN 7
EXTRAN 8
EXTRAN 9
EXTRAN10
EXTRAN11
EXTRAN12
EXTRAN13
EXTRAN14
EXTRAN15
GWT25 3
EXTRAN17
EXTRAN18
EXTRAN19
EXTRAN20
EXTRAN21
EXTRAN22
EXTRAN23
EXTRAN24
EXTRAN25
EXTRAN26
EXTRAN27
FILBUFV3
FILBUFV5
FILBUFV6
FILBUFV7
FILBUFV8
FILBUFV9
FILBUF10
FILBUF11
FILBUF12
FILBUF13
FILBUF14
FILBUF15
FILBUF16
FILBUF17

```

      N = BEGIN-1
      IF(N.LT.MAX) GO TO 5
      WRITE (FERR,10) FILE
10    FORMAT(49H FILBUF - INVALID REQUEST TO FILL BUFFER FOR UNIT,I3)
      ERFLAG = 1
      CALL TERM
      5  N = N + 1
      CALL INCD (BUF(1,N),FILE)
      IF(ERFLAG.NE.0) GO TO 16000
      IF (N.GE.MAX) GO TO 15
      IF(BUF(1,N).NE.6H$END 0 ) GO TO 5
15    END = N
      RETURN
16000 CONTINUE
      DO 16001 I=3,14
16001  BUF(I,N)=1H
      BUF(1,N)=6H$END 0
      BUF(2,N)=6HF FILE
      END = N
      RETURN
      END
*ELT,I DORMAN.FNDBUF
      SUBROUTINE FNDBUF (UNIT,X,Y)
C
C      CHECK TO SEE UNIT IS ACTIVE
C      RETURN MODE IN X
C
C      PROGRAMMER - VOIT
C
      COMMON/MISC/ERFLAG,FERR
      INTEGER ERFLAG,FERR
      COMMON /WORK/ILUNT(3),IIRW(3),IUTBL(3,19),IACT(8,3),NFILES
      INTEGER UNIT,X,Y
      DO 10 I=1,3
      J = I
      IF (UNIT.EQ.IACT(1,I)) GO TO 12

```

```

FILBUF18
FILBUF19
FILBUF20
FILBUF21
FILBUF22
FILBUF23
FILBUF24
FILBUF25
FILBUF26
FILBUF27
FILBUF28
FILBUF29
FILBUF30
FILBUF31
FILBUF32
FILBUF33
FILBUF34
FILBUF35
FILBUF36
FILBUF37
FILBUF38
FNDBUF 3
FNDBUF 5
FNDBUF 6
FNDBUF 7
FNDBUF 8
FNDBUF 9
FNDBUF10
FNDBUF11
FNDBUF12
FNDBUF13
WORK 2
FNDBUF15
FNDBUF16
FNDBUF17
FNDBUF18

```

```

10 CONTINUE
   WRITE (FERR,1000) UNIT
1000 FORMAT(28H FNDBUF - FATAL ERROR - UNIT,13,11H NOT ACTIVE)
   ERFLAG = 1
   CALL TERM
12 CONTINUE
   Y = J
   X = IACT(2,J)
   RETURN
   END
#ELT,I DORMAN.GETGEN
   SUBROUTINE GETGEN(NAME,IFILE,LIST,NLIST)
C
C GET GENEALOGY
C PROGRAMMER - B. GOLD
C
   DIMENSION NAME(2),LIST(2,20)
   COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14)
   INTEGER ERFLAG,FERR,ACTION,PRTFIL
   EQUIVALENCE (PRTFIL,FERR)
   COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH
   INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH
   DIMENSION IU(3),IR(3)
   DATA IU/2,2,2/,IU/0,0,0/
   IU(1) = IFILE
   CALL INTBUF(IU,IR)
   REWIND IFILE
   ERFLAG = 0
   LIST(1,1) = NAME(1)
   LIST(2,1) = NAME(2)
   NLIST = 1
10 CALL INCD(KARD,IFILE)
   IF(ERFLAG.NE.0) RETURN
   IF(KARD(1).NE.END ) GO TO 30
   IF(KARD(2).NE.TABLE) GO TO 30
   WRITE (FERR,20) NAME, ((LIST(I,J), I=1,2), J=1,NLIST)

```

```

FNDBUF19
FNDBUF20
FNDBUF21
FNDBUF22
FNDBUF23
FNDBUF24
FNDBUF25
FNDBUF26
FNDBUF27
FNDBUF28
GETGEN 3
GETGEN 5
GETGEN 6
GETGEN 7
GETGEN 8
GETGEN 9
GETGEN10
MISC 2
MISC 3
MISC 4
REST 2
REST 3
GETGEN13
GETGEN14
GETGEN15
GETGEN16
GETGEN17
GETGEN18
GETGEN19
GETGEN20
GETGEN21
GETGEN22
GETGEN23
GETGEN24
GETGEN25
GETGEN26

```

```

20 FORMAT(36H0GETGEN ERROR IN GENEALOGY FOR DECK 2A6/44H COULD NOT FIGETGEN27
*ND LAST DECK IN FOLLOWING LIST-//(5X2A6))
ERFLAG = 1
RETURN
30 IF (KARD(2).NE.LIST(1,NLIST)) GO TO 10
IF (KARD(3).NE.LIST(2,NLIST)) GO TO 10
IF(KARD(4).EQ.BLANK) RETURN
NLIST = NLIST+1
LIST(1,NLIST) = KARD(5)
LIST(2,NLIST) = KARD(6)
GO TO 10
END
#ELT,I DORMAN.INCRT
SUBROUTINE INCRT
C
C      CREATES A DATA BASE
C
C      PROGRAMMER - S. WRAY
C
DIMENSION IU(3),IR(3)
COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2
INTEGER BASIC,MTAPE,FINAL,S1,S2
COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH
INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH
COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14)
INTEGER ERFLAG,FERR,ACTION,PRTFIL
EQUIVALENCE (PRTFIL,FERR)
COMMON /VCARD/ILBL(14)
DATA IR/2,2,2/
IU(1) = 5
IU(2) = BASIC
IU(3) = 0
CALL INTBUF(IU,IR)
REWIND BASIC
CALL EXTRAN (ILBL(12),ILBL(10))
CALL OUTCD (ILBL,BASIC)

```

```

GETGEN28
GETGEN29
GETGEN30
GETGEN31
GETGEN32
GETGEN33
GETGEN34
GETGEN35
GETGEN36
GETGEN37
GETGEN38
INCR T 3
INCR T 5
INCR T 6
INCR T 7
INCR T 8
INCR T 9
INCR T 10
INCR T 11
FILES 2
FILES 3
REST 2
REST 3
MISC 2
MISC 3
MISC 4
INCR T 15
INCR T 16
INCR T 17
INCR T 18
INCR T 19
INCR T 20
INCR T 21
INCR T 22
INCR T 23

```

```

DO 5 I = 1,14
ACTION(I) = BLANK
5 KARD(I) = BLANK
KARD(1) = ENG
KARD(2) = TABLE
KARD(3) = 1HE
ACTION(1) = DDECK
ACTION(2) = 6HTESTOR
ACTION(4) = USES
ACTION(5) = 5H TEST
CALL OUTCD (ACTION,BASIC)
ACTION(4) = BLANK
ACTION(5) = BLANK
ACTION(2) = 5H TEST
CALL OUTCD (ACTION,BASIC)
CALL OUTCD(KARD,BASIC)
ACTION(2) = 6HTESTOR
ACTION(4) = USES
ACTION(5) = 5H TEST
CALL OUTCD(ACTION,BASIC)
KARD(2) = DECK
KARD(3) = BLANK
CALL OUTCD (KARD,BASIC)
ACTION(2) = 5H TEST
ACTION(4) = BLANK
ACTION(5) = BLANK
CALL OUTCD (ACTION,BASIC)
CALL OUTCD (KARD,BASIC)
KARD(2) = FILE
CALL OUTCD (KARD,BASIC)
REWIND 21
CALL ADDX(21)
RETURN
END
*ELT,I DORMAN.INCD
SUBROUTINE INCD (ICRD,IUNIT)

```

```

INCRT 24
INCRT 25
INCRT 26
INCRT 27
INCRT 28
INCRT 29
INCRT 30
INCRT 31
INCRT 32
INCRT 33
INCRT 34
INCRT 35
INCRT 36
INCRT 37
INCRT 38
INCRT 39
INCRT 40
INCRT 41
INCRT 42
INCRT 43
INCRT 44
INCRT 45
INCRT 46
INCRT 47
INCRT 48
INCRT 49
INCRT 50
INCRT 51
INCRT 52
INCRT 53
INCRT 54
INCRT 55
INCRT 56
INCRT 57
INCDM 3
INCDM 5

```

| | | |
|---|--|----------|
| C | | INCDM 6 |
| C | PUT IN FORCE OF +ASG | INCDM 7 |
| C | | INCDM 8 |
| C | MODE 1 - READ 55 CARDS | INCDM 9 |
| C | MODE 2 - READ 1 CARD | INCDM 10 |
| C | MODE 3 - READ 1 CARD, MAY BE PACKED | INCDM 11 |
| C | | INCDM 12 |
| C | PROGRAMMER: VOIT | INCDM 13 |
| C | | INCDM 14 |
| | INTEGER ICRD(14) | INCDM 15 |
| | COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST 2 |
| | INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST 3 |
| | COMMON /WORK/IUNIT(3),IIRW(3),IUTBL(3,19),IACT(8,3),NFILES | WORK 2 |
| | COMMON /BFRS/ XXX(54) | BFRS 2 |
| | COMMON /BFRS/ IMOD1(14,55,2),IWK1(168,2),ITEMP1(84),ITEMP2(84) | BFRS 3 |
| | COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC 2 |
| | INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC 3 |
| | EQUIVALENCE (PRTFIL,FERR) | MISC 4 |
| | DIMENSION ITMP1(1),ITMP2(1) | INCDM 20 |
| | EQUIVALENCE (ITMP1(1),ITEMP1(1)),(ITEMP2(1),ITMP2(1)) | INCDM 21 |
| | IF(IUNIT.EQ.5) GO TO 5 | INCDM 22 |
| | CALL FNDBUF (IUNIT,IX,IY) | INCDM 23 |
| | IF (IX.NE.2) GO TO 100 | INCDM 24 |
| C | | INCDM 25 |
| C | MODE EQ 2 -- READ ONE CARD | INCDM 26 |
| C | | INCDM 27 |
| | IF (IACT(4,IY).EQ.1) GO TO 510 | INCDM 28 |
| | CALL UREAD (IUNIT,ICRD) | INCDM 29 |
| C | | INCDM 30 |
| C | IF END OF FILE, SET FLAG | INCDM 31 |
| C | | INCDM 32 |
| | 20 CONTINUE | INCDM 33 |
| | IF(ICRD(1).NE.END) RETURN | INCDM 34 |
| | IF(ICRD(2).NE.FILE) RETURN | INCDM 35 |
| | IACT(4,IY) = 1 | INCDM 36 |
| | IF(IUNIT.EQ.5) CALL TERM | INCDM 37 |

| | | |
|-----|--|----------|
| | RETURN | INCDM 38 |
| 100 | CONTINUE | INCDM 39 |
| | IF (IX.NE.1) GO TO 200 | INCDM 40 |
| C | | INCDM 41 |
| C | MODE 1 | INCDM 42 |
| C | IS BUFFER EMPTY | INCDM 43 |
| C | | INCDM 44 |
| | IXX = IACT(8,IY) | INCDM 45 |
| | IYY = IACT(7,IY) | INCDM 46 |
| | IF (IACT(4,IY).EQ.1) GO TO 510 | INCDM 47 |
| | GO TO 155 | INCDM 48 |
| C | | INCDM 49 |
| C | CAN WORK AREA TAKE 84 CHARACTERS | INCDM 50 |
| C | | INCDM 51 |
| 117 | CONTINUE | INCDM 52 |
| | IF (IXX.GT.84) GO TO 170 | INCDM 53 |
| | IF (IACT(4,IY).EQ.1) GO TO 170 | INCDM 54 |
| C | | INCDM 55 |
| C | PUT 84 CHAR FROM BUFFER INTO TOP OF WORK AREA | INCDM 56 |
| C | | INCDM 57 |
| | CALL A6TA1(IMOD1(1,IYY,IY),IWK1(IXX+1,IY)) | INCDM 58 |
| | IXX = IXX + 84 | INCDM 59 |
| | IYY = IYY + 1 | INCDM 60 |
| | IF(IYY.NE.56) GO TO 117 | INCDM 61 |
| C | | INCDM 62 |
| C | READ IN NEXT BUFFER FULL | INCDM 63 |
| C | | INCDM 64 |
| 155 | CONTINUE | INCDM 65 |
| | DO 160 I=1,55 | INCDM 66 |
| | READ (IUNIT,156,END=510,ERR=510) (IMOD1(J,1,IY),J=1,770) | INCDM 71 |
| 156 | FORMAT(770A6) | INCDM 72 |
| 160 | CONTINUE | INCDM 74 |
| | IYY = 1 | INCDM 75 |
| | IACT(4,IY) = 0 | INCDM 76 |
| | GO TO 117 | INCDM 77 |
| C | | INCDM 78 |

| | | |
|-----|--|----------|
| C | MOVE CARDS FROM WORK AREA INTO CARD | INCDM 79 |
| C | | INCDM 80 |
| 170 | CONTINUE | INCDM 81 |
| | CALL UNPAC (ITEMP1,ITEMP2,ICH) | INCDM 82 |
| | CALL A1TA6(ITEMP2,ICRD) | INCDM 83 |
| C | | INCDM 84 |
| C | MOVE DOWN WORK AREA | INCDM 85 |
| C | | INCDM 86 |
| | K = IXX-ICH | INCDM 87 |
| | IF(K.EQ.0) GO TO 195 | INCDM 88 |
| | DO 190 I=1,K | INCDM 89 |
| | ICH1 = 1+ICH | INCDM 90 |
| | IWK1(I,IY) = IWK1(ICH1,IY) | INCDM 91 |
| 190 | CONTINUE | INCDM 92 |
| 195 | CONTINUE | INCDM 93 |
| | IXX = K | INCDM 94 |
| | IACT(8,IY)=IXX | INCDM 95 |
| | IACT(7,IY)=IYY | INCDM 96 |
| | GO TO 20 | INCDM 97 |
| 200 | CONTINUE | INCDM 98 |
| | IF(IX.NE.3) GO TO 500 | INCDM 99 |
| C | | INCDM100 |
| C | MODE 3 | INCDM101 |
| C | | INCDM102 |
| 5 | CONTINUE | INCDM103 |
| | CALL UREAD (IUNIT,ICRD) | INCDM104 |
| | CALL A6TA1(ICRD,ITEMP1) | INCDM105 |
| | CALL UNPAC (ITEMP1,ITEMP2,ICH) | INCDM106 |
| | CALL A1TA6(ITEMP2,ICRD) | INCDM107 |
| | IF(BATCH.EQ.IUNIT) CALL OUTCD(ICRD,PRTFIL) | INCDM108 |
| | GO TO 20 | INCDM109 |
| C | | INCDM110 |
| C | ERROR EXIT | INCDM111 |
| C | | INCDM112 |
| 500 | CONTINUE | INCDM113 |
| | WRITE (FERR,1000) IUNIT,IX | INCDM114 |

