Fol. 41 H

This paper not to be cited without prior reference to the author

International Council
for the Exploration of
the Sea

C.M. 1982/H:21
Pelagic Fish Committee
Ref.:
Demersal Fish Committee
Statistical Committee

Fiskeridirektoratet
Biblioteket

PROGRAMS FOR HANDLING AND ANALYSIS OF STOMACH CONTENTS DATA

by

Trond Westgård
Institute of Marine Research
P.O. box 1870
5011 Bergen-Nordnes
Norway

Abstract

A set of programs for convenient and flexible storage and retrieval of stomach contents data is presented. The data are stored in five levels, STATIONDATA, PREDATORDATA, PREDATORSIZE-DATA, PREYDATA and PREYSIZEDATA. Between these levels there are pointers up and down in the database. The pointers are automatically produced by the program. Data input is given to a terminal on a dataform with the same layout as the actual handwritten one. A facility to correct delete and add dataforms is provided. The retrieval program gives the user the possibility of interactivel to choose area, predator species, predator species size-group, time of the year, five different taxonomic groupings (species or genus or family....) of prey and if size of prey should be included or not. A full list of programs except 5 routines are given.

Contents:

Fiskeridirektoratet
Biblioteket

1. Introduction

When a stomach-content sampling program is started a vast amount of diverse data is gathered and stored. The importance of a flexible and logical storage of data is obvious. The questions that could be asked "to the material" are endless and it must be easy to write programs to answer them. Access to the data should be direct and execution of program fast.

For stomach-sampling projects that has access to a medium-sized computer and some knowledge of FORTRAN, the present program-package will be an useful way to store data.

The end-user, often a technical assistant, has only to update his file of dataforms, in such a way that his own file has the same sequence of the dataforms as the file in the computer. The logic of the program will do the rest to produce different types of tables and summaries.

The present program has evolved in the Norwegian project "Bestandsberegninger med flerartsmodeller" which is a part of the ICES international stomach-sampling project in the North Sea. The project is supported by the Norwegian Fisheries Research Council (I701.72).

2. The dataform with coding instructions.

The dataform is divided into five fields, and follows the recommendations given in the draft manual for the ICES stomach sampling project. The station data in field 1 is the upper data level. It is coded on just the first dataform that belongs to that station, on all the other forms that belongs to the station field 1 is left open. Therefore it is mandatory that a manual check is done of the dataform-number in the upper right corner, this number is not coded into the computer.

Each station may have none, one or many predator species sampled. The NODC 10-digit species code of the predator is coded in field 2 (if no predator was caught on the station the field is left open). In the next dataform that belongs to that predator field 2 is left open. The data for each predator or sizegroup of this predator is coded in field 3. Each species code of the prey-species in the stomach is written in field 4 and each sizegroup of these prey-species is recorded in field 5. The prey-species-code is only written in the first line that belongs to that prey-species, the next lines are left open.

Field 1:

Column

1 - 2	Country-codes as used in "Manual on ICES oceanographic punch cards".
3 - 4	Ship-codes as used in "Manual on ICES oceanographic punch cards".
5 - 8	Number of the station.
9 - 14	Date the station was sampled.
15 - 18	Bottom depth in meter.
19 - 22	Fishing depth in meter.
23	Quadrant of the position (0=N and E, 1=N and W, 2=S and E, 3=S and W).
24 - 25	Latitude (degrees).
26 - 27	Latitude (minutes).
28 - 30	Longitude (degrees).
31 - 32	Longitude (minutes).
33	"Square system" (0=ICES statistical rectangle, 1=Norwegian statistical rectangle).

34 - 37 ICES statistical rectangle (e.g. 37F9) or Norwegian statistical rectangle (e.g. 4308).

38 - 39 Code for gear.

40 - 41 Duration of fishing (hours)

42 - 43 Duration of fishing (minutes)

44 - 45 Time of day (hours)

46 - 47 Time of day (minutes)

Field 2:

1 - 10 National Oceanographic Data Center (NODC), USA, 10-digit species-code.

Field 3:

1 - 7 Number of individuals caught per hour of the actual predator.

8 - 15 Sizegroup in cm. If an individual length is taken this is recorded in column 8-11. 12-15 is then left open.

16 Sex of the animal.

17 - 18 Maturity stage of the animal.

19 - 20 Age of the animal.

21 Units the weight is measured in (0=mg, 1=grams, 2=kilograms, 3=tonnes).

22 - 26 Weight of the animal.

Note: Colums 16-26 and 28-29 is left open when a group of animals are sampled.

27 Degree of autolysis of the stomach content. (0=no autolysis, 1= little autolysis, no stomach is disintegrated, 2= some autolysis, some stomachs disintegrated, 3= many stomachs disintegrated).

28 Degree of digestion. (1=fresh, 2=some digestion all species identifiable, 3=some animals recognizable, 4=just parts of animals recognizable, 5=fully digested).

29 Filling degree. (1=empty, 2=some content, 3=half full, 4=full, 5=distended, 6=turned inside out).

30 - 31 Number of stomachs with food (not regurgitated).

32 - 33 Number of animals that had regurgitated.

34 - 35 Number of animals with empty stomachs.

36 - 37 Sum of columns 30-35. (Could be left open).

Field 4:

1 - 10 NODC-10-digit species code.

Field 5:

1 Scale that is used for size-spesification (just one is implemented yet. This scale is recorded by blank or zero).

2 - 6 Sizegroup according to the scale defined in column 1.

Scale 0:

-1 = Egg,

0 = Nauplii,

1 = 0.01 - 0.019 cm,
2 = 0.02 - 0.029 cm,
3 = 0.03 - 0.039 cm,

INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY

DATAFORM NUMBER

BESTANDSBEREGNINGER MED FLERARTSMODELLER (BMF)

1

COUNTRY	SHIP	ST. NR.	YEAR	MNT.	DAY
1 2	3 4	5 6 7 8	9 10	11 12	13 14

BOTTOM DEPTH	FISHING DEPTH	POSITION
15 16 17 18	19 20 21 22	23 24 25 26 27 28 29 30 31 32

S	SQUARE	GEAR	FISHING TIME	TIME
33	34 35 36 37	38 39	40 41 42 43	44 45 46 47

2

SPECIES:

CODE
1 2 3 4 5 6 7 8 9 10

3

NR. PR. HR	SIZE (Cm)	SEX	MAT.
1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16	17 18

AGE	D	WEIGHT	AUT	DG	DF	N-FOOD	N-REG	N-EMP	N-SAM
19 20	21	22 23 24 25 26	27	28	29	30 31	32 33	34 35	36 37

4 **5**

	SPECIESCODE	S	SIZEGR.	D	WEIGHT	NUMBER	COMMENTS
	1 2 3 4 5 6 7 8 9 10	1	2 3 4 5 6	7	8 9 10 11 12 13 14	15 16 17 18 19 20 21	
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							

4 = 0.04 - 0.049 cm,
5 = 0.05 - 0.059 cm,
6 = 0.06 - 0.069 cm,
7 = 0.07 - 0.079 cm,
8 = 0.08 - 0.089 cm,
9 = 0.09 - 0.099 cm,
10 = 0.1 - 0.14 cm,
15 = 0.15 - 0.19 cm,
20 = 0.2 - 0.24 cm,
25 = 0.25 - 0.29 cm,
30 = 0.3 - 0.39 cm,
40 = 0.4 - 0.49 cm,
50 = 0.5 - 0.69 cm,
70 = 0.7 - 0.9 cm,
100 = 1.0 - 1.4 cm,
150 = 1.5 - 1.9 cm,
200 = 2.0 - 2.4 cm,
250 - 2.5 - 2.9 cm,
300 = 3.0 - 3.9 cm,
400 = 4 - 4.9 cm,
500 = 5 - 6.9 cm,
700 = 7 - 9.9 cm,
1000 =10 - 14 cm,
1500 =15 - 19 cm,
2000 =20 - 24 cm,
2500 =25 - 29 cm,
3000 =30 - 39 cm,
4000 =40 - 49 cm,
5000 =50 - 69 cm,
7000 =70 - 99 cm,
10000 =100 -149 cm,
99999 = Unknown or not recorded.

7 Scale of weight or volume (0=mg, 1=grams, 2=kilograms, 3=tonnes, 4=volume in ml, 5=volume in dl, 6=volume in l).

8 - 14 Weight or volume in units according to column 7.

15 - 21 Number of individuals in the actual prey species size group category.

3. The data structure

The material is naturally divided into 5 levels, and to get a minimum of storage space and fast access to data we divide the material into 5 random access files with pointers to link the data. We create pointers to link data down and up in the database. The structure is shown in figure 1. The symbols used are:X1NR - X5NR are the record numbers of the datafiles (used as pointers up in the database). P1 - P8 are the pointers down in the database. X1 - X5 are the arrays that contains the information when reading/writing from/on the files. C1-C5 characters of field 1-5 in a dataform. K - array that contains data from a whole dataform.

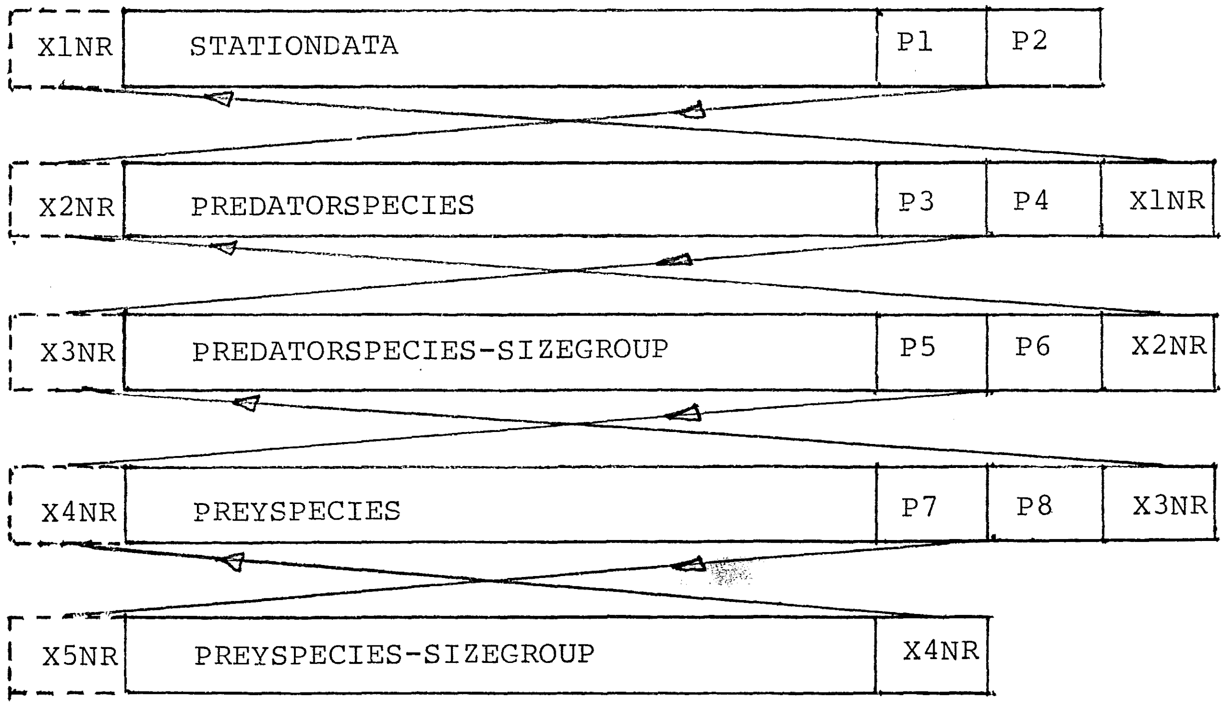

Fig. 1. The structure of the data files in the stomach-content data base.

4. Short description of the programs

The programs will be fully listed in chapter 6.

INN-SKJEMA-AM

The routine reads data from a fixed dataform on the screen. The screen handling routines made by P.G. Fadnes are used. The layout of the dataform on the screen is similar to the dataform on page 4. The number of the dataform will appear automatically in the upper right corner of the screen.

RETTE-SKJEMA-AM

The routine is used to insert new dataforms in the dataform file or delete forms from the file. One may also correct errors done when the dataform was punched under control of INN-SKJEMA--AM.

SKJEMA-TRANS-AM

Program that transfer the dataforms into the five structured files described in chapter 3.

UTSKRIFT-AM

Prints a full list of the five structured files.

TAB-POLY-AM

The program produces a summary table for the specified predators diet in a given area and time period. The area is an irregular polygon with up to 12 vertices. Wanted predator, predator size-group, timeperiod and area is given interactively on the screen. The user also choose the "taxonomic level" the prey should be presented at in the output table, and if prey size should be adhered to or not.

SORTX-AM

A sorting routine made by Gunnar Helle, Institute of Marine Research, Bergen.

NAVN-TRANS-AM

The program produces two files from a list of species codes (NODC-codes) with Latin, Norwegian and English names. The first file is a table of species codes and byte-pointers, the second file contains the names belonging to the codes. The species number codes are stored in increasing order.

NAVN-AM

This is a binary search routine that finds the corresponding name to a given 10-digit species code. The routine search in the table made by NAVN-TRANS-AM.

STRSEG-AM

The routine is made by Aage Fotland, Institute of Marine Research, Bergen. It finds the length of the none-blank field in a character field.

SAME-AM

The routine combines lines after certain rules and criteria. For further explanation consult the list of the program.

INS-AM

When geographical position is given the routine finds out if the position is inside or outside an irregular polygon with N vertices. The INSIDE routine made by "Oceanography EMR" is used.

SIZE-AM

A routine which returns the actual upper and lower limit in cm when a prey sizecode is given.

Screenhandling routines:

The routines are made by P.G. Fadnes.

ERPAGE: Blanks the screen and positions the cursor to the upper left corner.

ABSADR: Positions the cursor to a given line and column on the screen.
RECHA: Reads and verifies alpha-numeric characters from the screeen.
BACKSP: Backspaces the cursor and blanks the positions.

5. Necessary data files

QEDDYRENAVN

NODC-species code with Latin, Norwegian and English name stored sequentially in the order: ....., 10-digit code, Latin name, Norwegian name, English name, ...

DYREKODER

A sorted table of NODC-species codes with a bytepointer to the corresponding Latin name in the DYRENAVN file and the length of each of the three name types in bytes, stored in the order: ....., 10-digit code, byte address to start of Latin name, bytes in Latin name, bytes in Norwegian name, bytes in English name, ...

DYRENAVN

A sequential file stored in the order: ....Latin name, Norwegian name (if any),English name (if any)......