| | | |
|--------|---|----------|
| 1000 | FORMAT(26H INCD - MODE ERROR -- UNIT,I3,5H MODE,I3) | INCDM115 |
| | ERFLAG = 1 | INCDM116 |
| | CALL TERM | INCDM117 |
| 510 | CONTINUE | INCDM118 |
| | WRITE (FERR,2000) IUNIT | INCDM119 |
| 2000 | FORMAT(27H INCD - END OF FILE ON UNIT,I3) | INCDM120 |
| | ERFLAG = 1 | INCDM121 |
| | IF(IUNIT.EQ.5) CALL TERM | INCDM122 |
| | RETURN | INCDM123 |
| | END | INCDM124 |
| #ELT,I | DORMAN.INSERT | INSERT 3 |
| | SUBROUTINE INSERT(NSTP,II) | INSERT 5 |
| C | | INSERT 6 |
| C | INSERT | INSERT 7 |
| C | | INSERT 8 |
| | COMMON /BFRS/ ISYN(2,20),ISYN1,IT1,IT2,IT3,FULL,CN1,CN2,LIN1,LIN2 | BFRS1 2 |
| | * ,NWBUF,NB1,NB2,ICN1,ICN2 | BFRS1 3 |
| | COMMON /BFRS/ BUF1(14,50),BUF2(14,50),AAA(700) | BFRS1 4 |
| | INTEGER CN1,CN2,BUF1,BUF2 | BFRS1 5 |
| | LOGICAL FULL | BFRS1 6 |
| | INTEGER FIN1,FIN2 | INSERT10 |
| | EQUIVALENCE (FOUT,IT3) | INSERT11 |
| | EQUIVALENCE (FIN2,IT2) | INSERT12 |
| | EQUIVALENCE (FIN1,IT1) | INSERT13 |
| | INTEGER FOUT | INSERT14 |
| | IF(II.NE.0) GO TO 10 | INSERT15 |
| | J = LIN1 - 2 | INSERT16 |
| | WRITE(FOUT,46) J | INSERT17 |
| 46 | FORMAT (7H\$INSERT,I13) | INSERT18 |
| 10 | CONTINUE | INSERT19 |
| 50 | CONTINUE | INSERT20 |
| | LSTCD = (NB2-CN2) + LIN2 | INSERT21 |
| | IF (NSTP.GT.LSTCD) GO TO 100 | INSERT22 |
| | KK = NSTP-LIN2+CN2-1 | INSERT23 |
| | WRITE (FOUT,9) ((BUF2(I,K),I=1,14),K=CN2,KK) | INSERT24 |
| | KK = KK - CN2 + 1 | INSERT25 |

```

      LIN2 = LIN2 + KK
      CN2 = CN2 + KK
      RETURN
100  CONTINUE
      KK = NB2 - CN2 + 1
      WRITE (FOUT,9) ((BUF2(I,K),I=1,14),K=CN2,NB2)
      LIN2 = LIN2+KK
      CALL FILBUF (1,NB2,NWBUF,BUF2,FIN2)
      CN2 = 1
      GO TO 50
9    FORMAT (14A6)
      END
*ELT,I DORMAN.INTBUF
      SUBROUTINE INTBUF (IUNT,IRW)
C
C      ENTER WITH 3 UNITS AND READ/WRITE STATUS
C      INITIALIZE ACTIVE TABLE
C
C      PROGRAMMER: VOIT
C
      COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14)
      INTEGER ERFLAG,FERR,ACTION,PRTFIL
      EQUIVALENCE (PRTFIL,FERR)
      COMMON /WORK/IIUNT(3),IIRW(3),IUTBL(3,19),IACT(8,3),NFILES
      COMMON /BFRS/ XXX(54)
      COMMON /BFRS/ IMOD1(14,55,2),IWK1(168,2),ITEMP1(84),ITEMP2(84)
      DIMENSION IUNT(3),IRW(3)
C
C      IUTBL --- UNIT TABLE
C      1 - UNIT NUMBER
C      2 - MODE
C
C      IACT --- 3 ACTIVE FILES
C      1 - UNIT NUMBER
C      2 - MODE
C      3 - 0=READ ONLY, 1=WRITE ONLY, 2=READ OR WRITE

```

```

INSERT26
INSERT27
INSERT29
INSERT30
INSERT31
INSERT32
INSERT33
INSERT34
INSERT35
INSERT36
INSERT37
INSERT38
INTBUF 3
INTBUF 5
INTBUF 6
INTBUF 7
INTBUF 8
INTBUF 9
INTBUF10
INTBUF11
MISC 2
MISC 3
MISC 4
WORK 2
BFRS 2
BFRS 3
INTBUF15
INTBUF16
INTBUF17
INTBUF18
INTBUF19
INTBUF20
INTBUF21
INTBUF22
INTBUF23
INTBUF24

```

```

C
C      3 - READ=0,WRITE=1
C      4 - FILE STATUS 1=END OF FILE, 2=REWIND,0=IN USE
C      5 - BUFFER INDEX
C      6 - WORK AREA INDEX
C      7 - BUFFER CARD COUNTER
C      8 - WORK AREA CHARACTER COUNTER
C

```

```

      DIMENSION IMOD2(1780)
      EQUIVALENCE (IMOD2(1),IMOD1(1,1,1))
      KK = NFILES
      DO100 L=1,3
      DO 10 I=1,KK
      J = I
      IF(IUNT(L).EQ.0) GO TO 11
      IF (IUNT(L).EQ.IUTBL(1,I)) GO TO 12
10    CONTINUE
      GO TO 260
11    DO 9 K = 1,6
      9    IACT(K,L) = 0
      GO TO 100

```

```

C
C      UNIT FOUND
C
12    CONTINUE
      IACT(1,L) = IUNT(L)
      IACT(2,L) = IUTBL(2,J)
      IACT(3,L) = IRW(L)
      K = IACT(1,L)
      IACT(4,L) = 2
100  CONTINUE
C
C      ASSIGN BUFFER FOR MODE 1
C

```

```

      K = 0
      DO 200 L=1,3

```

```

INTBUF25
INTBUF26
INTBUF27
INTBUF28
INTBUF29
INTBUF30
INTBUF31
INTBUF32
INTBUF33
INTBUF34
INTBUF35
INTBUF36
INTBUF37
INTBUF38
INTBUF39
INTBUF40
INTBUF41
INTBUF42
INTBUF43
INTBUF44
INTBUF45
INTBUF46
INTBUF47
INTBUF48
INTBUF49
INTBUF50
INTBUF51
INTBUF52
INTBUF53
INTBUF54
INTBUF55
INTBUF56
INTBUF57
INTBUF58
INTBUF59
INTBUF60

```

```

        IF (IACT(2,L).NE.1) GO TO 150
        K = K+1
        IF (K.GT.2) GO TO 250
        DO 5 I=1,1780
        IMOD2(1)=0
5      CONTINUE
        IACT(5,L) = K
        IACT(6,L) = K
        GO TO 160
150    IACT(5,L) = 0
C
C      FORCE REFERENCE TO +ASG
C
        IACT(6,L) = 0
        IACT(7,L) = 0
        IACT(8,L) = 0
        GO TO 200
160    CONTINUE
        IACT(7,L) = 1
        IACT(8,L) = 0
200    CONTINUE
        RETURN
C
C      ERROR EXIT
C
260    CONTINUE
        WRITE (FERR,1000) IUNT(L)
1000  FORMAT(14H INTBUF - UNIT,I3,24H NOT FOUND IN UNIT TABLE)
1001  ERFLAG = 1
        CALL TERM
250    CONTINUE
        WRITE (FERR,1010) IUNT(L),(IACT(J,L),J=1,8)
1010  FORMAT(29H INTBUF - CANNOT ASSIGN UNITS,I3/5X,8I10)
        GO TO 1001
        END
#ELT,I DORMAN.LABLER

```

```

INTBUF61
INTBUF62
INTBUF63
INTBUF64
INTBUF65
INTBUF66
INTBUF67
INTBUF68
INTBUF69
INTBUF70
INTBUF71
INTBUF72
INTBUF73
INTBUF74
INTBUF75
INTBUF76
INTBUF77
INTBUF78
INTBUF79
INTBUF80
INTBUF81
INTBUF82
INTBUF83
INTBUF84
INTBUF85
INTBUF86
INTBUF87
INTBUF88
INTBUF89
INTBUF90
INTBUF91
INTBUF92
INTBUF93
INTBUF94
INTBUF95
LABLER 3

```

| | | |
|----|--|----------|
| C | SUBROUTINE LABLER (IVER,FILE1,FILE2) | LABLER 5 |
| C | WRITE LABEL AND VERSION ON FILE2 | LABLER 6 |
| C | THEN COPY FILE1 TO FILE2 | LABLER 7 |
| C | PROGRAMMER: VOIT | LABLER 8 |
| C | COMMON /MISC/ERFLAG,FERR | LABLER9 |
| C | INTEGER ERFLAG,FERR | LABLER10 |
| C | DIMENSION IFL(3),IRW(3) | LABLER11 |
| C | DIMENSION ICRD(14) | LABLER12 |
| C | COMMON /VCARD/ILBL(14) | LABLER13 |
| C | INTEGER FILE1,FILE2 | LABLER14 |
| C | FILL IN VERSION NUMBER,DATE AND TIME IN ILBL | LABLER15 |
| C | ILBL(7) = IVER | LABLER16 |
| C | CALL EXTRAN (ILBL(9),ILBL(11)) | LABLER17 |
| C | INITIALIZE FILES AND TRANSFER DATA | LABLER18 |
| C | IFL(1) = FILE1 | LABLER19 |
| C | IFL(2) = FILE2 | LABLER20 |
| C | IFL(3) = 0 | LABLER21 |
| C | IRW(1) = 0 | LABLER22 |
| C | IRW(2) = 1 | LABLER23 |
| C | CALL INIBUF (IFL,IRW) | LABLER24 |
| C | CALL OUTCD (ILBL,FILE2) | LABLER25 |
| C | CALL INCJ (ICRD,FILE1) | LABLER26 |
| C | IF(ERFLAG.NE.0) GO TO 50 | LABLER27 |
| 30 | CONTINUE | LABLER28 |
| C | CALL INCJ (ICRD,FILE1) | LABLER29 |
| C | IF(ERFLAG.NE.0) GO TO 50 | LABLER30 |
| C | CALL OUTCD (ICRD,FILE2) | LABLER31 |
| C | IF(ICRD(1).NE.6H\$END 0) GO TO 30 | LABLER32 |
| C | IF(ICRD(2).NE.6HF FILE) GO TO 30 | LABLER33 |
| | | LABLER34 |
| | | LABLER35 |
| | | LABLER36 |
| | | LABLER37 |
| | | LABLER38 |
| | | LABLER39 |
| | | LABLER40 |

```

      RETURN
50 WRITE(FERR,51)
51 FORMAT(41H LABEL - NEW DATA FILE COULD NOT BE SAVED )
      ERFLAG = 1
      RETURN
      END
*ELT,I DORMAN.LCRD
      SUBROUTINE LCRD(FILE,ISTRT,ISTP,PRTFIL)
C
C       LIST CARDS ON PRTFIL
C
C       PROGRAMMER: VOIT
C
      INTEGER PRTFIL,FILE,CARD(14)
      IF(PRTFIL.NE.6) WRITE (PRTFIL,6)
6  FORMAT(1H1)
      WRITE (PRTFIL,5)
5  FORMAT(1H0)
      IF(ISTRT.LT.0) ISTRT=0
      IF (ISTP.LT.1STRT) GO TO 50
10  CONTINUE
      IF(ISTRT.EQ.0) GO TO 25
      DO 20 I=1,ISTRT
      CALL INCD (CARD,FILE)
      IF(CARD(1).NE.6H$END 0 ) GO TO 20
      IF(CARD(2).EQ.6HF FILE ) GO TO 100
20  CONTINUE
25  CONTINUE
      DO 30 I=ISTRT,ISTP
      CALL INCD(CARD,FILE)
      DO 27 L=1,14
      J = 15 - L
      IF(CARD(J).NE.1H ) GO TO 28
27  CONTINUE
28  CONTINUE
      WRITE (PRTFIL,26) I,(CARD(LL),LL=1,J)

```

```

LABLER+1
LABLER+2
LABLER+3
LABLER+4
LABLER+5
LABLER+6
LCRD 3
LCRD 5
LCRD 6
LCRD 7
LCRD 8
LCRD 9
LCRD 10
LCRD 11
LCRD 12
LCRD 13
LCRD 14
LCRD 15
LCRD 16
LCRD 17
LCRD 18
LCRD 19
LCRD 20
LCRD 21
LCRD 22
LCRD 23
LCRD 24
LCRD 25
LCRD 26
LCRD 27
LCRD 28
LCRD 29
LCRD 30
LCRD 31
LCRD 32
LCRD 33

```

| | | | |
|--------|--|--------|----|
| 26 | FORMAT(15,1X,1+A6) | LGRD | 34 |
| | IF(CARD(1).NE.6HEND U) GO TO 24 | LGRD | 35 |
| | IF(CARD(2).EQ.6HF FILE) GO TO 100 | LGRD | 36 |
| 24 | CONTINUE | LGRD | 37 |
| 30 | CONTINUE | LGRD | 38 |
| | WRITE (PRTFIL,5) | LGRD | 39 |
| | RETURN | LGRD | 40 |
| 50 | ISTP = ISTRT | LGRD | 41 |
| | GO TO 10 | LGRD | 42 |
| 100 | CONTINUE | LGRD | 43 |
| | IF(PRTFIL.NE.6) RETURN | LGRD | 44 |
| | WRITE (PRTFIL,1000) I | LGRD | 45 |
| 1000 | FORMAT(45H LGRD - COUNT TOO HIGH, FOUND END OF FILE AT ,15) | LGRD | 46 |
| | WRITE (PRTFIL,5) | LGRD | 47 |
| | RETURN | LGRD | 48 |
| | END | LGRD | 49 |
| #ELT,I | DORMAN.LISTER | LISTER | 3 |
| | SUBROUTINE LISTER | LISTER | 5 |
| C | | LISTER | 6 |
| C | LIST OPTIONS SUBROUTINE | LISTER | 7 |
| C | | LISTER | 8 |
| C | PROGRAMMER - S. WRAY | LISTER | 9 |
| C | | LISTER | 10 |
| | DIMENSION IU(3),IR(3) | LISTER | 11 |
| | COMMON /NAMES/ IVER,DNAME(2),MODNAM(2),BNAME(2) | NAMES | 2 |
| | INTEGER DNAME,MODNAM,BNAME | NAMES | 3 |
| | COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2 | FILES | 2 |
| | INTEGER BASIC,MTAPE,FINAL,S1,S2 | FILES | 3 |
| | COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC | 2 |
| | INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC | 3 |
| | EQUIVALENCE (PRTFIL,FERR) | MISC | 4 |
| | WRITE (PRTFIL,66) | LISTER | 15 |
| 66 | FORMAT(63H ENTER LIST OPTION (CONTENTS, GENEALOGY, COUNT, CARDS OR | LISTER | 16 |
| | * PRINT) /9H READY -) | LISTER | 17 |
| | CALL INCD (KARD,5) | LISTER | 18 |
| | IF(KARD(1).EQ.6HCONTEN) GO TO 69 | LISTER | 19 |