SKJEMA-AM

Contains the dataforms.

SKJEMA-STATUS-AM

Contains the number of dataforms in SKJEMA-AM.

STATUS

Contains the number of records in the five structured files.

FLAGG-AM

Contains the "jump"orders of the SKJEMA-AM file, to produce the five structured files.

**

STASJON *

PREDATOR *

PSTRGR * The five structured files.

BYTTEDYR *

BSTRGR *

**

6. List of programs

Five routines are not listed. These are ERPAGE, ABSADR, RECHA, BACKSP and INSIDE. The programs are written in NORD-FORTRAN which is a language that has the DO, FOR, ENDDO and IF(), THEN, ELSEIF(), ENDIF facilities. Datafiles are linked to the program with the OPEN-statement in the program. A full documentation of the syntax-rules is given in NORD-10 FORTRAN SYSTEM, ND-60.074.02 by Norsk Data A/S.

```
C ** INN-SKJEMA-AM **
C
C**********************************************************************
C                                                                     *
C      THE PROGRAM READS DATA FROM A FIXED DATAFORM ON THE SCREEN.    *
C      THE SCREEN-HANDLING ROUTINES ARE MADE BY P.G.FADNES.           *
C      TWO COMMANDS ARE AVAILABLE WHEN PUNCHING THE DATA IN           *
C      EACH DATA FIELD :                                              *
C                                                                     *
C                (CTRL)A : BACKSPACES THE CURSOR ONE                  *
C                          POSITION IN THE ACTUAL FIELD.              *
C                                                                     *
C                (CTRL)Q : BACKSPACES THE CURSOR TO THE               *
C                          START OF THE ACTUAL FIELD                  *
C                                                                     *
C      IN EACH POSITION IN FIELD 4 AND 5 IN THE DATA FORM THE         *
C      FOLLOWING COMMANDS WILL APPLY :                                *
C                                                                     *
C                 &      : THE ACTUAL DATA-FORM IS FINISHED.          *
C                 $      : AN ERROR IS DETECTED FURTHER UP IN         *
C                          THE DATA-FORM. THE PROGRAM WILL START      *
C                          ON THE TOP OF THE FORM AGAIN.              *
C                                                                     *
C   TROND WESTGARD, INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY      *
C                                                                     *
C**********************************************************************
C
      INTEGER TRMTYP,RECNR
      DIMENSION K(528)
      CHARACTER C1*47,C2*10,C3*37,C4(30)*10
      CHARACTER C5(30)*21,CFAST*34,SVAR*3
      EQUIVALENCE (K(1),C1),(K(25),C2),(K(30),C3)
      EQUIVALENCE (K(49),C4(1)),(K(199),C5(1))
      COMMON /TRM/ TRMTYP
      DATA CFAST/':             !                      !'/
      CALL MGTTY(0,TRMTYP)
      OPEN(10,FILE = 'SKJEMA-AM',ACCESS = 'WX',RECL=528)
      OPEN(11,FILE = 'SKJEMA-STATUS-AM',ACCESS = 'RW')
      OPEN(6,FILE = 'L-P-1',ACCESS='W')
      READ(11,100) RECNR
  100 FORMAT(I4)
      RECNR = RECNR + 1
      ISTART = RECNR
      ISTOPP = ISTART
C BLANKS THE SCREEN :
    1 CALL ERPAGE
C PUTS ALL CHARACTERS IN THE DATA-FORM TO 'SPACE' :
      DO FOR I = 1,528
         K(I) = 8224
      ENDDO
C CREATES THE DATA-FORM ON THE SCREEN :
      WRITE(1,101) RECNR
  101 FORMAT(12X,'BESTANDSBEREGNINGER MED FLERARTSMODELLER,BMF'
     F,/,27X,'1!',26X,'2',9X,'SKJNR:',I4,/,
     F1X,'1: 123456789.1234'10X'!'1X'1: 123456789.',/,
     F28X,'!',/,
     F28X,'!--------------------------',/,
     F1X,'2: 56789.123456789.12'6X'! 1: 123456789.12345678'5X,'3',/,
     F28X'!',/,
```

```
         F28X'!',/,
         F1X'3:.3456789.1234567'9X'! 2:.9.123456789.1234567',/,
         F28X,'!',/,
         F1X,'-------------!-------------!-------!-------------------',/,
         F13X,'4!'20X'5!',/,
         F4X,'123456789.!123456789.123456789.1!',/,
         F3X':'10X'!'21X'!',/,
         F3X':'10X'!'21X'!',/,
         F3X':'10X'!'21X'!'5X'FLER SKJEMA ?(_JA/NEI):',/,
         F3X':'10X'!'21X'!',/,
         F3X':'10X'!'21X'!',5X,'KOMMANDOER SOM GJELDER',/,
         F3X':'10X'!'21X'!',5X,'I FELT 4 OG 5 :',/,
         F3X':'10X'!'21X'!',5X,'& = FERDIG MED SKJEMA',/,
         F3X':'10X'!'21X'!',5X,'$ = FEIL LENGER OPP',/,
         F3X':'10X'!'21X'!',/,
         F3X':'10X'!'21X'!')
C READS DATA IN FIELD 1,2 AND 3 IN THE DATA-FORM :
          CALL ABSADR(4,3)
          CALL RECHA(14,C1(1:14),IST)
          CALL ABSADR(7,3)
          CALL RECHA(18,C1(15:32),IST)
          CALL ABSADR(10,3)
          CALL RECHA(15,C1(33:47),IST)
          CALL ABSADR(4,32)
          CALL RECHA(10,C2,IST)
          CALL ABSADR(7,32)
          CALL RECHA(18,C3(1:18),IST)
          CALL ABSADR(10,32)
          CALL RECHA(19,C3(19:37),IST)
C STARTS TO READ THE LINES 1 TO 30 IN FIELD 4 AND 5 IN THE DATA FORM :
          DO FOR J = 0,2,1
             DO FOR I = 1,10
                I1 = J*10 + I
                LIN = 12 + I
                CALL ABSADR(LIN,1)
                WRITE(1,102) I1,CFAST
  102 FORMAT(/,1H$,I2,A)
             ENDDO
             DO FOR I = 1,10
                I1 = J*10 + I
                LIN = 13 + I
                CALL ABSADR(LIN,3)
                CALL RECHA(10,C4(I1),IST)
C CHECKS IF THE COMMANDS '&' OR '$' IS GIVEN :
                DO FOR L1 = 1,10
                   IF(C4(I1)(L1:L1) .EQ. '&') THEN
                      DO FOR L2 = 1,10
                         C4(I1)(L2:L2) = ' '
                      ENDDO
                      GOTO 2
                   ELSEIF(C4(I1)(L1:L1) .EQ. '$') THEN
                      GOTO 1
                   ENDIF
                ENDDO
                CALL ABSADR(LIN,14)
                CALL RECHA(21,C5(I1),IST)
                DO FOR L1 = 1,21
                   IF(C5(I1)(L1:L1) .EQ. '&') THEN
                      DO FOR L2 = 1,21
                         C5(I1)(L2:L2) = ' '
                      ENDDO
```

```
                     GOTO 2
                  ELSEIF(C5(I1)(L1:L1) .EQ. '$') THEN
                     GOTO 1
                  ENDIF
               ENDDO
            ENDDO
         ENDDO
C STORES THE DATA GIVEN TO THE SCREEN :
       2 CALL WFILE(10,0,K(1),RECNR,528)
       3 CALL ABSADR(16,64)
         CALL RECHA(3,SVAR,IST)
         IF( SVAR .EQ. ' JA') THEN
            RECNR = RECNR + 1
            ISTOPP = RECNR
            GOTO 1
         ENDIF
         IF(SVAR .EQ. 'NEI') THEN
C WRITES THE PUNCHED DATA-FORMS ON THE LINE-PRINTER :
            DO FOR I = ISTART,ISTOPP
               CALL RFILE(10,0,K(1),I,528)
               WRITE(6,103) I
  103 FORMAT(1H1,12X,'BESTANDSBEREGNINGER MED FLERARTSMODELLER,BMF',3X,
     F 'SKJNR :',I5,/,31X,'1    !',30X,'2',/,
     F 10X,'123456789.1234',11X,'!',8X,'123456789.')
               WRITE(6,104) C1(1:14),C2
  104 FORMAT(10X,A,11X,'!',8X,A)
               WRITE(6,105) C1(15:32),C3(1:18)
  105 FORMAT(35X,'!---------------------------------',/,10X,
     F '56789.123456789.12'7X'!'8X'123456789.12345678'5X'3',/,
     F 10X,A,7X,'!',8X,A,/,35X,'!',/,10X,'3456789.1234567',10X,'!',8X,
     F '9.123456789.1234567')
               WRITE(6,106) C1(33:47),C3(19:37)
  106 FORMAT(10X,A,10X,'!',8X,A)
               WRITE(6,107)
  107 FORMAT(' -----------------!-----------------!-----!----------------
     F----------',/,16X,'4!',22X,'5!',/,
     F6X,'123456789.',1X,'!',1X,'123456789.123456789.1 !')
               DO FOR J = 1,30
                  WRITE(6,108) J,C4(J),C5(J)
  108 FORMAT(I4,': ',A,1X,'!',1X,A,' !')
               ENDDO
            ENDDO
C SAVES THE VARIABLE RECNR :
            REWIND 11
            WRITE(11,100) RECNR
            GOTO 4
         ELSE
            GOTO 3
         ENDIF
       4 CALL ERPAGE
         STOP ' GOODBYE, YOU WILL GET A PRINTOUT ON THE LINEPRINTER'
         END
```

```
C** RETTE-SKJEMA-AM **
C
C***********************************************************************
C                                                                      *
C       THE PROGRAM MAKES IT POSSIBLE TO CORRECT ERRORS IN PREVIOUSLY   *
C       PUNCHED DATA-FORMS. THE SAME COMMANDS APPLY AS IN THE           *
C       "INN-SKJEMA-AM" PROGRAM. DATA-FORMS MAY BE ADDED OR            *
C       DELETED FROM THE DATA-FORM FILE.                               *
C       SCREEN HANDLING ROUTINES MADE BY P.G. FADNES ARE USED.         *
C                                                                      *
C     TROND WESTGARD, INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY     *
C                                                                      *
C***********************************************************************
C
      INTEGER TRMTYP,RECNR
      DIMENSION K(528)
      CHARACTER C1*47,C2*10,C3*37,C4(30)*10
      CHARACTER C5(30)*21,CFAST*34,SVAR*3
      EQUIVALENCE (K(1),C1),(K(25),C2),(K(30),C3)
      EQUIVALENCE (K(49),C4(1)),(K(199),C5(1))
      COMMON /TRM/ TRMTYP
      DATA CFAST/':            !                   !'/
      CALL MGTTY(0,TRMTYP)
      OPEN(10,FILE = 'SKJEMA-AM',ACCESS = 'WX',RECL=528)
      OPEN(11,FILE = 'SKJEMA-STATUS-AM',ACCESS = 'RW')
      OPEN(6,FILE = 'L-P-1',ACCESS='W')
      READ(11,100) MAXREC
  100 FORMAT(I4)
   10 CALL ERPAGE
      WRITE(1,107)
  107 FORMAT(' FLER RETTINGER ?(JA/NEI)',/)
      READ(1,*) SVAR
      IF(SVAR .EQ. 'NEI') GOTO 11
      WRITE(1,110)
  110 FORMAT(' RETTE OPP GAMMELT SKJEMA ?(JA/NEI)',/)
      READ(1,*) SVAR
      IF(SVAR .EQ. 'JA') THEN
       WRITE(1,111)
  111  FORMAT(' GI SKJEMANR.:',/)
       READ(1,*) RECNR
       WRITE(1,112)
  112  FORMAT(' GI FELTNR. OG LINJENR. I FELT(EKS.: 1,3) :',/)
       READ(1,*) IFNR,LNR
       CALL RFILE(10,0,K(1),RECNR,528)
       IF(IFNR .EQ. 1) THEN
         IF(LNR .EQ. 1) THEN
           WRITE(1,113)(C1(J:J),J = 1,14)
  113      FORMAT(1X,14A,/)
           CALL RECHA(14,C1(1:14),IST)
           GOTO 50
         ELSEIF(LNR .EQ. 2) THEN
           WRITE(1,114)(C1(J:J),J = 15,32)
           CALL RECHA(18,C1(15:32),IST)      114 FORMAT(1X,18A,/)
           GOTO 50
         ELSE
           WRITE(1,115)(C1(J:J),J = 33,47)
           CALL RECHA(15,C1(33:47),IST)      115 FORMAT(1X,15A,/)
           GOTO 50
         ENDIF
       ENDIF
       IF(IFNR .EQ. 2) THEN
         WRITE(1,116)(C2(J:J),J = 1,10)
         CALL RECHA(10,C2(1:10),IST)         116 FORMAT(1X,10A,/)
         GOTO 50
```

```
          ENDIF
          IF(IFNR .EQ. 3) THEN
            IF(LNR .EQ. 1) THEN
              WRITE(1,114)(C3(J:J),J = 1,18)      ! 114 FORMAT(?
              CALL RECHA(18,C3(1:18),IST)
              GOTO 50
            ELSE
              WRITE(1,117)(C3(J:J),J = 19,37)
              CALL RECHA(19,C3(19:37),IST)         ! 117 FORMAT(1X,19A,/)
              GOTO 50
            ENDIF
           ENDIF
           IF(IFNR .EQ. 4) THEN
             WRITE(1,118)C4(LNR)                   ! 118 FORMAT(1X,A,/)