| | |
|---|--------------|
| IF(KARD(1).EQ.6HGENEAL) GO TO 67 | LISTER20 |
| IF(KARD(1).EQ.6HCOUNT) GO TO 70 | LISTER21 |
| IF(KARD(1).EQ.6HCARDS) GO TO 80 | LISTER22 |
| IF(KARD(1).EQ.6HPRINT) GO TO 90 | LISTER23 |
| RETURN | LISTER24 |
| 67 WRITE (PRTFIL,68) | LISTER25 |
| 68 FORMAT(27H ENTER NAME OF DECK DESIRED/9H READY -) | LISTER26 |
| CALL INCD(ACTION,5) | LISTER27 |
| CALL LISTG(ACTION,BASIC) | LISTER28 |
| RETURN | LISTER29 |
| 69 CALL LISTTC(BASIC) | LISTER30 |
| 100 RETURN | LISTER31 |
| 70 CONTINUE | LISTER32 |
| IX = 1 | LISTER33 |
| GO TO 75 | LISTER34 |
| 80 CONTINUE | LISTER35 |
| IX = 2 | LISTER36 |
| GO TO 75 | LISTER37 |
| 90 CONTINUE | LISTER38 |
| IX = 3 | LISTER39 |
| 75 CONTINUE | LISTER40 |
| 71 FORMAT(19H ENTER NAME OF DECK/18H OR FILE NUMBER/24H OR | CURRLISTER41 |
| *ENT FILE NAME/11H OR DONE/9H READY -) | LISTER42 |
| WRITE (PRTFIL,71) | LISTER43 |
| CALL INCD (KARD,5) | LISTER44 |
| IF(KARD(1).EQ.4HDONE) RETURN | LISTER45 |
| ITAPE=0 | LISTER46 |
| CALL VALUE (KARD,V,IERR) | LISTER47 |
| IF(IERR.EQ.0) ITAPE = V | LISTER48 |
| IF(KARD(1).EQ.6HBASIC) ITAPE = BASIC | LISTER49 |
| IF(KARD(1).EQ.6HFINAL) ITAPE = FINAL | LISTER50 |
| IF((KARD(1).EQ.BNAME(1)).AND.(KARD(2).EQ.BNAME(2))) ITAPE = FINAL | LISTER51 |
| FINAL = 14 | LISTER52 |
| IF(ITAPE.EQ.0) CALL USE (KARD) | LISTER53 |
| IF(ERFLAG.NE.0) RETURN | LISTER54 |
| IF(ITAPE.EQ.0) ITAPE=FINAL | LISTER55 |

```

IU(1) = 5
IU(2) = ITAPE
IU(3) = 0
IR(1) = 2
IR(2) = 2
IR(3) = 2
REWIND ITAPE
CALL INTBUF(IU, IR)
IF(IX.EQ.1) GO TO 86
IF(IX.EQ.3) GO TO 95
81 WRITE (PRTFIL,82)
82 FORMAT(27H ENTER NUMBER OF FIRST CARD/9H READY - )
CALL INCD (KARD,5)
CALL VALUE (KARD,V,IERR)
IF (IERR.NE.0) GO TO 81
ISTART = V
83 WRITE (PRTFIL,84)
84 FORMAT(27H ENTER NUMBER OF LAST CARD /9H READY - )
CALL INCD (KARD,5)
ISTOP = 0
IF(KARD(1).EQ.1H ) ISTOP = ISTART
IF(ISTOP.NE.0) GO TO 85
CALL VALUE (KARD,V,IERR)
IF(IERR.NE.0) GO TO 83
ISTOP = V
85 CONTINUE
CALL LCRD (ITAPE,ISTART,ISTOP,6)
GO TO 80
86 CALL COUNT(ITAPE)
GO TO 79
95 CONTINUE
CALL LCRD(ITAPE,0,50000,20)
GO TO 90
END
#ELT,I DORMAN.LISTG
SUBROUTINE LISTG(NAME,IFILE)

```

```

LISTER56
LISTER57
LISTER58
LISTER59
LISTER60
LISTER61
LISTER62
LISTER63
LISTER64
LISTER65
LISTER66
LISTER67
LISTER68
LISTER69
LISTER70
LISTER71
LISTER72
LISTER73
LISTER74
LISTER75
LISTER76
LISTER77
LISTER78
LISTER79
LISTER80
LISTER81
LISTER82
LISTER83
LISTER84
LISTER85
LISTER86
LISTER87
LISTER88
LISTER89
LISTG 3
LISTG 5

```

| | | |
|--------|--|----------|
| C | | LISTG 6 |
| C | LIST GENEALOGY | LISTG 7 |
| C | PROGRAMMER - B. GOLD | LISTG 8 |
| C | | LISTG 9 |
| | DIMENSION LIST(2,20), NAME(2) | LISTG 10 |
| | COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC 2 |
| | INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC 3 |
| | EQUIVALENCE (PRTFIL,FERR) | MISC 4 |
| C | | LISTG 12 |
| | CALL GETGEN(NAME,IFILE,LIST,NLIST) | LISTG 13 |
| | IF (ERFLAG.NE.0) RETURN | LISTG 14 |
| | WRITE (PRTFIL,10) NAME | LISTG 15 |
| 10 | FORMAT(20H0GENEALOGY FOR DECK 2A6) | LISTG 16 |
| | IF (NLIST.EQ.1) GO TO 40 | LISTG 17 |
| | DO 20 N = 2,NLIST | LISTG 16 |
| 20 | WRITE (PRTFIL,30) LIST(1,N-1),LIST(2,N-1), LIST(1,N),LIST(2,N) | LISTG 19 |
| 30 | FORMAT(7H \$DECK 2A6,6H USES ,2A6) | LISTG 20 |
| 40 | WRITE (PRTFIL,50) LIST(1,NLIST), LIST(2,NLIST) | LISTG 21 |
| 50 | FORMAT(7H \$DECK 2A6) | LISTG 22 |
| | WRITE (PRTFIL,60) | LISTG 23 |
| 60 | FORMAT(1H0) | LISTG 24 |
| | RETURN | LISTG 25 |
| | END | LISTG 26 |
| #ELT,I | DORMAN,LISTTC | LISTTC 3 |
| | SUBROUTINE LISTTC(IFILE) | LISTTC 5 |
| C | | LISTTC 6 |
| C | LIST TABLE OF CONTENTS | LISTTC 7 |
| C | | LISTTC 8 |
| C | PROGRAMMER - B. GOLD | LISTTC 9 |
| C | | LISTTC10 |
| | DIMENSION IU(3),IR(3) | LISTTC11 |
| | COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST 2 |
| | INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST 3 |
| | COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC 2 |
| | INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC 3 |
| | EQUIVALENCE (PRTFIL,FERR) | MISC 4 |

| | |
|---|----------|
| INTEGER VERS | LISTTC14 |
| DATA VERS/6HVERSI0/ | LISTTC15 |
| DATA IR/2,2,2/,IU/0,0,0/ | LISTTC16 |
| IU(1) = IFILE | LISTTC17 |
| CALL INTBUF(IU,IR) | LISTTC18 |
| REWIND IFILE | LISTTC19 |
| ERFLAG = 0 | LISTTC20 |
| WRITE (PRTFIL,10) | LISTTC21 |
| 10 FORMAT(21H0 TABLE OF CONTENTS/) | LISTTC22 |
| 20 CALL INCD(KARD,IFILE) | LISTTC23 |
| IF(ERFLAG.NE.0) RETURN | LISTTC24 |
| IF(KARD(5).EQ.VERS) GO TO 25 | LISTTC25 |
| IF((KARD(1).NE.DDECK).AND.(KARD(1).NE.END)) GO TO 60 | LISTTC26 |
| 25 CONTINUE | LISTTC27 |
| CALL OUTCD (KARD,PRTFIL) | LISTTC28 |
| IF(KARD(1).NE.END) GO TO 20 | LISTTC29 |
| IF(KARD(2).NE.TABLE) GO TO 20 | LISTTC30 |
| RETURN | LISTTC31 |
| 60 ERFLAG = 1 | LISTTC32 |
| WRITE (FERR,70) KARD | LISTTC33 |
| 70 FORMAT(+9H0LISTC - ERRONEOUS CARD WITHIN TABLE OF CONTENTS-/5X,14A | LISTTC34 |
| *6) | LISTTC35 |
| GO TO 20 | LISTTC36 |
| END | LISTTC37 |
| *ELT,I DORMAN.OPT | OPT 3 |
| SUBROUTINE OPT | OPT 5 |
| C | OPT 6 |
| C | OPT 7 |
| C | OPT 8 |
| C | OPT 9 |
| C | OPT 10 |
| C | OPT 11 |
| COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC 2 |
| INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC 3 |
| EQUIVALENCE (PRTFIL,FERR) | MISC 4 |
| WRITE (PRTFIL,5) | OPT 13 |

| | | | |
|-----|--|-----|----|
| 5 | FORMAT(25H OPTION LIST (SHORT FORM) / | OPT | 14 |
| | *57H 1. CREATE 2. USE (U) 3. OPTION (O) 4. SAVE (S) / | OPT | 15 |
| | *57H 5. ADD (A) 6. DELETE (D) 7. EDIT (E) 8. CONVERT (C) / | OPT | 16 |
| | *57H 9. REPLACE (R) 10. LIST (L) 11. DONE 12. TAPE LIST / | OPT | 17 |
| | *) | OPT | 18 |
| 9 | WRITE (PRTFIL,10) | OPT | 19 |
| 10 | FORMAT(39H ENTER NUMBER OF OPTION TO BE EXPLAINED/8H OR DONE/ | OPT | 20 |
| | *9H READY -) | OPT | 21 |
| 15 | CALL INCD (KARD,5) | OPT | 22 |
| | IF(KARD(1).EQ.4HDONE) RETURN | OPT | 23 |
| | CALL VALUE (KARD,V,IERR) | OPT | 24 |
| | IF(IERR.EQ.0) GO TO 20 | OPT | 25 |
| 17 | WRITE (PRTFIL,16) | OPT | 26 |
| 16 | FORMAT(26H ILLEGAL ENTRY - TRY AGAIN/9H READY -) | OPT | 27 |
| | GO TO 15 | OPT | 28 |
| 20 | I = V | OPT | 29 |
| | IF(I.LT.1) GO TO 17 | OPT | 30 |
| | IF(I.GT.12) GO TO 17 | OPT | 31 |
| | GO TO (101,102,103,104,105,106,107,108,109,110,111,112,113 | OPT | 32 |
| | *),I | OPT | 33 |
| 101 | WRITE (PRTFIL,201) | OPT | 34 |
| 201 | FORMAT(71H IF THE KEY WORD -CREATE- IS ENTERED, THE PROGRAM WILL GOPT | OPT | 35 |
| | *ENERATE A DUMMY/63H DATA BASE AND THEN USE THE ADD OPTION TO ADD DOPT | OPT | 36 |
| | *ECKS FROM TAPE21/29H CREATE MAY BE USED ONLY ONCE) | OPT | 37 |
| | GO TO 9 | OPT | 38 |
| 102 | WRITE (PRTFIL,202) | OPT | 39 |
| 202 | FORMAT(71H IF THE KEY WORD -USE- IS ENTERED, THE PROGRAM WILL REQUOPT | OPT | 40 |
| | *EST THE NAME OF/74H A DECK TO BE EXTRACTED FROM THE DATA BASE. THOPT | OPT | 41 |
| | *ERE WILL BE A DELAY FOR THE/11H EXTRACTION) | OPT | 42 |
| | GO TO 9 | OPT | 43 |
| 103 | WRITE(PRTFIL,203) | OPT | 44 |
| 203 | FORMAT(70H IF THE KEY WORD -OPTION- IS ENTERED, THE USER IS PROVIDOPT | OPT | 45 |
| | *ED WITH A LIST/68H AND THE DESCRIPTION OF EACH ITEM IN THE LIST ASOPT | OPT | 46 |
| | * REQUESTED VIA INPUT) | OPT | 47 |
| | GO TO 9 | OPT | 48 |
| 104 | WRITE (PRTFIL,204) | OPT | 49 |

| | | |
|-----|--|--------|
| 204 | FORMAT(72H IF THE KEY WORD -SAVE- IS ENTERED, THE USER WILL HAVE TOPT | 50 |
| | *HE CAPABILITY TO/71H SAVE THE MOST RECENT VERSION OF THE DATA BASEOPT | 51 |
| | * OR SAVE A BASIC DECK FOR/60H INPUT TO DORCA OR SAVE A MOD DECK TOOPT | 52 |
| | * BE RETAINED EXTERNALLY) | OPT 53 |
| | GO TO 9 | OPT 54 |
| 105 | WRITE (PRTFIL,205) | OPT 55 |
| 205 | FORMAT(71H IF THE KEY WORD -ADD- IS ENTERED, THE PROGRAM WILL BEGIOPT | 56 |
| | *N TO LOAD DECKS/42H FROM TAPE21 FOR ADDITION TO THE DATA BASE) | OPT 57 |
| | GO TO 9 | OPT 58 |
| 106 | WRITE (PRTFIL,206) | OPT 59 |
| 206 | FORMAT(71H IF THE KEY WORD -DELETE- IS ENTERED, THE PROGRAM WILL ROPT | 60 |
| | *EQUEST THE NAME/72H OF THE DECK TO BE DELETED FROM THE DATA BASE. OPT | 61 |
| | * THE DECK IS THEN DELETED) | OPT 62 |
| | GO TO 9 | OPT 63 |
| 107 | WRITE (PRTFIL,207) | OPT 64 |
| 207 | FORMAT(71H IF THE KEY WORD -EDIT- IS ENTERED, THE USER IS DIRECTEDOPT | 65 |
| | * TO NAME A DECK/71H IN THE DATA BASE AND THEN ENTER A MOD DECK VIAOPT | 66 |
| | * THE TERMINAL WHICH WILL/43H ALTER THE ORIGINAL DECK VIA THE USE OPT | 67 |
| | *PTION) | OPT 68 |
| | GO TO 9 | OPT 69 |
| 108 | WRITE (PRTFIL,208) | OPT 70 |
| 208 | FORMAT(73H IF THE KEY WORD -CONVERT- IS ENTERED, THE PROGRAM WILL OPT | 71 |
| | *REQUEST THE NAMES/76H OF TWO DECKS IN THE DATA BASE. THE TWO DECKOPT | 72 |
| | *S WILL BE COMPARED CARD BY CARD/74H AND A MOD DECK WILL BE GENERATOPT | 73 |
| | *ED THAT WILL BE SMALLER THAN THE BASIC DECK/68H BUT EQUIVALENT IN OPT | 74 |
| | *DATA CONTENT WHEN APPLIED TO THE OTHER BASIC DECK) | OPT 75 |
| | GO TO 9 | OPT 76 |
| 109 | WRITE (PRTFIL,209) | OPT 77 |
| 209 | FORMAT(74H IF THE KEY WORD -REPLACE- IS ENTERED, THE USER WILL BE OPT | 78 |
| | *ALLOWED TO REPLACE/63H A DECK IN THE DATA BASE IF THERE IS NOT A MOPT | 79 |
| | *OD DECK NEEDING THE/14H ORIGINAL DECK) | OPT 80 |
| | GO TO 9 | OPT 81 |
| 110 | WRITE (PRTFIL,210) | OPT 82 |
| 210 | FORMAT(53H IF THE KEY WORD -LIST- IS ENTERED, THE USER CAN LIST/ OPT | 83 |
| | *36H TABLE OF CONTENTS OF THE DATA BASE / | OPT 84 |
| | *40H GENEALOGY OF ANY DECK IN THE DATA BASE / | OPT 85 |