             CALL RECHA(10,C4(LNR),IST)
             GOTO 50
           ENDIF
          WRITE(1,118)C5(LNR)
          CALL RECHA(21,C5(LNR),IST)
   50     CALL WFILE(10,0,K(1),RECNR,528)
          GOTO 10
        ENDIF
        WRITE(1,103)
  103   FORMAT(' NYTT SKJEMA INN?(JA/NEI)',/)
        READ(1,*) SVAR
        IF(SVAR .EQ. 'NEI') GOTO 12
        WRITE(1,104)
  104   FORMAT(' NYTT SKJEMA SKAL INN ETTER SKJEMA ?',/)
        READ(1,*) RECNR
        IF(RECNR .GT. MAXREC) GOTO 10
        DO FOR I = MAXREC,RECNR+1,-1
        CALL RFILE(10,0,K(1),I,528)
        J = I + 1
        CALL WFILE(10,0,K(1),J,528)
        ENDDO
        MAXREC = MAXREC + 1
        RECNR = RECNR+1
        GOTO 1
   12   WRITE(1,105)
  105   FORMAT(' FJERNE SKJEMA?(JA/NEI)',/)
        READ(1,*) SVAR
        IF(SVAR .EQ. 'NEI') GOTO 11
        WRITE(1,106)
  106   FORMAT(' HVILKE SKJEMAER VIL DU FJERNE?(FRA/TIL)',/)
        READ(1,*) NR1,NR2
        IF(NR2 .GT. MAXREC) GOTO 12
        RECNR = NR1
        DO FOR INR = NR1,NR2
           IF(RECNR .EQ. MAXREC) THEN
              MAXREC = MAXREC - 1
           ELSE
              DO FOR I = RECNR + 1,MAXREC
                 CALL RFILE(10,0,K(1),I,528)
                 J = I-1
                 CALL WFILE(10,0,K(1),J,528)
              ENDDO
              MAXREC = MAXREC - 1
            ENDIF
        ENDDO
        GOTO 10
ERASE THE SCREEN AND PLACES THE CURSOR IN THE UPPER LEFT CORNER :
   1 CALL ERPAGE
```

```
C ALL CHARACTERS IN THE DATA-FORM ARE GIVEN THE VALUE 'SPACE' :
      DO FOR I = 1,528
         K(I) = 8224
      ENDDO
C CREATES THE DATA-FORM ON THE SCREEN, WITH ITS DATA-FORM NUMBER :
      WRITE(1,101) RECNR
  101 FORMAT(12X,'BESTANDSBEREGNINGER MED FLERARTSMODELLER,BMF'
     F,/,27X,'1!',26X,'2',9X,'SKJNR:',I4,/,
     F1X,'1: 123456789.1234'10X'!'1X'1: 123456789.',/,
     F28X,'!',/,
     F28X,'!---------------------------',/,
     F1X,'2: 56789.123456789.12'6X'! 1: 123456789.12345678'5X,'3',/,
     F28X'!',/,
     F28X'!',/,
     F1X'3: 3456789.1234567'9X'! 2: 9.123456789.1234567',/,
     F28X,'!',/,
     F1X,'-------------!-------------!-------!-------------------',/,
     F13X,'4!'20X'5!',/,
     F4X,'123456789.!123456789.123456789.1!',/,
     F3X':'10X'!'21X'!',/,
     F3X':'10X'!'21X'!',/,
     F3X':'10X'!'21X'!'5X'FLER SKJEMA ?(_JA/NEI):',/,
     F3X':'10X'!'21X'!',/,
     F3X':'10X'!'21X'!',5X,'KOMMANDOER SOM GJELDER',/,
     F3X':'10X'!'21X'!',5X,'I FELT 4 OG 5 :',/,
     F3X':'10X'!'21X'!',5X,'& = FERDIG MED SKJEMA',/,
     F3X':'10X'!'21X'!',5X,'$ = FEIL LENGER OPP',/,
     F3X':'10X'!'21X'!',/,
     F3X':'10X'!'21X'!')
C READS CHARACTERS IN FIELD 1, 2, 3 IN THE DATA-FORM :
      CALL ABSADR(4,3)
      CALL RECHA(14,C1(1:14),IST)
      CALL ABSADR(7,3)
      CALL RECHA(18,C1(15:32),IST)
      CALL ABSADR(10,3)
      CALL RECHA(15,C1(33:47),IST)
      CALL ABSADR(4,32)
      CALL RECHA(10,C2,IST)
      CALL ABSADR(7,32)
      CALL RECHA(18,C3(1:18),IST)
      CALL ABSADR(10,32)
      CALL RECHA(19,C3(19:37),IST)
C READS CHARACTERS INTO LINE 1,30 IN FIELD 4 AND 5 IN THE DATA-FORM :
      DO FOR J = 0,2,1
         DO FOR I = 1,10
            I1 = J*10 + I
            LIN = 12 + I
            CALL ABSADR(LIN,1)
            WRITE(1,102) I1,CFAST
  102 FORMAT(/,1H$,I2,A)
         ENDDO
         DO FOR I = 1,10
            I1 = J*10 + I
            LIN = 13 + I
            CALL ABSADR(LIN,3)
            CALL RECHA(10,C4(I1),IST)
C CHECKS IF THE COMMANDS & OR $ IS GIVEN :
            DO FOR L1 = 1,10
               IF(C4(I1)(L1:L1) .EQ. '&') THEN
                  DO FOR L2 = 1,10
                     C4(I1)(L2:L2) = ' '
                  ENDDO
                  GOTO 2
               ELSEIF(C4(I1)(L1:L1) .EQ. '$') THEN
```

```
                      GOTO 1
                   ENDIF
                ENDDO
                CALL ABSADR(LIN,14)
                CALL RECHA(21,C5(I1),IST)
                DO FOR L1 = 1,21
                   IF(C5(I1)(L1:L1) .EQ. '$') THEN
                      DO FOR L2 = 1,21
                         C5(I1)(L2:L2) = ' '
                      ENDDO
                      GOTO 2
                   ELSEIF(C5(I1)(L1:L1) .EQ. '$') THEN
                      GOTO 1
                   ENDIF
                ENDDO
             ENDDO
          ENDDO
C SAVES THE DATA THAT IS GIVEN :
    2 CALL WFILE(10,0,K(1),RECNR,528)
    3 CALL ABSADR(16,64)
      CALL RECHA(3,SVAR,IST)
      IF( SVAR .EQ. ' JA') GOTO 10
   11 IF(SVAR .EQ. 'NEI') THEN
         REWIND 11
         WRITE(11,100) MAXREC
         GOTO 4
      ELSE
         GOTO 3
      ENDIF
    4 CALL ERPAGE
      WRITE(1,120)
  120 FORMAT(' ØNSKER DU SKJEMAER SKREVET UT?(JA/NEI)',/)
      READ(1,*) SVAR
      IF(SVAR .EQ. 'JA') THEN
      WRITE(1,121)
  121 FORMAT(' GI SKJEMANR. FRA/TIL :',/)
      READ(1,*) ISTART,ISTOPP
      IF(ISTOPP .GT. MAXREC) ISTOPP = MAXREC
         DO FOR I = ISTART,ISTOPP
            CALL RFILE(10,0,K(1),I,528)
            WRITE(6,123) I
  123 FORMAT(1H1,12X,'BESTANDSBEREGNINGER MED FLERARTSMODELLER,BMF',3
     F 'SKJNR :',I5,/,31X,'1   !',30X,'2',/,
     F 10X,'123456789.1234',11X,'!',8X,'123456789.')
            WRITE(6,124) C1(1:14),C2
  124 FORMAT(10X,A,11X,'!',8X,A)
            WRITE(6,125) C1(15:32),C3(1:18)
  125 FORMAT(35X,'!---------------------------------',/,10X,
     F '56789.123456789.12'7X'!'8X'123456789.12345678'5X'3',/,
     F 10X,A,7X,'!',8X,A,/,35X,'!',/,10X,'3456789.1234567',10X,'!',8X,
     F '9.123456789.1234567')
            WRITE(6,126) C1(33:47),C3(19:37)
  126 FORMAT(10X,A,10X,'!',8X,A)
            WRITE(6,127)
  127 FORMAT(' ----------------!-----------------!-----!---------------
     F----------',/,16X,'4!',22X,'5!',/,
     F6X,'123456789.',1X,'!',1X,'123456789.123456789.1 !')
            DO FOR J = 1,30
               WRITE(6,128) J,C4(J),C5(J)
  128 FORMAT(I4,': ',A,1X,'!',1X,A,' !')
            ENDDO
      ENDDO
```

```
          WRITE(1,129)
      129 FORMAT(/,' UT AV PROGRAM ?(JA/NEI)',/)
          READ(1,*) SVAR
          IF(SVAR .EQ. 'NEI') GOTO 10
          WRITE(1,109)
      109 FORMAT(/,' TAKK FOR NÅ !!',//,' DU FÅR LISTE PÅ LINJESKRIVER')
          ENDIF
          END
```

```
C** SKJEMA-TRANS-AM **
C
C**********************************************************************
C                                                                     *
C       PROGRAM THAT CREATES FIVE STRUCTURED FILES FROM THE DATA-FORMS *
C       THE PROGRAM CREATES THE POINTERS, TO LINK THE DATA UP AND     *
C       DOWN IN THE DATABASE. THE FILES ARE :                         *
C                                                                     *
C                   1. STATIONDATA,P1,P2                              *
C                   2. PREDATORSPECIES,P3,P4,X1NR                     *
C                   3. PREDATORSPECIES-SIZEGROUP,P5,P6,X2NR           *
C                   4. PREYSPECIES,P7,P8,X3NR                         *
C                   5. PREYSPECIES-SIZEGROUP,X4NR                     *
C                                                                     *
C       THE PROGRAM FIRST CREATES A "FLAG"-FILE FROM THE DATA-FORMS   *
C       THIS FLAG-FILE WILL THEN TAKE THE PROPER ACTION WHEN THE      *
C       DATA-FORMS ARE RE-READ THROUGH THE SECOND PART OF THE PROGRAM *
C       THE PROGRAM UPDATES THE STATUS-FILE THAT TELLS HOW MANY RECORDS *
C       EACH OF THE FIVE STRUCTURED FILES CONTAINS.                   *
C                                                                     *
C     TROND WESTGÅRD, INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY    *
C                                                                     *
C**********************************************************************
C
      DIMENSION X1(22),X2(5),X3(15),X4(5),X5(5),KX1(68),KX2(18)
      DIMENSION KX3(48),KX4(18),KX5(16),K(528),FLAGG5(30),FL(32)
      CHARACTER*3 SVAR1,SVAR2,CH1*1,CH2*1
      INTEGER P1,P2,P3,P4,P5,P6,P7,P8,X1NR,X2NR,X3NR,FLAGG,FLAGG1
      INTEGER X4NR,X5NR,SSNR,SKJNR,SSKJNR,PP5,PP6,PP1,PP2,FLAGG3,FLAGG5
      INTEGER FL
```

```
      CHARACTER C1*47,C2*10,C3*37,C4(30)*10
      CHARACTER C5(30)*21,C11*47,C22*10,C33*37,C44(30)*10,C55(30)*21
      EQUIVALENCE (K(1),C1),(K(25),C2),(K(30),C3)
      EQUIVALENCE (K(49),C4(1)),(K(199),C5(1))
      EQUIVALENCE (KX1(1),X1(1)),(KX1(67),P1),(KX1(68),P2)
      EQUIVALENCE(KX2(1),X2(1)),(KX2(16),P3),(KX2(17),P4),(KX2(18),X1NR)
      EQUIVALENCE(KX3(1),X3(1)),(KX3(46),P5),(KX3(47),P6),(KX3(48),X2NR)
      EQUIVALENCE(KX4(1),X4(1)),(KX4(16),P7),(KX4(17),P8),(KX4(18),X3NR)
      EQUIVALENCE (KX5(1),X5(1)),(KX5(16),X4NR)
      EQUIVALENCE (FL(1),FLAGG1),(FL(2),FLAGG3),(FL(3),FLAGG5(1))
      DATA P2,P4,P6,P8,SSNR,X1NR,X2NR,X3NR,X4NR,X5NR/5*-1,5*0/
C OPENS THE NECESSARY FILES :
      OPEN(10,FILE = 'STATUS',ACCESS = 'RW')
      OPEN(11,FILE='STASJON',STATUS='UNKNOWN',ACCESS='WX',RECL =68)
      OPEN(12,FILE='PREDATOR',STATUS = 'UNKNOWN',ACCESS = 'WX',RECL=18)
      OPEN(13,FILE = 'PSTRGR',STATUS = 'UNKNOWN',ACCESS ='WX',RECL=48)
      OPEN(14,FILE='BYTTEDYR',STATUS = 'UNKNOWN',ACCESS = 'WX',RECL=18)
      OPEN(15,FILE = 'BSTRGR',STATUS = 'UNKNOWN',ACCESS = 'WX',RECL=16)
      OPEN(16,FILE = 'SKJEMA-AM',ACCESS = 'WX',RECL = 528)
      OPEN(17,FILE = 'SKJEMA-STATUS-AM',ACCESS = 'RW')
      OPEN(18,FILE = 'FLAGG-AM',ACCESS = 'WX',RECL = 32)
      OPEN(6,FILE = 'L-P-1',ACCESS = 'W')
      READ(17,100) SSKJNR
  100 FORMAT(I4)
C
C********** GENERATES THE FLAG-FILE TO GET THE PROPER JUMPS ***********
C
      DO  5  SKJNR = 1,SSKJNR - 1
         IF(SKJNR .GT. 1) CALL WFILE(18,0,FL(1),SKJNR-1,32)
         CALL RFILE(16,0,K(1),SKJNR+1,528)
         C11 = C1
         C22 = C2
         C33 = C3
         DO FOR I = 1,30
            C44(I) = C4(I)
            C55(I) = C5(I)
         ENDDO
         CALL RFILE(16,0,K(1),SKJNR,528)
C ASSIGN VALUES TO THE FLAG-VARIABLES :
         FLAGG1 = 0
         FLAGG3 = 0
         DO FOR I = 1,30
            FLAGG5(I) = 0
         ENDDO
         DO FOR I = 1,47
            IF(C1(I:I) .NE. ' ') THEN
               DO FOR J = 1,10
                  IF(C2(J:J) .NE. ' ') THEN
                     GOTO 1
                  ENDIF
               ENDDO
               FLAGG1 = -1
               GOTO 5
            ENDIF
         ENDDO
    1    DO FOR I = 1,10
            IF(C4(1)(I:I) .NE. ' ') GOTO 2
         ENDDO
         DO FOR I = 1,47
            IF(C11(I:I) .NE. ' ') THEN
```

```
                 FLAGG3 = -1
                 GOTO 5
              ENDIF
           ENDDO
           DO FOR I = 1,10
              IF(C22(I:I) .NE. ' ') THEN
                 FLAGG3 = -2
                 GOTO 5
              ENDIF
           ENDDO
           DO FOR I = 1,37
              IF(C33(I:I) .NE. ' ') THEN
                 FLAGG3 = -3
                 GOTO 5
              ENDIF
           ENDDO
C ASSIGN VALUES TO THE ARRAY FLAGG5(I) :