| | | |
|--|-------|-----|
| *48H COUNT OF CARDS ON ANY TAPE USED BY THE PROGRAM / | OPT | 86 |
| *50H INDIVIDUAL CARDS ON ANY TAPE USED BY THE PROGRAM / | OPT | 87 |
| *55H PRINTOUT OF THE COMPLETE CONTENTS OF ANY TAPE OR DECK) | OPT | 88 |
| GO TO 9 | OPT | 89 |
| 111 WRITE (PRTFIL,211) | OPT | 90 |
| 211 FORMAT(58H IF THE KEY WORD - DONE- IS ENTERED, THE PROGRAM TERMINA | OPT | 91 |
| *ES) | OPT | 92 |
| GO TO 9 | OPT | 93 |
| 112 WRITE (PRTFIL,212) | OPT | 94 |
| 212 FORMAT(44H THIS IS A LIST OF TAPESUSED BY THE PROGRAM/ | OPT | 95 |
| *62H TAPE1 - SOURCE DATA BASE TAPE14 - CURRENT BASIC DECK | /OPT | 96 |
| *62H TAPE2 - ALTERNATE DATA BASE TAPE20 - OUTPUT DECK LISTINGS | /OPT | 97 |
| *62H TAPE3 - ALTERNATE DATA BASE TAPE21 - INPUT BASIC DECKS | /OPT | 98 |
| *62H TAPE4 - SAVED DATA BASE TAPE22 - SCRATCH | /OPT | 99 |
| *62H TAPE5 - CONSOLE INPUT TAPE23 - SCRATCH | /OPT | 100 |
| *62H TAPE6 - CONSOLE OUTPUT TAPE24 - SAVED MOD DECKS | /OPT | 101 |
| *62H TAPE11 - INPUT MOD DECKS TAPE25 - SCRATCH | /OPT | 102 |
| *62H TAPE12 - DORGA DATA DECK TAPE26 - SCRATCH | /OPT | 103 |
| *62H TAPE13 - CURRENT MOD DECK TAPE27 - SCRATCH | /OPT | 104 |
| *) | OPT | 105 |
| GO TO 9 | OPT | 106 |
| 113 WRITE (PRTFIL,213) | OPT | 107 |
| 213 FORMAT(4H 13.) | OPT | 108 |
| GO TO 9 | OPT | 109 |
| END | OPT | 110 |
| *ELT,I DORMAN.OUTCD | OUTCD | 3 |
| SUBROUTINE OUTCD (ICRD,IUNIT) | OUTCD | 5 |
| C PUT IN FORCE CLOSE | OUTCD | 6 |
| C | OUTCD | 7 |
| C MODE 1 - WRITE 55 CARDS | OUTCD | 8 |
| C MODE 2 - WRITE 1 CARD | OUTCD | 9 |
| C | OUTCD | 10 |
| C PROGRAMMER: VOIT | OUTCD | 11 |
| C | OUTCD | 12 |
| COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST | 2 |
| INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST | 3 |

| | | |
|--|-------|----|
| COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC | 2 |
| INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC | 3 |
| EQUIVALENCE (PRTFIL,FERR) | MISC | 4 |
| COMMON /WORK/IUNIT(3),IIRW(3),IUTBL(3,19),IACT(8,3),NFILES | WORK | 2 |
| COMMON /BFRS/ XXX(54) | BFRS | 2 |
| COMMON /BFRS/ IMOD1(14,55,2),IWK1(168,2),ITEMP1(84),ITEMP2(84) | BFRS | 3 |
| DIMENSION ICRD(14) | OUTCD | 17 |
| DIMENSION ITMP1(84),ITMP2(84) | OUTCD | 18 |
| EQUIVALENCE (ITMP1(1),ITEMP1(1)),(ITEMP2(1),ITMP2(1)) | OUTCD | 19 |
| DATA IIX,IYY,IIZ /14,55,2/ | OUTCD | 20 |
| DATA IIW /84/ | OUTCD | 21 |
| IF(IUNIT.EQ.6) GO TO 8 | OUTCD | 22 |
| CALL FNDBUF (IUNIT,IX,IY) | OUTCD | 23 |
| IF (IX.NE.2) GO TO 100 | OUTCD | 24 |
| | OUTCD | 25 |
| MODE EQ 2 - WRITE ONE CARD | OUTCD | 26 |
| | OUTCD | 27 |
| IF (IACT(4,IY).EQ.1) GO TO 510 | OUTCD | 28 |
| 8 CONTINUE | OUTCD | 29 |
| DO 2 I=1,14 | OUTCD | 30 |
| J=15-I | OUTCD | 31 |
| IF(ICRD(J).NE.BLANK) GO TO 3 | OUTCD | 32 |
| 2 CONTINUE | OUTCD | 33 |
| 3 CONTINUE | OUTCD | 34 |
| IF(IUNIT.EQ.6) GO TO 10 | OUTCD | 35 |
| WRITE (IUNIT,5) (ICRD(I),I=1,J) | OUTCD | 36 |
| 5 FORMAT(14A6) | OUTCD | 37 |
| IF(ICRD(1).NE.END) RETURN | OUTCD | 38 |
| IF(ICRD(2).NE.FILE) RETURN | OUTCD | 39 |
| IACT(4,IY) = 1 | OUTCD | 40 |
| END FILE IUNIT | OUTCD | 41 |
| RETURN | OUTCD | 42 |
| | OUTCD | 43 |
| SPECIAL FORMAT FOR TERMINAL OUTPUT | OUTCD | 44 |
| | OUTCD | 45 |
| 10 WRITE (IUNIT,11) (ICRD(I),I=1, J) | OUTCD | 46 |

| | | |
|-----|---|----------|
| 11 | FORMAT(1X,14A6) | OUTCD 47 |
| | RETURN | OUTCD 48 |
| C | | OUTCD 49 |
| C | MODE 1 | OUTCD 50 |
| C | | OUTCD 51 |
| 100 | CONTINUE | OUTCD 52 |
| | IF(IX.NE.1) GO TO 500 | OUTCD 53 |
| | IXX = IACT(8,IY) | OUTCD 54 |
| | IYY = IACT(7,IY) | OUTCD 55 |
| | IF (IXX.LT.IIW) GO TO 120 | OUTCD 56 |
| C | | OUTCD 57 |
| C | MOVE 84 CHAR FROM WORK AREA INTO BUFFER | OUTCD 58 |
| C | | OUTCD 59 |
| | CALL A1TA6 (IWK1(1,IY),IMOD1(1,IYY,IY)) | OUTCD 60 |
| | IXX = IXX-84 | OUTCD 61 |
| | IYY = IYY+1 | OUTCD 62 |
| C | | OUTCD 63 |
| C | MOVE REMAINING CHAR DOWN | OUTCD 64 |
| C | | OUTCD 65 |
| | DO 110 I=1,IXX | OUTCD 66 |
| | IZ = I+84 | OUTCD 67 |
| | IWK1 (1,IY) = IWK1 (IZ,IY) | OUTCD 68 |
| 110 | CONTINUE | OUTCD 69 |
| C | | OUTCD 70 |
| C | MOVE NEW CARD INTO WORK AREA | OUTCD 71 |
| C | | OUTCD 72 |
| 120 | CONTINUE | OUTCD 73 |
| | CALL A6TA1 (ICKU,ITMP1) | OUTCD 74 |
| | CALL PACCON(ITEMP1,ITMP2,1CH) | OUTCD 75 |
| | DO 130 I=1,1CH | OUTCD 76 |
| | ICH1 = IXX+I | OUTCD 77 |
| | IWK1(ICH1,IY) = ITMP2(I) | OUTCD 78 |
| 130 | CONTINUE | OUTCD 79 |
| | IXX = IXX+1CH | OUTCD 80 |
| C | | OUTCD 81 |
| C | CHECK FOR END OF FILE | OUTCD 82 |

| | | |
|-----|---|----------|
| C | IAC(7,IY) = IYY | OUTCD 83 |
| | IAC(8,IY) = IXX | OUTCD 84 |
| | IF(ICRD(1).NE.6HSEND 0) GO TO 135 | OUTCD 85 |
| | IF(ICRD(2).EQ.6HF FILE) GO TO 150 | OUTCD 86 |
| 135 | CONTINUE | OUTCD 87 |
| | IF(IYY.GT.1IY) GO TO 140 | OUTCD 88 |
| | RETURN | OUTCD 89 |
| 140 | CONTINUE | OUTCD 90 |
| | DO 145 I=1,IIY | OUTCD 91 |
| | WRITE (IUNIT,5) (IMOD1(L,I,IY),L=1,14) | OUTCD 92 |
| 145 | CONTINUE | OUTCD 93 |
| | IAC(7,IY) = 1 | OUTCD 94 |
| | RETURN | OUTCD 95 |
| C | | OUTCD 96 |
| C | MOVE WORK AREA INTO BUFFER AND | OUTCD 97 |
| C | BLANK REMAINING BUFFER | OUTCD 98 |
| C | | OUTCD 99 |
| 150 | CONTINUE | OUTCD100 |
| | IAC(4,IY) = 1 | OUTCD101 |
| | K = (2*IIW)-IXX | OUTCD102 |
| | K1= 2*IIW | OUTCD103 |
| | DO 155 I=K,K1 | OUTCD104 |
| | IWK1(I,IY) = 6H | OUTCD105 |
| 155 | CONTINUE | OUTCD106 |
| | CALL A1TA6 (IWK1(1,IY),IMOD1(1,IYY,IY)) | OUTCD107 |
| | IS = IYY+1 | OUTCD108 |
| | DO 170 I=IS,IIY | OUTCD109 |
| | DO 165 J=1,14 | OUTCD110 |
| | IMOD1(J,I,IY) = 6H | OUTCD111 |
| 165 | CONTINUE | OUTCD112 |
| 170 | CONTINUE | OUTCD113 |
| | DO 180 I=1,55 | OUTCD114 |
| | WRITE (IUNIT,5) (IMOD1(L,I,IY),L=1,14) | OUTCD115 |
| 180 | CONTINUE | OUTCD116 |
| | RETURN | OUTCD117 |
| | | OUTCD118 |

| | | |
|---|---|----------|
| C | | OUTCD119 |
| C | ERROR EXIT | OUTCD120 |
| C | | OUTCD121 |
| | 500 CONTINUE | OUTCD122 |
| | WRITE (FERR,1000) IUNIT,IX | OUTCD123 |
| | 1000 FORMAT(27H OUTCD - MODE ERROR -- UNIT,I3,5H MODE,I3) | OUTCD124 |
| | ERFLAG = 1 | OUTCD125 |
| | CALL TERM | OUTCD126 |
| | 510 CONTINUE | OUTCD127 |
| | WRITE (FERR,1010) IUNIT | OUTCD128 |
| | 1010 FORMAT(28H OUTCD - END OF FILE ON UNIT,I3) | OUTCD129 |
| | ERFLAG = 1 | OUTCD130 |
| | CALL TERM | OUTCD131 |
| | RETURN | OUTCD132 |
| | END | OUTCD133 |
| | *ELT,I DORMAN.PACCON | PACCON 3 |
| | SUBROUTINE PACCON (CHAR,OCHAR,NCHAR) | PACCON 5 |
| C | | PACCON 6 |
| C | PACK CARDS WITH = AND ; | PACCON 7 |
| C | PROGRAMMER - S. WRAY | PACCON 8 |
| C | | PACCON 9 |
| | INTEGER CHAR(80),OCHAR(80),PKTAB(9,2) | PACCON10 |
| | DO 5000 I=1,80 | PACCON11 |
| | IF(CHAR(I).EQ.1H) CHAR(I)=1H- | PACCON12 |
| | IF(CHAR(I).EQ.1H;) CHAR(I) = 1H | PACCON13 |
| | 5000 CONTINUE | PACCON14 |
| | DO 5 I = 1,80 | PACCON15 |
| | J = 81 - I | PACCON16 |
| | IF(CHAR(J).NE.1H) GO TO 10 | PACCON17 |
| | 5 CONTINUE | PACCON18 |
| | NCHAR = 1 | PACCON19 |
| | 9 OCHAR(NCHAR)=1H; | PACCON20 |
| | RETURN | PACCON21 |
| | 10 NCHAR = 81 - I | PACCON22 |
| | PKTAB(1,1) = 1 | PACCON23 |
| | NPK = 0 | PACCON24 |

```

DO 11 I=1,8
DO 12 J=1,10
L = 10*I + 1 - J
IF(L.GE.NCHAR) GO TO 14
IF(CHAR(L).NE.1H ) GO TO 13
12 CONTINUE
L=L-1
13 IF(J.EQ.1) GO TO 11
CHAR(L+1) = 1H=
NPK=NPK+1
PKTAB(NPK,2) = L+1
PKTAB(NPK+1,1) = 10*I +1
11 CONTINUE
GO TO 15
14 NPK = NPK+1
PKTAB(NPK,2) = NCHAR+1
CHAR(NCHAR+1) = 1H;
15 NCHAR = 0
DO 16 L=1,NPK
L1=PKTAB(L,1)
L2=PKTAB(L,2)
DO 17 K=L1,L2
NCHAR=NCHAR+1
OCHAR(NCHAR)=CHAR(K)
17 CONTINUE
16 CONTINUE
RETURN
END
*ELT,I DORMAN.REPL
SUBROUTINE REPL
C
C          CONTROLS REPLACEMENT OF DECKS
C
C          PROGRAMMER - S. WRAY
C
COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14)

```

```

PACCON25
PACCON26
PACCON27
PACCON28
PACCON29
PACCON30
PACCON31
PACCON32
PACCON33
PACCON34
PACCON35
PACCON36
PACCON37
PACCON38
PACCON39
PACCON40
PACCON41
PACCON42
PACCON43
PACCON44
PACCON45
PACCON46
PACCON47
PACCON48
PACCON49
PACCON50
PACCON51
PACCON52
REPL 3
REPL 5
REPL 6
REPL 7
REPL 8
REPL 9
REPL 10
MISC 2

```

| | | |
|---|-------|----|
| INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC | 3 |
| EQUIVALENCE (PRTFIL,FERR) | MISC | 4 |
| COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2 | FILES | 2 |
| INTEGER BASIC,MTAPE,FINAL,S1,S2 | FILES | 3 |
| COMMON /NAMES/ IVER,DNAME(2),MODNAM(2),BNAME(2) | NAMES | 2 |
| INTEGER DNAME,MODNAM,BNAME | NAMES | 3 |
| DIMENSION IU(3),IR(3) | REPL | 14 |
| DATA IU,IR/0,0,0,2,2,2/ | REPL | 15 |
| WRITE (PRTFIL,5) | REPL | 16 |
| 5 FORMAT(30H ENTER NAME OF DECK TO BE USED/9H READY -) | REPL | 17 |
| CALL INCD (KARD,5) | REPL | 18 |
| M1 = 13 | REPL | 19 |
| I = 0 | REPL | 20 |
| M2 = 14 | REPL | 21 |
| IF((KARD(1).EQ.MODNAM(1)).AND.(KARD(2).EQ.MODNAM(2))) I = M1 | REPL | 22 |
| IF((KARD(1).EQ.BNAME(1)).AND.(KARD(2).EQ.BNAME(2))) I = M2 | REPL | 23 |
| IF(I.NE.0) GO TO 20 | REPL | 24 |
| WRITE (PRTFIL,15) | REPL | 25 |
| 15 FORMAT(24H ORIGINAL DECK NOT FOUND) | REPL | 26 |
| ERFLAG = 1 | REPL | 27 |
| RETURN | REPL | 28 |
| 20 CONTINUE | REPL | 29 |
| WRITE (PRTFIL,10) | REPL | 30 |
| 10 FORMAT(34H ENTER NAME OF DECK TO BE REPLACED/9H READY -) | REPL | 31 |
| CALL INCD (ACTION,5) | REPL | 32 |
| IF((KARD(1).EQ.ACTION(1)).AND.(KARD(2).EQ.ACTION(2))) GO TO 100 | REPL | 33 |
| WRITE (PRTFIL,25) | REPL | 34 |
| 25 FORMAT(24H DECK NAMES DO NOT MATCH/38H REQUEST REJECTED - USE DELEREPL | REPL | 35 |
| *TE AND ADD) | REPL | 36 |
| ERFLAG = 1 | REPL | 37 |
| RETURN | REPL | 38 |
| 100 CONTINUE | REPL | 39 |
| S1 = BASIC + 1 | REPL | 40 |
| IF(S1.GT.3) S1 = 2 | REPL | 41 |
| IU(1) = BASIC | REPL | 42 |
| IU(2) = S1 | REPL | 43 |