    2      I = 0
    3      I = I + 1
           IF(I .EQ. 30) GOTO 4
           DO FOR J = 1,10
              IF(C4(I + 1)(J:J) .NE. ' ') THEN
                 FLAGG5(I) = -4
                 GOTO 3
              ENDIF
           ENDDO
           DO FOR J = 1,21
              IF(C5(I+1)(J:J) .NE. ' ') GOTO 3
           ENDDO
    4      DO FOR J = 1,47
              IF(C11(J:J) .NE. ' ') THEN
                 FLAGG5(I) = -1
                 GOTO 5
              ENDIF
           ENDDO
           DO FOR J = 1,10
              IF(C22(J:J) .NE. ' ') THEN
                 FLAGG5(I) = -2
                 GOTO 5
              ENDIF
           ENDDO
           DO FOR J = 1,37
              IF(C33(J:J) .NE. ' ') THEN
                 FLAGG5(I) = -3
                 GOTO 5
              ENDIF
           ENDDO
           DO FOR J = 1,10
              IF(C44(I)(J:J) .NE. ' ')THEN
                 FLAGG5(I) = -4
                 GOTO 5
              ENDIF
              FLAGG5(I) = -5
           ENDDO
    5 CONTINUE
      CALL WFILE(18,0,FL(1),SKJNR-1,32)
C DECIDES THE VALUES OF THE FLAGS FOR THE LAST DATA-FORM :
      FLAGG1 = 0
      FLAGG3 = 0
      DO FOR I = 1,30
```

```
            FLAGG5(I) = 0
         ENDDO
         DO FOR I = 1,47
            IF(C11(I:I) .NE. ' ') THEN
               DO FOR J = 1,10
                  IF(C22(J:J) .NE. ' ') THEN
                     GOTO 6
                  ENDIF
               ENDDO
               FLAGG1 = -6
               GOTO 8
            ENDIF
         ENDDO
       6 IF(C33(36:36) .EQ. ' ' .AND. C33(37:37) .EQ. ' ') THEN
            FLAGG3 = -6
            GOTO 8
         ENDIF
         I = 0
       7 I = I + 1
         IF(I .EQ. 30) THEN
            FLAGG5(I) = -6
            GOTO 8
         ENDIF
         DO FOR J = 1,10
            IF(C44(I+1)(J:J) .NE. ' ') THEN
               FLAGG5(I) = -4
               GOTO 7
            ENDIF
         ENDDO
         DO FOR J = 1,21
            IF(C55(I+1)(J:J) .NE. ' ') GOTO 7
         ENDDO
         FLAGG5(I) = -6
       8 CALL WFILE(18,0,FL(1),SSKJNR,32)
C
C***********           FLAG-FILE READY             ***************************
C
C********* WRITES THE DATA FROM THE DATA-FORMS INTO THE ****************
C**************** FIVE STRUCTURED FILES *********************************
C
         SKJNR = 1
         CALL RFILE(16,0,K(1),SKJNR,528)
         CALL RFILE(18,0,FL(1),SKJNR,32)
       9 READ(C1,103)(X1(I),I=1,15),CH1,CH2,(X1(I),I=18,22)
     103 FORMAT(2F2.0,F4.0,3F2.0,2F4.0,F1.0,2F2.0,F3.0,F2.0,F1.0,F2.0,2A1,5
        F F2.0)
         X1(16) = ICHAR(CH1)
         X1(17) = ICHAR(CH2)
         FLAGG = FLAGG1
         IF(FLAGG .EQ. -1) THEN
            PP1 = P1
            PP2 = P2
            P1 = -1
            P2 = -1
            X1NR = SSNR + 1
            SSNR = X1NR
C WRITES THE STATIONRECORD WITH POINTERS :
            CALL WFILE(11,0,KX1(1),X1NR,68)
            X1NR = X1NR + 1
            P1 = PP1
```

```
            P2 = PP2
            SKJNR = SKJNR + 1
            CALL RFILE(16,0,K(1),SKJNR,528)
            CALL RFILE(18,0,FL(1),SKJNR,32)
            GOTO 9
         ENDIF
         T2 = -1
      10 READ(C2,107)(X2(I),I = 1,5)
     107 FORMAT(5F2.0)
         T3 = -1
C THE PREDATORSPECIES-RECORD MUST HAVE A PRED.SPEC.SIZEGROUP-RECORD :
      11 READ(C3,109)(X3(I), I = 1,15)
     109 FORMAT(F7.1,2F4.1,F1.0,2F2.0,F1.0,F5.2,3F1.0,4F2.0)
C CHECKS IF STOMACH-CONTENT WERE DETECTED :
         FLAGG = FLAGG3
         IF(FLAGG .LT. 0) THEN
            T3 = T3 + 1
            X3NR = P4 + 1
            P4 = X3NR
            PP5 = P5
            PP6 = P6
            P5 = -1
            P6 = -1
C WRITES THE PRED.SPEC.SIZEGROUP-RECORD WITH POINTERS :
            CALL WFILE(13,0,KX3(1),X3NR,48)
            X3NR = X3NR + 1
            P5 = PP5
            P6 = PP6
         ELSE
            GOTO 12
         ENDIF
C DECIDES WHAT RECORD THAT SHOULD BE WRITTEN NEXT :
         IF(FLAGG .EQ. -1 .OR. FLAGG .EQ. -2) GOTO 16
         IF(FLAGG .EQ. -6) GOTO 17
         IF(FLAGG .EQ. -3) THEN
            SKJNR = SKJNR + 1
            CALL RFILE(16,0,K(1),SKJNR,528)
            CALL RFILE(18,0,FL(1),SKJNR,32)
            GOTO 11
         ENDIF
C READS THE PREYSPECIES-RECORD :
      12 I1 = 1
         T4 = -1
      13 READ(C4(I1),107)(X4(I), I = 1,5)
C READS THE PREYSPECIES-SIZEGROUP-RECORD :
         T5 = -1
      14 READ(C5(I1),113)(X5(I),I = 1,5)
     113 FORMAT(F1.0,F5.0,F1.0,F7.3,F7.0)
         FLAGG = FLAGG5(I1)
         T5 = T5 + 1
         X5NR = P8 + 1
         P8 = X5NR
C SAVES THE PREYSPECIES-SIZEGROUP-RECORD :
         CALL WFILE(15,0,KX5(1),X5NR,16)
         IF(FLAGG .LT. 0) GOTO 15
         I1 = I1 + 1
         GOTO 14
      15 T4 = T4 + 1
         P7 = P8 - T5
         I1 = I1 + 1
```

```
      X4NR =P6 + 1
      P6 = X4NR
C SAVES THE PREYSPECIES-RECORD :
      CALL WFILE(14,0,KX4(1),X4NR,18)
      X4NR = X4NR + 1
      IF(FLAGG .EQ. -4) GOTO 13
      T3 = T3 + 1
      P5 = P6 - T4
      X3NR = P4 + 1
      P4 = X3NR
C SAVES THE PREDATORSPECIES-SIZEGROUP-RECORD :
      CALL WFILE(13,0,KX3(1),X3NR,48)
      X3NR = X3NR + 1
      IF(FLAGG .EQ. -3) THEN
         SKJNR = SKJNR + 1
         CALL RFILE(16,0,K(1),SKJNR,528)
         CALL RFILE(18,0,FL(1),SKJNR,32)
         GOTO 11
      ENDIF
   16 T2 = T2 + 1
      P3 = P4 - T3
      X2NR = P2 + 1
      P2 = X2NR
C SAVES THE PREDATORSPECIES-RECORD :
      CALL WFILE(12,0,KX2(1),X2NR,18)
      X2NR = X2NR + 1