| | |
|---|----------|
| IU(3) = I | REPL 44 |
| REWIND I | REPL 45 |
| REWIND S1 | REPL 46 |
| CALL INTBUF(IU,IR) | REPL 47 |
| CALL REPLAC(BASIC,I,S1) | REPL 48 |
| IF(ERFLAG.NE.C) RETURN | REPL 49 |
| BASIC = S1 | REPL 50 |
| RETURN | REPL 51 |
| END | REPL 52 |
| *ELT,I DORMAN.REPLAC | REPLAC 3 |
| SUBROUTINE REPLAC(IN1,IN2,OUT) | REPLAC 5 |
| C SUBSTITUTE DECK WHICH IS NEXT ON FILE IN2 FOR THE DECK OF THE SAME | REPLAC 6 |
| C NAME WHICH IS ON THE BASIC DATA FILE IN1. COPY THE REVISED BASIC DATA | REPLAC 7 |
| C FILE ONTO FILE OUT. | REPLAC 8 |
| C PROGRAMMER - B. GOLD | REPLAC 9 |
| C | REPLAC10 |
| COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC 2 |
| INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC 3 |
| EQUIVALENCE (PRTFIL,FERR) | MISC 4 |
| INTEGER OUT | REPLAC12 |
| C | REPLAC13 |
| INTEGER TYPE, FLAG, BASIK, MOD, TEMP(14) | REPLAC14 |
| COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST 2 |
| INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST 3 |
| INTEGER VERS | REPLAC16 |
| DATA VERS/6HVERSIO/ | REPLAC17 |
| DATA BASIK, MOD / 5HBASIC, 3HMOD / | REPLAC18 |
| C | REPLAC19 |
| C INITIALIZE FILES. READ FIRST CARD OF NEW DECK AND VERSION CARD FROM | REPLAC20 |
| C DATA FILE. | REPLAC21 |
| C | REPLAC22 |
| FLAG = 1 | REPLAC23 |
| REWIND IN1 | REPLAC24 |
| REWIND OUT | REPLAC25 |
| ERFLAG = 0 | REPLAC26 |
| CALL INCD(KARD,IN2) | REPLAC27 |

```

KOUNT2 = 1
IF (ERFLAG.NE.0) GO TO 100
TYPE = BASIK
IF (KARD(4).EQ.USES) TYPE = MOD
IF (KARD(1).NE.DDECK) GO TO 110
NAME1 = KARD(2)
NAME2 = KARD(3)
CALL EXTRAN (KARD(7),KARD(9))
WRITE (PRTFIL,10) KARD(2), KARD(3)
10 FORMAT(20HOREPLACE DECK NAMED ,2A6,13H ON DATA FILE/)
CALL OUTCU (KARD,PRTFIL)
IF ((KARD(2).EQ.KARD(5)).AND.(KARD(3).EQ.KARD(6))) GO TO 170
KOUNT1= 0

```

C

C COPY TABLE OF CONTENTS. CHECK TO SEE THAT REPLACEMENT DECK AND ANY
C DECK IT USES ARE LISTED AMONG CONTENTS.

C

```

FLAG = 2
INDEX = 0
NEED = 0
LASTM = 0
NTC = 0
IUSE = 0
30 CALL INCD(TEMP,1N1)
IF (ERFLAG.NE.0) GO TO 120
KOUNT1 = KOUNT1+1
IF (TEMP(1).EQ.END) GO TO 40
IF(TEMP(5).EQ.VERS) GO TO 30
IF (TEMP(1).NE.DDECK) GO TO 150
NTC = NTC+1
IF (TEMP(4).EQ.USES) LASTM = NTC
IF ((KARD(2).EQ.TEMP(5)).AND.(KARD(3).EQ.TEMP(6))) IUSE = NTC
IF ((TEMP(2).EQ.KARD(5)).AND.(TEMP(3).EQ.KARD(6))) NEED = NTC
IF ((TEMP(2).EQ.KARD(2)).AND.(TEMP(3).EQ.KARD(3))) INDEX= NTC
GO TO 30

```

C

```

REPLAC28
REPLAC29
REPLAC30
REPLAC31
REPLAC32
REPLAC33
REPLAC34
REPLAC35
REPLAC36
REPLAC37
REPLAC38
REPLAC39
REPLAC40
REPLAC41
REPLAC42
REPLAC43
REPLAC44
REPLAC45
REPLAC46
REPLAC47
REPLAC48
REPLAC49
REPLAC50
REPLAC51
REPLAC52
REPLAC53
REPLAC54
REPLAC55
REPLAC56
REPLAC57
REPLAC58
REPLAC59
REPLAC60
REPLAC61
REPLAC62
REPLAC63

```

| | | |
|---|---|----------|
| C | DECIDE WHERE TO INSERT NEW \$DECK CARD IN TABLE OF CONTENTS | REPLAC64 |
| C | | REPLAC65 |
| | 40 LOC = INDEX | REPLAC66 |
| | IF (INDEX.EQ.0) GO TO 130 | REPLAC67 |
| | IF ((NEED.EQ.0).AND.(TYPE.EQ.MOD)) GO TO 140 | REPLAC68 |
| | IF ((NEED.GT.0).AND.(NEED.LT.IUSE)) GO TO 160 | REPLAC69 |
| | INDEX = IUSE+1 | REPLAC70 |
| | IF (TYPE.EQ.BASIK) INDEX = NTC | REPLAC71 |
| C | | REPLAC72 |
| C | READ TABLE OF CONTENTS AGAIN AND COPY ONTO OUT FILE WITH NEW \$DECK | REPLAC73 |
| C | CARD INSERT AND OLD ONE DELETED | REPLAC74 |
| C | | REPLAC75 |
| | REWIND IN1 | REPLAC76 |
| | KOUNT1 = 0 | REPLAC77 |
| | KOUNT3 = 0 | REPLAC78 |
| | 45 CALL INCD(TEMP,IN1) | REPLAC79 |
| | IF (ERFLAG.NE.0) GO TO 120 | REPLAC80 |
| | KOUNT1 = KOUNT1+1 | REPLAC81 |
| | IF ((TEMP(2).EQ.NAME1).AND.(TEMP(3).EQ.NAME2)) GO TO 45 | REPLAC82 |
| | IF (INDEX.NE.KOUNT3) GO TO 47 | REPLAC83 |
| | CALL OUTCD(KARD,OUT) | REPLAC84 |
| | KOUNT3 = KOUNT3+1 | REPLAC85 |
| | 47 CALL OUTCD(TEMP,OUT) | REPLAC86 |
| | KOUNT3 = KOUNT3+1 | REPLAC87 |
| | IF ((TEMP(1).NE.END).OR.(TEMP(2).NE.TABLE)) GO TO 45 | REPLAC88 |
| C | | REPLAC89 |
| C | COPY ALL DECKS FROM IN1 TO OUT EXCEPT DECK TO BE REPLACED. | REPLAC90 |
| C | INSERT NEW DECK WHEN DECK COUNT NDECK = NUMBER | REPLAC91 |
| C | | REPLAC92 |
| | FLAG = 3 | REPLAC93 |
| | NDECK = 0 | REPLAC94 |
| | IF ((TYPE.EQ.MOD) .AND.(LOC.LE.LASTM)) NUMBER = LASTM-INDEX+1 | REPLAC95 |
| | IF ((TYPE.EQ.MOD) .AND.(LOC.GT.LASTM)) NUMBER = LASTM-INDEX+2 | REPLAC96 |
| | IF ((TYPE.EQ.BASIK).AND.(LOC.LE.LASTM)) NUMBER = LASTM | REPLAC97 |
| | IF ((TYPE.EQ.BASIK).AND.(LOC.GT.LASTM)) NUMBER = LASTM+1 | REPLAC98 |
| | 50 CALL INCD(TEMP,IN1) | REPLAC99 |

```
IF (ERFLAG.NE.0) GO TO 120
KOUNT1 = KOUNT1 + 1
IF (TEMP(1).NE.DDECK) GO TO 83
ISKIP = 0
IF ((TEMP(2).EQ.NAME1).AND.(TEMP(3).EQ.NAME2)) ISKIP = 1
IF (ISKIP.EQ.1) GO TO 80
NDECK = NDECK+1
IF (NDECK.NE.NUMBER) GO TO 80
C
C TRANSFER NEW DECK ONTO OUT FILE
C
55 FLAG = 4
CALL OUTCD(KARD,OUT)
60 CALL INCD(KARD,IN2)
IF (ERFLAG.NE.0) GO TO 100
KOUNT2 = KOUNT2+1
CALL OUTCD(KARD,OUT)
IF (KARD(1).NE.END) GO TO 60
WRITE (PRTFIL,70) KOUNT2
70 FORMAT(/I10,6H CARDS)
FLAG = 5
C
C DO NOT COPY THIS CARD ONTO OUT FILE IF IT IS PART OF THE OLD DECK TO
C BE REPLACED (IF ISKIP=1)
C
80 IF (ISKIP.EQ.0) CALL OUTCD(TEMP,OUT)
IF ((TEMP(1).NE.END).OR.(TEMP(2).NE.FILE)) GO TO 50
IF (ISKIP.EQ.1) CALL OUTCD(TEMP,OUT)
RETURN
C
C ERROR STOPS
C
100 WRITE (FERR,105)
105 FORMAT(23HOERROR READING FILE IN2)
GO TO 200
110 WRITE (FERR,115)
```

REPLA100
REPLA101
REPLA102
REPLA103
REPLA104
REPLA105
REPLA106
REPLA107
REPLA108
REPLA109
REPLA110
REPLA111
REPLA112
REPLA113
REPLA114
REPLA115
REPLA116
REPLA117
REPLA118
REPLA119
REPLA120
REPLA121
REPLA122
REPLA123
REPLA124
REPLA125
REPLA126
REPLA127
REPLA128
REPLA129
REPLA130
REPLA131
REPLA132
REPLA133
REPLA134
REPLA135

| | | |
|-----|---|----------|
| 115 | FORMAT(37H0FIRST CARD ON FILE IN2 IS NOT \$DECK) | REPLA136 |
| | GO TO 200 | REPLA137 |
| 120 | WRITE (FERR,125) | REPLA138 |
| 125 | FORMAT(23H0ERROR READING FILE IN1) | REPLA139 |
| | GO TO 200 | REPLA140 |
| 130 | WRITE (FERR,135) KARD(2), KARD(3) | REPLA141 |
| 135 | FORMAT(22H0DECK TO BE REPLACED (,2A6,36H) IS NOT LISTED IN TABLE OF | REPLA142 |
| | *F CONTENTS) | REPLA143 |
| | GO TO 200 | REPLA144 |
| 140 | WRITE (FERR,145) (KARD(I), I=1,6) | REPLA145 |
| 145 | FORMAT(52H0DECK TO BE ADDED USES DECK NOT IN TABLE OF CONTENTS/ | REPLA146 |
| | * 5X,6A6) | REPLA147 |
| | GO TO 200 | REPLA148 |
| 150 | WRITE (FERR,155) | REPLA149 |
| 155 | FORMAT(50H0ILLEGAL CARD WITHIN TABLE OF CONTENTS ON FILE IN1) | REPLA150 |
| | GO TO 200 | REPLA151 |
| 160 | WRITE (FERR,165) (NAME1,NAME2, J=1,3) | REPLA152 |
| 165 | FORMAT(63H0ERROR IN DECK POSITIONS DISCOVERED DURING REPLACEMENT OF | REPLA153 |
| | 1F DECK 2A6/17H DECK WHICH USES 2A6,22H FOLLOWS DECK USED BY 2A6) | REPLA154 |
| | GO TO 200 | REPLA155 |
| 170 | WRITE (FERR,175) | REPLA156 |
| 175 | FORMAT(17H0DECK USES ITSELF) | REPLA157 |
| C | | REPLA158 |
| | 200 ERFLAG = 1 | REPLA159 |
| C | | REPLA160 |
| | C ADVANCE FILE IN2 TO END OF CURRENT DECK | REPLA161 |
| C | | REPLA162 |
| | ERFLAG = 0 | REPLA163 |
| 210 | IF (KARD(1).EQ.END) GO TO 220 | REPLA164 |
| | CALL INCD(KARD,IN2) | REPLA165 |
| | KOUNT2 = KOUNT2 + 1 | REPLA166 |
| | IF (ERFLAG.EQ.0) GO TO 210 | REPLA167 |
| | WRITE (FERR,105) | REPLA168 |
| 220 | ERFLAG = 1 | REPLA169 |
| | RETURN | REPLA170 |
| | END | REPLA171 |

| | |
|---|-----------|
| #ELT,I DORMAN,RESTOR | RESTORM3 |
| SUBROUTINE RESTOR (BUF,NC,NL,FILE,NCNOW) | RESTORM5 |
| C | RESTORM6 |
| REPOSITION CARD FILE, PLACES THE DESIRED CARD | RESTORM7 |
| FIRST IN BUFFER AND FILLS BUFFER | RESTORM8 |
| C | RESTORM9 |
| UPON ENTRY | RESTOR10 |
| BUF BUFFER TO BE FILLED | RESTOR11 |
| NC NR OF CARD TO BE AT POSITION 1 | RESTOR12 |
| NL NR OF CARDS IN BUFFER | RESTOR13 |
| FILE TAPE NUMBER | RESTOR14 |
| NCNOW CARD NUMBER CURRENTLY IN POSITION 1 | RESTOR15 |
| C | RESTOR16 |
| INTEGER MAX | RESTOR17 |
| INTEGER END | RESTOR18 |
| INTEGER BUF,FILE | RESTOR19 |
| DIMENSION BUF(14,20) | RESTOR20 |
| COMMON /WORK/IIUNT(3),IIRW(3),IUTBL(3,19),IACT(8,3),NFILES | WORK 2 |
| COMMON /BFRS/ ISYN(2,20),ISYN1,IT1,IT2,IT3,FULL,CN1,CN2,LIN1,LIN2 | BFRS1 2 |
| *,NWBUF,NB1,NB2,ICN1,ICN2 | BFRS1 3 |
| COMMON /BFRS/ BUF1(14,50),BUF2(14,50),AAA(700) | BFRS1 4 |
| INTEGER CN1,CN2,BUF1,BUF2 | BFRS1 5 |
| LOGICAL FULL | BFRS1 6 |
| EQUIVALENCE (NWBUF,MAX) | RESTOR22 |
| IF (NC.EQ.NCNOW) RETURN | RESTOR23 |
| NNC = NC | RESTOR24 |
| NNOW = NCNOW | RESTOR25 |
| END = NL | RESTOR26 |
| IF (NNC.GT.NNOW) GO TO 100 | RESTOR27 |
| C | RESTOR28 |
| CARD HAS PREVIOUSLY BEEN READ | RESTOR29 |
| C | RESTOR30 |
| REWIND FILE | RESTOR31 |
| CALL FNDBUF(FILE,IX,IY) | GWT3 2 |
| IACT(4,IY) = 0 | GWT3 3 |
| NNC = 1 | RESTOR32 |

```

      NNOW= 1
10  CONTINUE
      CALL FILBUF(1,END,MAX,BUF,FILE)
      NL = END
      IF (NC.EQ.NNC) RETURN
100 CONTINUE
      IF (NC.LE.NNOW+END-1) GO TO 120
      NNC = NNOW+END
      NNOW= NNC
      GO TO 10
120 CONTINUE
      N = NC-NNOW
C      MOVE CARDS UP, PLACING NC INTO POSITION 1
      K = END - N
      DO 140 I=1,K
      L= N+I
      DO 130 J=1,14
      BUF(J,I) = BUF(J,L)
130 CONTINUE
140 CONTINUE
      NL = K
      IF (BUF(1,K).EQ.6H$END 0) RETURN
      K = K+1
      CALL FILBUF(K,END,MAX,BUF,FILE)
      NL = END
      RETURN
      END
#ELT,I DORMAN.SAVER
      SUBROUTINE SAVER
C
C      SAVE NECESSARY FILES/DECKS
C
C      PROGRAMMER - S. WRAY
C
      DIMENSION IU(3),IR(3)
      COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14)

```

```

RESTOR33
GWT13 5
GWT13 6
GWT13 7
RESTOR36
RESTOR37
RESTOR38
RESTOR39
RESTOR40
RESTOR41
RESTOR42
RESTOR43
RESTOR44
GWT13 8
RESTOR46
RESTOR47
RESTOR48
RESTOR49
RESTOR50
RESTOR51
RESTOR52
RESTOR53
RESTOR54
GWT13 9
GWT13 10
RESTOR58
RESTOR59
SAVER 3
SAVER 5
SAVER 6
SAVER 7
SAVER 8
SAVER 9
SAVER 10
SAVER 11
MISC 2

```

```

INTEGER ERFLAG,FERR,ACTION,PRTFIL
EQUIVALENCE (PRTFIL,FERR)
COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2
INTEGER BASIC,MTAPE,FINAL,S1,S2
COMMON /NAMES/ IVER,DNAME(2),MODNAM(2),BNAME(2)
INTEGER DNAME,MODNAM,BNAME
COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH
INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH
IU(1) = 5
IR(1) = 2
IR(2) = 2
IR(3) = 2
WRITE (PRTFIL,5)
5 FORMAT(49H ENTER SAVE OPTION (DATA BASE, DORCA OR MOD DECK)
*/9H READY - )
CALL INCD(KARD,5)
IF(KARD(1).EQ.6HDATA B) GO TO 10
IF(KARD(1).EQ.6HDORCA ) GO TO 20
IF(KARD(1).EQ.6HMOD DE ) GO TO 30
WRITE (PRTFIL,16)
16 FORMAT(40H COMMAND NOT UNDERSTOOD - PLEASE RETRY )
RETURN
10 CONTINUE
IF(BASIC.NE.1) GO TO 6
WRITE (PRTFIL,7)
7 FORMAT(31H FILE DOES NOT NEED TO BE SAVED)
RETURN
6 CONTINUE
WRITE (PRTFIL,8)
8 FORMAT(25H ENTER VERSION IDENTIFIER/9H READY - )
CALL INCD (ACTION,5)
IU(1) = 5
IU(2) = BASIC
IU(3) = 4
CALL INT3UF(IU,IR)
IVER = ACTION(1)

```

```

MISC 3
MISC 4
FILES 2
FILES 3
NAMES 2
NAMES 3
REST 2
REST 3
SAVER 16
SAVER 17
SAVER 18
SAVER 19
SAVER 20
SAVER 21
SAVER 22
SAVER 23
SAVER 24
SAVER 25
SAVER 26
SAVER 27
SAVER 28
SAVER 29
SAVER 30
SAVER 31
SAVER 32
SAVER 33
SAVER 34
SAVER 35
SAVER 36
SAVER 37
SAVER 38
SAVER 39
SAVER 40
SAVER 41
SAVER 42
SAVER 43

```

| | |
|---|----------|
| CALL LABLER(IVER,BASIC,4) | SAVER 44 |
| RETURN | SAVER 45 |
| 20 CONTINUE | SAVER 46 |
| FINAL = 14 | SAVER 47 |
| WRITE (PRTFIL,21) | SAVER 48 |
| 21 FORMAT(46H ENTER NAME OF DECK TO BE USED FOR LORCA INPUT/9H READY *-) | SAVER 49 |
| CALL INCD (KARD,5) | SAVER 50 |
| IF((KARD(1).EQ.BNAME(1)).AND.(KARD(2).EQ.BNAME(2))) GO TO 40 | SAVER 51 |
| IF(BNAME(1).EQ.0) GO TO 38 | SAVER 52 |
| 37 WRITE (PRTFIL,39) | SAVER 53 |
| 39 FORMAT(18H DECK IS NOT BASIC/29H DESTRUCT PERMISSION REQUIRED/ *16H ENTER YES OR NO/9H READY -) | SAVER 54 |
| CALL INCD(ACTION,5) | SAVER 55 |
| IF(ACTION(1).EQ.2HNO) RETURN | SAVER 56 |
| IF(ACTION(1).NE.3HYES) GO TO 37 | SAVER 57 |
| 38 CONTINUE | SAVER 58 |
| ACTION(1) = KARD(1) | SAVER 59 |
| ACTION(2) = KARD(2) | SAVER 60 |
| CALL USE (ACTION) | SAVER 61 |
| 40 CONTINUE | SAVER 62 |
| REWIND FINAL | SAVER 63 |
| IU(1) = 5 | SAVER 64 |
| IU(2) = FINAL | SAVER 65 |
| IU(3) = 12 | SAVER 66 |
| CALL INTBUF(IU,IR) | SAVER 67 |
| REWIND 12 | SAVER 68 |
| REWIND FINAL | SAVER 69 |
| 22 CALL INCD (KARD,FINAL) | SAVER 70 |
| IF(ERFLAG.NE.0) RETURN | SAVER 71 |
| IF(KARD(1).EQ.DDECK) GO TO 22 | SAVER 72 |
| IF(KARD(1).EQ.END) GO TO 23 | SAVER 73 |
| CALL OUTCD(KARD,12) | SAVER 74 |
| GO TO 22 | SAVER 75 |
| 23 CONTINUE | SAVER 76 |
| DNAME (1) = ACTION(1) | SAVER 77 |
| | SAVER 78 |
| | SAVER 79 |

| | |
|---|----------|
| DNAME (2) = ACTION(2) | SAVER 80 |
| REWIND 12 | SAVER 81 |
| RETURN | SAVER 82 |
| 30 CONTINUE | SAVER 83 |
| WRITE (PRTFIL,51) | SAVER 84 |
| 51 FORMAT(43H ENTER NAME OF DECK TO BE SAVED AS MOD DECK/9H READY -) | SAVER 85 |
| CALL INCD (ACTION,5) | SAVER 86 |
| IF((ACTION(1).EQ.MODNAM(1)).AND.(ACTION(2).EQ.MODNAM(2))) GO TO 60 | SAVER 87 |
| WRITE (PRTFIL,52) | SAVER 88 |
| 52 FORMAT(24H DECK NAMES DO NOT MATCH/23H MOD DECK NOT AVAILABLE/ *15H REQUEST DENIED) | SAVER 89 |
| ERFLAG = 1 | SAVER 90 |
| RETURN | SAVER 91 |
| 60 CONTINUE | SAVER 92 |
| IU(1) = 13 | SAVER 93 |
| IU(2) = 24 | SAVER 94 |
| CALL INTBUF(IU,IR) | SAVER 95 |
| M1 = 24 | SAVER 96 |
| REWIND M1 | SAVER 97 |
| M2 = 13 | SAVER 98 |
| REWIND M2 | SAVER 99 |
| 65 CALL INCD (KARD,M2) | SAVER100 |
| IF(ERFLAG.NE.0) RETURN | SAVER101 |
| CALL OUTCD (KARD,M1) | SAVER102 |
| IF(KARD(1).NE.END) GO TO 65 | SAVER103 |
| RETURN | SAVER104 |
| END | SAVER105 |
| *ELT,I DORMAN.SYNCF | SAVER106 |
| SUBROUTINE SYNCF (BUF1,NC1,NL1,FILE1,NSYN1, | SYNCFM3 |
| * ISYN1, | SYNCFM5 |
| * BUF2,NC2,NL2,FILE2,NSYN2) | SYNCFM6 |
| COMMON /BUFFER/ MAX | SYNCFM7 |
| INTEGER BUF1,BUF2,FILE1,FILE2 | SYNCFM8 |
| COMMON /BFRS/ | SYNCFM9 |
| * ISYN(2,20) | SYNCF10 |
| DIMENSION BUF1(14,20),BUF2(14,20) | SYNCF11 |
| | SYNCF12 |

| | | |
|----|---|----------|
| | INTEGER END | SYNCBF13 |
| C | | SYNCBF14 |
| C | NCC11 - FIRST CARD IN BUFFER 1 | SYNCBF15 |
| C | NCC21 - FIRST CARD IN BUFFER 2 | SYNCBF16 |
| C | | SYNCBF17 |
| C | NCC1 - NCC2 CURRENT CARD BEING CONSIDERED | SYNCBF18 |
| C | NLL1 - NLL2 NR OF CARDS IN BUFFER | SYNCBF19 |
| C | NCB1 - NCB2 PRESENT INDEX IN BUFFER | SYNCBF20 |
| C | ISYN1 - INDEX TO THE SYNC CARDS | SYNCBF21 |
| C | | SYNCBF22 |
| | NCC11= NC1 | SYNCBF23 |
| | NCC21= NC2 | SYNCBF24 |
| | NFLG1 = 0 | GWT13 11 |
| | NFLG2 = 0 | GWT13 12 |
| | NCC1 = 1 | GWT13 13 |
| | NCC2 = 1 | SYNCBF25 |
| | NLL1 = NL1 | SYNCBF27 |
| | NLL2 = NL2 | SYNCBF28 |
| | NCB1 = 1 | SYNCBF29 |
| | NCB2 = 1 | SYNCBF30 |
| | NSYN1 = ISYN(1,ISYN1) -NC1 + 1 | SYNCBF31 |
| | NSYN2 = ISYN(2,ISYN1) -NC2 + 1 | SYNCBF32 |
| | GO TO 70 | GWT13 14 |
| 10 | CONTINUE | GWT13 15 |
| | IF(BUF2(1,NLL2).NE.6H\$END 0) GO TO 30 | GWT13 16 |
| 15 | NSYNC1 = NCC1 | GWT13 17 |
| | NSYNC2 = NCC2 | GWT13 18 |
| | CALL RESTOR(BUF1,NC1,NLL1,FILE1,NCC11) | GWT13 19 |
| | CALL RESTOR(BUF2,NC2,NLL2,FILE2,NCC21) | GWT13 20 |
| | RETURN | GWT13 21 |
| 30 | IF(IFLG2.EQ.1) GO TO 15 | GWT13 22 |
| | CALL FILBUF(1,END,MAX,BUF2,FILE2) | GWT13 23 |
| | NCC21 = NCC21 + NLL2 | GWT13 24 |
| | NCC2 = NCC21 - NC2 + 1 | GWT13 25 |
| | NLL2 = END | GWT13 26 |
| | NCB2 = 1 | GWT13 27 |

| | | |
|----|---|----------|
| | CALL RESTOR(BUF1,NC1,NLL1,FILE1,NCC11) | GWT13 28 |
| | NCC11 = NC1 | GWT13 29 |
| | NCB1 = 1 | GWT13 30 |
| | NCC1 = 1 | GWT13 31 |
| | NLL1 = NLL1 | GWT13 32 |
| | IFLG1 = 0 | GWT13 33 |
| | GO TO 40 | GWT13 34 |
| 40 | CONTINUE | GWT13 35 |
| | DO 45 I = 1,14 | GWT13 36 |
| | IF(BUF1(I,NCB1).NE.BUF2(I,NCB2)) GO TO 50 | GWT13 37 |
| 45 | CONTINUE | GWT13 38 |
| | GO TO 15 | GWT13 39 |
| 50 | NCB2 = NCB2 + 1 | GWT13 40 |
| | NCC2 = NCC2 + 1 | GWT13 41 |
| | IF(NSYN2.NE.NCC2) GO TO 55 | GWT13 42 |
| | IFLG2 = 1 | GWT13 43 |
| | GO TO 60 | GWT13 44 |
| 55 | IF(NCB2.LT.NLL2) GO TO 40 | GWT13 45 |
| 60 | CONTINUE | GWT13 46 |
| | NCB1 = NCB1 + 1 | GWT13 47 |
| | NCC1 = NCC1 + 1 | GWT13 48 |
| | IF(NSYN1.NE.NCC1) GO TO 65 | GWT13 49 |
| | IFLG1 = 1 | GWT13 50 |
| | GO TO 10 | GWT13 51 |
| 65 | IF(NCB1.LT.NLL1) GO TO 85 | GWT13 52 |
| 70 | IF(BUF1(1,NLL1).NE.6H\$END 0) GO TO 80 | GWT13 53 |
| | GO TO 10 | GWT13 54 |
| 80 | CONTINUE | GWT13 55 |
| | CALL FILBUF(1,END,MAX,BUF1,FILE1) | GWT13 56 |
| | NCB1 = 1 | GWT13 57 |
| | NCC11 = NCC11 + NLL1 | GWT13 58 |
| | NCC1 = NCC11 - NC1 + 1 | GWT13 59 |
| | NLL1 = END | GWT13 60 |
| 85 | CALL RESTOR(BUF2,NCC21,NLL2,FILE2,NCC21) | GWT13 61 |
| | NCB2 = 1 | GWT13 62 |
| | NCC2 = NCC21 - NC2 + 1 | GWT13 63 |

```

        IFLG2 = 0
        GO TO 40
        END
*ELT,I  DORMAN.SYNCDS
        SUBROUTINE SYNCDS (FSYN)
        COMMON /BFRS/ ISYN(2,20)
        INTEGER FSYN,KARD(16),ITMP(16)
C
C      READ IN SYNC CARDS
C
        DIMENSION IUNT(3),IRW(3)
        IUNT(1)=FSYN
        IUNT(2)= 0
        IUNT(3)= 0
        IRW(1) = 0
        IRW(2) = 0
        IRW(3) = 1
        DO 1 I=1,20
        DO 4 J=1,2
        ISYN(J,I) = 100000
+      CONTINUE
1      CONTINUE
        CALL INTBUF(IUNT,IRW)
        WRITE (6,100)
100  FORMAT(17H INPUT SYNC CARDS)
        DO 2 I=1,20
105  CONTINUE
        WRITE (6,110)
110  FORMAT(9H READY - )
        CALL INCD (KARD,FSYN)
        IF(KARD(1).EQ.4HDONE) GO TO 3
        ENCODE (8+,62,ITMP) (KARD(J),J=1,14)
        62  FORMAT(1+A6)
        DECODE(80,6+,ITMP) KARD
        64  FORMAT(8(A6,A4))
        CALL VALUE (KARD,A,IERR)

```

```

GWT13 64
GWT13 65
SYNCB135
SYNCDS 3
SYNCDS 5
SYNCDS 6
SYNCDS 7
SYNCDS 8
SYNCDS 9
SYNCDS10
SYNCDS11
SYNCDS12
SYNCDS13
SYNCDS14
SYNCDS15
SYNCDS16
SYNCDS17
SYNCDS18
SYNCDS19
SYNCDS20
SYNCDS21
SYNCDS22
SYNCDS23
SYNCDS24
SYNCDS25
SYNCDS26
SYNCDS27
SYNCDS28
SYNCDS29
SYNCDS30
SYNCDS31
SYNCDS32
SYNCDS33
SYNCDS34
SYNCDS35
SYNCDS36

```

| | |
|---|----------|
| IF(IERR.GT.0) GO TO 520 | SYNCDS37 |
| ISYN(1,I) = A | SYNCDS38 |
| CALL VALUE (KARD(3),A,IERR) | SYNCDS39 |
| IF(IERR.GT.0) GO TO 520 | SYNCDS40 |
| ISYN(2,I) = A | SYNCDS41 |
| GO TO 17 | SYNCDS42 |
| 520 CONTINUE | SYNCDS43 |
| WRITE (6,530) | SYNCDS44 |
| 530 FORMAT(37H BAD NUMERIC ENTRY - PLEASE REED CARD) | SYNCDS45 |
| GO TO 105 | SYNCDS46 |
| 17 CONTINUE | SYNCDS47 |
| 2 CONTINUE | SYNCDS48 |
| 3 CONTINUE | SYNCDS49 |
| RETURN | SYNCDS50 |
| END | SYNCDS51 |
| *ELT,I DORMAN.SYNC1 | SYNC1 3 |
| SUBROUTINE SYNC1 | SYNC1 5 |
| COMMON /BFRS/ ISYN(2,20),ISYN1,IT1,IT2,IT3,FULL,CN1,CN2,LIN1,LIN2 | BFRS1 2 |
| * ,NWBUF,NB1,NB2,ICN1,ICN2 | BFRS1 3 |
| COMMON /BFRS/ BUF1(14,50),BUF2(14,50),AAA(700) | BFRS1 4 |
| INTEGER CN1,CN2,BUF1,BUF2 | BFRS1 5 |
| LOGICAL FULL | BFRS1 6 |
| INTEGER FIN1,FIN2 | SYNC1 7 |
| INTEGER FOUT | SYNC1 8 |
| EQUIVALENCE (FIN1,IT1) | SYNC1 9 |
| EQUIVALENCE (FIN2,IT2) | SYNC1 10 |
| EQUIVALENCE (FOUT,IT3) | SYNC1 11 |
| INTEGER PRTFIL | SYNC1 12 |
| DATA PRTFIL/6/ | SYNC1 13 |
| C | SYNC1 14 |
| C FILES NOW OUT OF SYNC,RECOVER | SYNC1 16 |
| C | SYNC1 17 |
| 21 CONTINUE | SYNC1 18 |
| FULL = .TRUE. | SYNC1 19 |
| IF(CN1.EQ.1) GO TO 30 | SYNC1 21 |
| J=1 | SYNC1 22 |

```

      DO 25 L=CN1,NB1
      DO 22 I=1,14
22   BUF1(I,J)=BUF1(I,L)
25   J=J+1
      NB1 = J - 1
      CN1 = 1
      IF(BUF1(1,NB1).EQ.6H$END 0) GO TO 30
28   CONTINUE
      CALL FILBUF(J,NB1,NWBUF,BUF1,FIN1)
30   CONTINUE
      IF(CN2.EQ.1) GO TO 40
      J=1
      DO 35 L=CN2,NB2
      DO 32 I=1,14
32   BUF2(I,J)=BUF2(I,L)
35   J=J+1
      NB2 = J - 1
      CN2 = 1
      IF(BUF2(1,NB2).EQ.6H$END 0) GO TO 40
38   CONTINUE
      CALL FILBUF(J,NB2,NWBUF,BUF2,FIN2)
40   CONTINUE
C
C   ATTEMPT TO MATCH IN BUFFER
C
10   CONTINUE
      N1 = ICN1 - LIN1 + 1
      N2 = ICN2 - LIN2 + 1
      IS=0
      NBX = NB1
      IF(NBX.GT.N1) NBX = N1
      NBY = NB2
      IF(NBY.GT.N2) NBY = N2
      DO 60 K=CN1,NBX
      LL=K-CN1+CN2
      IF(LL.GT.NBY) LL = NBY

```

```

SYNC1 23
SYNC1 24
SYNC1 25
SYNC1 26
GWT13 66
GWT13 67
GWT13 68
GWT13 69
SYNC1 27
SYNC1 29
SYNC1 31
SYNC1 32
SYNC1 33
SYNC1 34
SYNC1 35
SYNC1 36
GWT13 70
GWT13 71
GWT13 72
GWT13 73
SYNC1 37
SYNC1 39
SYNC1 40
SYNC1 41
SYNC1 42
SYNC1 43
SYNC1 44
SYNC1 45
SYNC1 46
SYNC1 47
SYNC1 48
SYNC1 49
SYNC1 50
SYNC1 51
SYNC1 52
SYNC1 53

```

| | |
|---------------------------------------|----------|
| DO 55 L=CN2,LL | SYN01 54 |
| IF(BUF1(1,K).NE.BUF2(1,L)) GO TO 55 | SYN01 55 |
| DO 50 I=2,14 | SYN01 56 |
| IF(BUF1(I,K).NE.BUF2(I,L)) GO TO 55 | SYN01 57 |
| 50 CONTINUE | SYN01 58 |
| GO TO 70 | SYN01 59 |
| 55 CONTINUE | SYN01 60 |
| 60 CONTINUE | SYN01 61 |
| K = NBX | GWT13 74 |
| L = NBY | GWT13 75 |
| IS= 1 | GWT13 76 |
| 70 CONTINUE | SYN01 64 |
| DO 73 N=CN2,NBY | SYN01 65 |
| NN=N-CN2+CN1 | SYN01 66 |
| IF(NN.GT.NBX) NN = NBX | SYN01 67 |
| DO 72 M=CN1,NN | SYN01 68 |
| IF(BUF1(1,M).NE.BUF2(1,N)) GO TO 72 | SYN01 69 |
| DO 71 I = 2,14 | SYN01 70 |
| IF(BUF1(I,M).NE.BUF2(I,N)) GO TO 72 | SYN01 71 |
| 71 CONTINUE | SYN01 72 |
| GO TO 171 | GWT 1 |
| 72 CONTINUE | SYN01 78 |
| 73 CONTINUE | SYN01 79 |
| M = NBX | GWT13 77 |
| N = NBY | GWT13 78 |
| IF(IS.EQ.0) GO TO 74 | SYN01 81 |
| GO TO 300 | SYN01 82 |
| 171 CONTINUE | GWT 2 |
| IF((K+L).LE.(M+N)) GO TO 74 | GWT 3 |
| L=N | GWT 4 |
| K=M | GWT 5 |
| 74 CONTINUE | SYN01 83 |
| C | SYN01 84 |
| C USE SYN CARUS | SYN01 85 |
| C | SYN01 86 |
| IF((K.LT.N1).AND.(L.LT.N2)) GO TO 100 | SYN01 87 |

| | | |
|--------|---|----------|
| | CALL CKSYN(ICN1,ICN2,ISYN1,1) | SYNG1 88 |
| 100 | CONTINUE | SYNG1 89 |
| | IF(K.NE.CN1) GO TO 80 | SYNG1 90 |
| C | | SYNG1 91 |
| C | MUST DO INSERT | SYNG1 92 |
| C | | SYNG1 93 |
| | CALL INSERT (L+LIN2-CN2,0) | SYNG1 94 |
| | RETURN | SYNG1 95 |
| C | | SYNG1 96 |
| C | MUST DO DELETE | SYNG1 97 |
| C | | SYNG1 98 |
| 30 | CONTINUE | SYNG1 99 |
| | CALL DELCD (K-CN1+LIN1) | SYNG1100 |
| | IF(L.NE.CN2) CALL INSERT (LIN2+L-CN2,1) | SYNG1101 |
| | RETURN | SYNG1102 |
| 300 | CONTINUE | SYNG1103 |
| | WRITE (PRTFIL,9) | SYNG1104 |
| 9 | FORMAT(54H THERE WILL BE A WAIT FOR OUT OF CORE SEARCH FOR MATCH) | SYNG1105 |
| | CALL SYNCBF(BUF1,LIN1,NB1,FIN1,K,ISYN1, | GWT+ 1 |
| * | BUF2,LIN2,NB2,FIN2,L) | GWT+ 2 |
| | CALL SYNCBF(BUF2,LIN2,NB2,FIN2,N,ISYN1, | GWT+ 3 |
| * | BUF1,LIN1,NB1,FIN1,M) | GWT+ 4 |
| | GO TO 171 | SYNG1110 |
| | END | SYNG1111 |
| #ELT,I | DORMAN.UNPAC | UNPAC 3 |
| | SUBROUTINE UNPAC(A,B,N) | UNPAC 5 |
| C | | UNPAC 6 |
| C | REMOVE = SIGNS AND REPLACE WITH BLANKS | UNPAC 7 |
| C | REMOVE ; | UNPAC 8 |
| C | | UNPAC 9 |
| C | PROGRAMMER: VUIT | UNPAC 10 |
| C | | UNPAC 11 |
| | COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC 2 |
| | INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC 3 |
| | EQUIVALENCE (PRTFIL,FERR) | MISC 4 |
| | INTEGER A(8+),B(84) | UNPAC 13 |

```

        ICHB = 1
        DO 10 I=1,84
        3(I) = 6H
10      CONTINUE
        DO 100 I=1,84
        N = I
        IF (A(I).EQ.1H;) RETURN
        IF (A(I).EQ.1H=) GO TO 70
        IF(A(I).EQ.1H ) GO TO 50
        IF(ICHB.GT.84) GO TO 220
        B(ICHB) = A(I)
        50 CONTINUE
        ICHB = ICHB+1
        IF(ICHB.GT.84) RETURN
        GO TO 100
        70 CONTINUE
        IF(ICHB.GT.71) GO TO 200
        ICHB=ICHB+1
        90 ICHB =((ICHB+9)/10)*10 + 1
100     CONTINUE
        GO TO 220

C
C         ERROR EXIT
C
200     CONTINUE
220     CONTINUE
        WRITE (FERR,310) A
        310 FORMAT(28H UNPAC - CARD IMAGE TOO LONG/1X,84A1)
        ERFLAG = 1
        CALL TERM
        RETURN
        END
*ELT,1 DORMAN.UREAL
        SUBROUTINE UREAD (IUNIT,ICRD)
C
C         PROGRAMMER: VOIT

```

```

UNPAC 14
UNPAC 15
UNPAC 16
UNPAC 17
UNPAC 18
UNPAC 19
UNPAC 20
UNPAC 21
UNPAC 22
UNPAC 23
UNPAC 24
UNPAC 25
UNPAC 26
UNPAC 27
UNPAC 28
UNPAC 29
UNPAC 30
GWT25 4
UNPAC 31
UNPAC 32
UNPAC 33
UNPAC 34
UNPAC 35
UNPAC 36
UNPAC 37
UNPAC 38
UNPAC 39
UNPAC 40
UNPAC 41
UNPAC 42
UNPAC 43
UNPAC 44
UREAD 3
UREAD 5
UREAD 6
UREAD 7

```

| | | | |
|--------|--|-------|----|
| C | | UREAD | 8 |
| C | | UREAD | 9 |
| | DIMENSION ICRD(14) | UREAD | 10 |
| | READ (IUNIT,10,END=200,FERR=200) (ICRD(I),I=1,14) | UREAD | 17 |
| | RETURN | UREAD | 19 |
| 200 | CONTINUE | UREAD | 20 |
| | DO 40 I=3,14 | UREAD | 21 |
| | ICRD(I) = 6H | UREAD | 22 |
| 40 | CONTINUE | UREAD | 23 |
| | ICRD(1) = 6H\$END 0 | UREAD | 24 |
| | ICRD(2) = 6HF FILE | UREAD | 25 |
| | RETURN | UREAD | 26 |
| 10 | FORMAT(13A6,A2) | UREAD | 27 |
| | END | UREAD | 28 |
| *ELT,I | DURMAN.USER | USER | 3 |
| | SUBROUTINE USER (IFLAG) | USER | 5 |
| C | | USER | 6 |
| C | CONTROLS FILES | USER | 7 |
| C | | USER | 8 |
| C | PROGRAMMER - S. WRAY | USER | 9 |
| C | | USER | 10 |
| | COMMON /NAMES/ IVER,DNAME(2),MODNAM(2),BNAME(2) | NAMES | 2 |
| | INTEGER DNAME,MODNAM,BNAME | NAMES | 3 |
| | COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2 | FILES | 2 |
| | INTEGER BASIC,MTAPE,FINAL,S1,S2 | FILES | 3 |
| | COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST | 2 |
| | INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST | 3 |
| | COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC | 2 |
| | INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC | 3 |
| | EQUIVALENCE (PRTFIL,FERR) | MISC | 4 |
| | DIMENSION IU(3),IR(3) | USER | 15 |
| | DATA IU,IR/0,0,0,2,2,2/ | USER | 16 |
| | IF(IFLAG.EQ.0) WRITE (PRTFIL,5) | USER | 17 |
| 5 | FORMAT(25H CREATE OPTION IS BLOCKED) | USER | 18 |
| | WRITE (PRTFIL,10) | USER | 19 |
| 10 | FORMAT(31H IF YOU WISH TO HAVE A DECK EXTRACTED FROM THE BANK/ | USER | 20 |

| | | |
|--|------|----|
| *49H ENTER THE NAME OF THE DECK, OTHERWISE ENTER DONE/9H READY -) | USER | 21 |
| CALL INCD (ACTION,5) | USER | 22 |
| IF(ACTION(1).EQ.4HDONE) RETURN | USER | 23 |
| IF((ACTION(1).EQ.MODNAM(1)).AND.(ACTION(2).EQ.MODNAM(2))) GO TO 15 | USER | 24 |
| IF((ACTION(1).EQ.BNAME(1)).AND.(ACTION(2).EQ.BNAME(2))) RETURN | USER | 25 |
| FINAL = 14 | USER | 26 |
| CALL USE (ACTION) | USER | 27 |
| RETURN | USER | 28 |
| 15 M1 = 13 | USER | 29 |
| FINAL = 14 | USER | 30 |
| IU(1) = M1 | USER | 31 |
| IU(2) = FINAL | USER | 32 |
| CALL INTBUF(IU,IR) | USER | 33 |
| REWIND M1 | USER | 34 |
| CALL INCD(KARD,M1) | USER | 35 |
| IF((KARD(5).EQ.BNAME(1)).AND.(KARD(6).EQ.BNAME(2))) GO TO 20 | USER | 36 |
| BNAME(1) = KARD(5) | USER | 37 |
| BNAME(2) = KARD(6) | USER | 38 |
| CALL USE (BNAME) | USER | 39 |
| 20 CONTINUE | USER | 40 |
| M2 = 23 | USER | 41 |
| IU(1) = M1 | USER | 42 |
| IU(2) = M2 | USER | 43 |
| IU(3) = FINAL | USER | 44 |
| CALL INTBUF(IU,IR) | USER | 45 |
| REWIND M1 | USER | 46 |
| REWIND M2 | USER | 47 |
| REWIND FINAL | USER | 48 |
| CALL EDITOK (FINAL,M1,M2) | USER | 49 |
| REWIND M2 | USER | 50 |
| REWIND FINAL | USER | 51 |
| 25 CALL INCD (KARD,M2) | USER | 52 |
| IF(ERFLAG.NE.0) RETURN | USER | 53 |
| CALL OUTCD (KARD,FINAL) | USER | 54 |
| IF(KARD(1).NE.END) GO TO 25 | USER | 55 |
| BNAME(1) = ACTION(1) | USER | 56 |

| | | |
|--------|--|----------|
| | BNAME(2) = ACTION(2) | USER 57 |
| | RETURN | USER 58 |
| | END | USER 59 |
| #ELT,I | DORMAN.VALUE | VALUE 3 |
| | SUBROUTINE VALUE (A,V,IERR) | VALUE 5 |
| C | CONVERTS NUMERIC DATA FROM CODED TO FLOATING POINT FORMAT. | VALUE 6 |
| C | UPON ENTRY, A IS A 2-CELL ARRAY CONTAINING CODED VALUE IN (A6,A4) | VALUE 7 |
| C | FORMAT, 1-10 DIGITS LOCATED ANYWHERE IN 10-CHARACTER FIELD. | VALUE 8 |
| C | DECIMAL POINT OPTIONAL. NO EXPONENTS OR + OR - SIGNS. NO | VALUE 9 |
| C | BLANKS EMBEDDED BETWEEN DIGITS. | VALUE 10 |
| C | UPON EXIT, V CONTAINS CONVERTED VALUE IN FLOATING POINT. IERR IS A | VALUE 11 |
| C | ERROR FLAG-- | VALUE 12 |
| C | IERR = 0 FOR NO ERROR. | VALUE 13 |
| C | IERR = -1 FOR COMPLETELY BLANK FIELD | VALUE 14 |
| C | IERR = 1,2,...,10 INDICATES POSITION OF ILLEGAL CHARACTER, | VALUE 15 |
| C | MULTIPLE DECIMAL POINTS, OR EMBEDDED BLANKS. | VALUE 16 |
| C | | VALUE 17 |
| | INTEGER A(1),C(19),CL(10),P(2) | VALUE 18 |
| | DATA C/19*1H / | VALUE 19 |
| | DATA CL/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/ | VALUE 20 |
| C | | VALUE 21 |
| C | STORE CHARACTERS INDIVIDUALLY IN C(10)-C(19) | VALUE 22 |
| C | | VALUE 23 |
| | DECODE (6,70,A(1))(C(I),I=10,15) | VALUE 24 |
| | DECODE (4,70,A(2))(C(I),I=16,19) | VALUE 25 |
| C | | VALUE 26 |
| C | FIND POSITION OF FIRST AND LAST NON-BLANK CHARACTERS | VALUE 27 |
| C | | VALUE 28 |
| | NF=100 | VALUE 29 |
| | NL=0 | VALUE 30 |
| | DO 10 I=10,19 | VALUE 31 |
| | IF (C(I).EQ.1H) GO TO 10 | VALUE 32 |
| | NF=MIN0(I,NF) | VALUE 33 |
| | NL=I | VALUE 34 |
| 10 | CONTINUE | VALUE 35 |
| | IF (NL.GT.0) GO TO 20 | VALUE 36 |

| | | |
|--------|---|----------|
| | IERR=-1 | VALUE 37 |
| | GO TO 60 | VALUE 38 |
| C | | VALUE 39 |
| C | CHECK FOR ILLEGAL CHARACTERS, INCLUDING EMBEDDED BLANKS OR MULTIPLE | VALUE 40 |
| C | DECIMAL POINTS | VALUE 41 |
| C | | VALUE 42 |
| 20 | NDP=0 | VALUE 43 |
| | DO 50 I=NF,NL | VALUE 44 |
| | DO 30 J=1,10 | VALUE 45 |
| | IF (C(I).EQ.CL(J)) GO TO 50 | VALUE 46 |
| 30 | CONTINUE | VALUE 47 |
| | IF (C(I).NE.1H.) GO TO 40 | VALUE 48 |
| | NDP=NDP+1 | VALUE 49 |
| | IF (NDP.EQ.1) GO TO 50 | VALUE 50 |
| 40 | IERR=I-9 | VALUE 51 |
| | GO TO 60 | VALUE 52 |
| 50 | CONTINUE | VALUE 53 |
| C | | VALUE 54 |
| C | REPACK CHARACTERS RIGHT-ADJUSTED IN FIELD OF 10 CHARACTERS. | VALUE 55 |
| C | | VALUE 56 |
| | N1=NL-9 | VALUE 57 |
| | ENCODE (10,70,P) (C(I),I=N1,NL) | VALUE 58 |
| | DECODE (10,80,P) V | VALUE 59 |
| | IERR=0 | VALUE 60 |
| 60 | CONTINUE | VALUE 61 |
| | RETURN | VALUE 62 |
| 70 | FORMAT (10A1) | VALUE 63 |
| 80 | FORMAT (F10.0) | VALUE 64 |
| | END | VALUE 65 |
| [ELT,I | DORMAN.ZEBRA | ZEBRA 3 |
| #FOR,S | DORMAN.BLOCK,.BLOCK | ZEBRA 4 |
| #FOR,S | DORMAN.DORMAN,.DORMAN | ZEBRA 5 |
| #FOR,S | DORMAN.REPL,.REPL | ZEBRA 6 |
| #FOR,S | DORMAN.USER,.USER | ZEBRA 7 |
| #FOR,S | DORMAN.SAVER,.SAVER | ZEBRA 8 |
| #FOR,S | DORMAN.ADDER,.ADDER | ZEBRA 9 |

| | |
|--------------------------------|----------|
| #FOR, S DORMAN.DELET, .DELET | ZEBRA 10 |
| #FOR, S DORMAN.EDITER, .EDITER | ZEBRA 11 |
| #FOR, S DORMAN.CONV, .CONV | ZEBRA 12 |
| #FOR, S DORMAN.OPT, .OPT | ZEBRA 13 |
| #FOR, S DORMAN.INCRT, .INCRT | ZEBRA 14 |
| #FOR, S DORMAN.ADDX, .ADDX | ZEBRA 15 |
| #FOR, S DORMAN.TERM, .TERM | ZEBRA 16 |
| #FOR, S DORMAN.LISTER, .LISTER | ZEBRA 17 |
| #FOR, S DORMAN.A1TA6, .A1TA6 | ZEBRA 18 |
| #FOR, S DORMAN.A6TA1, .A6TA1 | ZEBRA 19 |
| #FOR, S DORMAN.CLOSE, .CLOSE | ZEBRA 20 |
| #FOR, S DORMAN.ASSGN, .ASSGN | ZEBRA 21 |
| #FOR, S DORMAN.DIFDEC, .DIFDEC | ZEBRA 22 |
| #FOR, S DORMAN.FILBUF, .FILBUF | ZEBRA 23 |
| #FOR, S DORMAN.FNDBUF, .FNDBUF | ZEBRA 24 |
| #FOR, S DORMAN.INCD, .INCD | ZEBRA 25 |
| #FOR, S DORMAN.INTBUF, .INTBUF | ZEBRA 26 |
| #FOR, S DORMAN.LABLER, .LABLER | ZEBRA 27 |
| #FOR, S DORMAN.LCRD, .LCRD | ZEBRA 28 |
| #FOR, S DORMAN.OUTCD, .OUTCD | ZEBRA 29 |
| #FOR, S DORMAN.PACCON, .PACCON | ZEBRA 30 |
| #FOR, S DORMAN.RESTOR, .RESTOR | ZEBRA 31 |
| #FOR, S DORMAN.SYNCBF, .SYNCBF | ZEBRA 32 |
| #FOR, S DORMAN.UNPAC, .UNPAC | ZEBRA 33 |
| #FOR, S DORMAN.UREAD, .UREAD | ZEBRA 34 |
| #FOR, S DORMAN.GETGEN, .GETGEN | ZEBRA 35 |
| #FOR, S DORMAN.LISTG, .LISTG | ZEBRA 36 |
| #FOR, S DORMAN.LISTTC, .LISTTC | ZEBRA 37 |
| #FOR, S DORMAN.USE, .USE | ZEBRA 38 |
| #FOR, S DORMAN.ADD, .ADD | ZEBRA 39 |
| #FOR, S DORMAN.REPLAC, .REPLAC | ZEBRA 40 |
| #FOR, S DORMAN.DELETE, .DELETE | ZEBRA 41 |
| #FOR, S DORMAN.COUNT, .COUNT | ZEBRA 42 |
| #FOR, S DORMAN.EDITOK, .EDITOK | ZEBRA 43 |
| #FOR, S DORMAN.VALUE, .VALUE | ZEBRA 44 |
| #FOR, S DORMAN.SYNCDS, .SYNCDS | ZEBRA 45 |

| | |
|--|----------|
| #FOR,S DORMAN.SYNC1,.SYNC1 | ZEBRA 46 |
| #FOR,S DORMAN.DELCD,.DELCD | ZEBRA 47 |
| #FOR,S DORMAN.INSERT,.INSERT | ZEBRA 48 |
| #FOR,S DORMAN.CKSYN,.CKSYN | ZEBRA 49 |
| #FOR,S DORMAN.EXTRAN,.EXTRAN | ZEBRA 50 |
| #ELT,I DORMAN.USE | USE 3 |
| SUBROUTINE USE(NAME) | USE 5 |
| C PROGRAMMER - B. GOLD | USE 6 |
| DIMENSION NAME(2),KG(20) | USE 7 |
| DIMENSION IU(3),IR(3) | USE 8 |
| INTEGER OUT | USE 9 |
| COMMON /MISC/ERFLAG,FERR,KARD(14),ACTION(14) | MISC 2 |
| INTEGER ERFLAG,FERR,ACTION,PRTFIL | MISC 3 |
| EQUIVALENCE (PRTFIL,FERR) | MISC 4 |
| COMMON /FILES/ BASIC,MTAPE,FINAL,S1,S2 | FILES 2 |
| INTEGER BASIC,MTAPE,FINAL,S1,S2 | FILES 3 |
| COMMON /REST/TABLE,USES,FILE,END,DECK,DDECK,BLANK ,BATCH | REST 2 |
| INTEGER TABLE,USES,FILE,END,DECK,DDECK,BLANK,BATCH | REST 3 |
| COMMON /NAMES/ IVER,DNAME(2),MODNAM(2),BNAME(2) | NAMES 2 |
| INTEGER DNAME,MODNAM,BNAME | NAMES 3 |
| INTEGER YES,NO | USE 14 |
| INTEGER WORD | USE 15 |
| INTEGER VERS | USE 16 |
| DATA VERS /6HVERS10/ | USE 17 |
| DATA YES,NO/3HYES,2HNO/ | USE 18 |
| C | USE 19 |
| C S1 AND S2 ARE SCRATCH TAPES | USE 20 |
| C MTAPE IS THE TAPE ONTO WHICH THE MOD DECKS ARE COPIED | USE 21 |
| C | USE 22 |
| S1 = 25 | USE 23 |
| S2 = 26 | USE 24 |
| MTAPE = 27 | USE 25 |
| IU(1) = BASIC | USE 26 |
| IU(2) = S1 | USE 27 |
| IU(3) = MTAPE | USE 28 |
| IR(1) = 2 | USE 29 |

| | | |
|--|-----|----|
| IR(2) = 2 | USE | 30 |
| IR(3) = 2 | USE | 31 |
| CALL INIBUF(10,IR) | USE | 32 |
| ERFLAG = 0 | USE | 33 |
| REWIND BASIC | USE | 34 |
| REWIND MTAPE | USE | 35 |
| REWIND FINAL | USE | 36 |
| NG = 0 | USE | 37 |
| LASTM = 0 | USE | 38 |
| LASTB = 0 | USE | 39 |
| NAME1 = NAME(1) | USE | 40 |
| NAME2 = NAME(2) | USE | 41 |
| IFLG = 1 | USE | 42 |
| NTC = 0 | USE | 43 |
| C | USE | 44 |
| C READ TABLE OF CONTENTS. STORE LIST OF ALL DECKS ON BASIC FILE IN | USE | 45 |
| C ARRAY TC. STORE GENEALOGY OF DECK NAME IN ARRAY G. | USE | 46 |
| C | USE | 47 |
| 10 CALL INCD(KARD,BASIC) | USE | 48 |
| IF(ERFLAG.NE.0) RETURN | USE | 49 |
| IF(KARD(5).EQ.VERS) GO TO 10 | USE | 50 |
| IF (KARD(1).EQ.END) GO TO 20 | USE | 51 |
| NTC = NTC+1 | USE | 52 |
| IF (KARD(4).EQ.BLANK) LASTB = NTC | USE | 53 |
| IF (KARD(4).EQ.USES) LASTM = NTC | USE | 54 |
| IF ((KARD(2).NE.NAME1).OR.(KARD(3).NE.NAME2)) GO TO 10 | USE | 55 |
| NG = NG+1 | USE | 56 |
| KG(NG) = NTC | USE | 57 |
| NAME1 = KARD(5) | USE | 58 |
| NAME2 = KARD(6) | USE | 59 |
| GO TO 10 | USE | 60 |
| C | USE | 61 |
| C READ MOD DECKS FROM BASIC FILE. IF REQUIRED FOR DECK NAME, STORE | USE | 62 |
| C MOD DECK ON MTAPE. | USE | 63 |
| C | USE | 64 |
| 20 NDECK = 0 | USE | 65 |

| | | |
|---|-----|-----|
| IF (NAME1.NE.BLANK) GO TO 200 | USE | 66 |
| IFLG = 2 | USE | 67 |
| IF (NG.EQ.1) GO TO 50 | USE | 68 |
| IG = NG-1 | USE | 69 |
| 30 NDECK = NDECK+1 | USE | 70 |
| IF (NDECK.GT.NTC) GO TO 220 | USE | 71 |
| INDEX = LASTM - KG(IG) + 1 | USE | 72 |
| WORD=NO | USE | 73 |
| IF(INDEX.EQ.NDECK) WORD=YES | USE | 74 |
| 40 CALL INCD(KARD,BASIC) | USE | 75 |
| IF(ERFLAG.NE.0) RETURN | USE | 76 |
| IF (WORD.EQ.YES) CALL OUTCD(KARD,MTAPE) | USE | 77 |
| IF (KARD(1).NE.END) GO TO 40 | USE | 78 |
| IF (WORD.EQ.NO) GO TO 30 | USE | 79 |
| IG = IG-1 | USE | 80 |
| IF (IG.GT.0) GO TO 30 | USE | 81 |
| C | USE | 82 |
| C ALL MOD DECKS REQUIRED FOR DECK NAME HAVE BEEN PLACED ON MTAPE. | USE | 83 |
| C NOW FIND THE BASIC DECK REQUIRED. | USE | 84 |
| C | USE | 85 |
| 50 IFLG = 3 | USE | 86 |
| INDEX = LASTB - (KG(NG)-LASTM) + 1 | USE | 87 |
| 55 NDECK = NDECK+1 | USE | 88 |
| IF (NDECK.GT.NTC) GO TO 220 | USE | 89 |
| IF (NDECK.EQ.INDEX) GO TO 70 | USE | 90 |
| 60 CALL INCD(KARD,BASIC) | USE | 91 |
| IF(ERFLAG.NE.0) RETURN | USE | 92 |
| IF (KARD(1)-END) 60,55,60 | USE | 93 |
| C | USE | 94 |
| C BASIC FILE IS NOW POSITIONED AT BEGINNING OF BASIC DECK REQUIRED. | USE | 95 |
| C COMBINED DECKS VIA EDIT. | USE | 96 |
| C | USE | 97 |
| 70 IFLG = 4 | USE | 98 |
| REWIND MTAPE | USE | 99 |
| REWIND FINAL | USE | 100 |
| BNAME(1) = NAME(1) | USE | 101 |

| | | |
|---|-----|-----|
| BNAME(2) = NAME(2) | USE | 102 |
| IF (NG.EQ.1) GO TO 90 | USE | 103 |
| IU(2) = FINAL | USE | 104 |
| CALL INTBUF(IU,IR) | USE | 105 |
| 80 CALL INCD(KARD,BASIC) | USE | 106 |
| IF(ERFLAG.NE.0) RETURN | USE | 107 |
| CALL OUTCU(KARD,FINAL) | USE | 108 |
| IF (KARD(1).EQ.END) RETURN | USE | 109 |
| GO TO 80 | USE | 110 |
| 90 IN = BASIC | USE | 111 |
| OUT = S1 | USE | 112 |
| IG = NG | USE | 113 |
| 100 IG = IG-1 | USE | 114 |
| IF (IG.EQ.1) OUT = FINAL | USE | 115 |
| REWIND OUT | USE | 116 |
| IU(1) = IN | USE | 117 |
| IU(2) = MTAPE | USE | 118 |
| IU(3) = OUT | USE | 119 |
| CALL INTBUF(IU,IR) | USE | 120 |
| CALL EDITDK (IN,MTAPE,OUT) | USE | 121 |
| IF(ERFLAG.NE.0) RETURN | USE | 122 |
| IF (IG.EQ.1) RETURN | USE | 123 |
| IN = OUT | USE | 124 |
| OUT = S2 | USE | 125 |
| IF (IN.EQ.S2) OUT = S1 | USE | 126 |
| REWIND IN | USE | 127 |
| GO TO 100 | USE | 128 |
| C | USE | 129 |
| C ERROR STOPS | USE | 130 |
| C | USE | 131 |
| 200 WRITE (FERR,210) NAME1, NAME2, NAME | USE | 132 |
| 210 FORMAT(28HCOULD NOT FIND DECK NAMED *2A6,20H* REQUIRED FOR DECK , | USE | 133 |
| *2A6) | USE | 134 |
| ERFLAG = 1 | USE | 135 |
| RETURN | USE | 136 |
| 220 WRITE (FERR,230) NAME | USE | 137 |

```
230 FORMAT(38H0INDEXING ERROR IN USE FOR DECK NAMED 2A6)
    ERFLAG = 1
    RETURN
    END
```

```
USE 138
USE 139
USE 140
USE 141
```