      IF(FLAGG .EQ. -2) THEN
         SKJNR = SKJNR + 1
         CALL RFILE(16,0,K(1),SKJNR,528)
         CALL RFILE(18,0,FL(1),SKJNR,32)
         GOTO 10
      ENDIF
      P1 = P2 - T2
      X1NR = SSNR + 1
      SSNR = X1NR
C SAVES THE STATIONRECORD WITH POINTERS :
      CALL WFILE(11,0,KX1(1),X1NR,68)
      X1NR = X1NR + 1
      IF(FLAGG .EQ. -1) THEN
         SKJNR = SKJNR + 1
         CALL RFILE(16,0,K(1),SKJNR,528)
         CALL RFILE(18,0,FL(1),SKJNR,32)
         GOTO 9
      ENDIF
C
C************ STRUCTURED FILES READY, UPDATES "STATUS"-FILE ************
C
   17 WRITE(10,120) P2,P4,P6,P8,SSNR,SKJNR,X1NR,X2NR,X3NR,X4NR,X5NR
  120 FORMAT(11I6)
      STOP ' TRANSFER OF DATAFORMS INTO STRUCTURED FILES READY '
      END
```

```
C** UTSKRIFT-AM **
C
C***********************************************************************
C                                                                      *
C        THE PROGRAM PRODUCES A PRINTOUT OF THE CONTENT IN THE         *
C        FIVE STRUCTURED FILES CREATED OF THE PROGRAM "SKJEMA-TRANS-AM" *
C        THE POINTERS OF THE DATBASE ARE LISTED FOR CONTROL.           *
C                                                                      *
C      TROND WESTGÅRD, INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY    *
C                                                                      *
C***********************************************************************
C
      DIMENSION X1(22),X2(5),X3(15),X4(5),X5(5),KX1(68),KX2(18)
      DIMENSION KX3(48),KX4(18),KX5(16)
      CHARACTER*3 SVAR1,SVAR2,CH1*1,CH2*1
      INTEGER P1,P2,P3,P4,P5,P6,P7,P8,X1NR,X2NR,X3NR
      INTEGER X4NR,X5NR,SSNR,SKJNR,P22,P44,P66,P88
      EQUIVALENCE (KX1(1),X1(1)),(KX1(67),P1),(KX1(68),P2)
      EQUIVALENCE(KX2(1),X2(1)),(KX2(16),P3),(KX2(17),P4),(KX2(18),X1NR)
      EQUIVALENCE(KX3(1),X3(1)),(KX3(46),P5),(KX3(47),P6),(KX3(48),X2NR)
      EQUIVALENCE(KX4(1),X4(1)),(KX4(16),P7),(KX4(17),P8),(KX4(18),X3NR)
      EQUIVALENCE (KX5(1),X5(1)),(KX5(16),X4NR)
      OPEN(6,FILE = 'L-P-1',ACCESS = 'W')
      OPEN(10,FILE = 'STATUS',ACCESS = 'RW')
      OPEN(11,FILE='STASJON',ACCESS='WX',RECL =68)
      OPEN(12,FILE='PREDATOR',ACCESS = 'WX',RECL=18)
      OPEN(13,FILE = 'PSTRGR',ACCESS = 'WX',RECL=48)
      OPEN(14,FILE='BYTTEDYR',ACCESS = 'WX',RECL=18)
      OPEN(15,FILE = 'BSTRGR',ACCESS = 'WX',RECL=16)
C READ THE CONTENT IN "STATUS" TO SEE HOW MANY RECORDS THE FIVE
C STRUCTURED FILES CONTAINS.
      READ(10,100) P22,P44,P66,P88,SSNR,SKJNR,X1NR,X2NR,X3NR,X4NR,X5NR
  100 FORMAT(11I6)
      X5NR = X5NR + 1
      WRITE( 6,101)
  101 FORMAT('  P2    P4    P6    P8   SSNR SKJNR  X1NR  X2NR  X3NR  X4
     FNR  X5NR'//)
      WRITE(6,102) P22,P44,P66,P88,SSNR,SKJNR,X1NR,X2NR,X3NR,X4NR,X5NR
  102 FORMAT(11I6//)
      WRITE(6,103)
  103 FORMAT(/' INNHOLDET PÅ FILEN STASJON'/)
      DO 1 I = 0,SSNR,1
      CALL RFILE(11,0,KX1(1),I,68)
      ICH1 = X1(16)
      ICH2 = X1(17)
      CH1 = CHAR(ICH1)
      CH2 = CHAR(ICH2)
      WRITE(6,104)I,(X1(J),J = 1,15),CH1,CH2,(X1(J),J = 18,22),P1,P2
  104 FORMAT(I5,2F3.0,F4.0,3F3.0,2F5.0,F2.0,2F3.0,F4.0,F3.0,F2.0,F3.0,2A
     F,5F3.0,2I6)
    1 CONTINUE
      WRITE(6,105)
  105 FORMAT(//'  INNHOLDET PÅ FILEN PREDATOR'/)
      DO 2 I = 0,P22,1
      CALL RFILE(12,0,KX2(1),I,21)
      WRITE(6,106) I,(X2(J),J = 1,5),P3,P4,X1NR
  106 FORMAT(I5,5F3.0,3I6)
    2 CONTINUE
      WRITE(6,107)
  107 FORMAT(//'  INNHOLDET PÅ FILEN PSTRGR'//)
```

```
      DO 3 I = 0,P44 , 1
      CALL RFILE(13,0,KX3(1),I,48)
      WRITE(6,108)I,(X3(J),J = 1,15),P5,P6,X2NR
  108 FORMAT(I5,F8.1,2F5.1,F2.0,2F3.0,F2.0,F6.2,3F2.0,4F3.0,3I6)
    3 CONTINUE
      WRITE(6,109)
  109 FORMAT(//'  INNHOLDET PÅ FILEN BYTTEDYR'//)
      DO 4 I = 0,P66,1
      CALL RFILE(14,0,KX4(1),I,21)
      WRITE(6,106) I,(X4(J),J = 1,5),P7,P8,X3NR
    4 CONTINUE
      WRITE(6,110)
  110 FORMAT(//'  INNHOLDET PÅ FILEN BSTRGR'//)
      DO 5 I = 0,P88,1
      CALL RFILE(15,0,KX5(1),I,16)
      WRITE(6,111) I,(X5(J),J = 1,5),X4NR
  111 FORMAT( I5,F2.0,F6.0,F2.0,F9.3,F8.0,I6)
    5 CONTINUE
      STOP
      END
C** TAB-POLY-AM **
C
C************************************************************************
C                                                                       *
C      THE PROGRAM MAKES A SUMMARY TABLE OF THE PREYSPECIES AND          *
C      PREYSPECIES-SIZEGROUPS THAT A PREDATORSPECIES-SIZEGROUP           *
C      HAS EATEN. THE USER CHOOSES INTERACTIVELY :                       *
C                                                                       *
C                     - PREDATORSPECIES                                 *
C                     - PREDATORSPECIESSIZEGROUP                        *
C                     - TIMEPERIOD                                      *
C                     - RESOLUTION OF TAXONOMY OF PREY(5 LEVELS)        *
C                     - AREA TO BE INCLUDED                             *
C                                                                       *
C      THE AREA IS AN IRREGULARLY POLYGON WITH UP TO 12 VERTICES         *
C      THE ROUTINE "INSIDE" FROM "OCEANOGRAPHY EMR" IS USED              *
C                                                                       *
C    TROND WESTGÅRD , INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY       *
C                                                                       *
C************************************************************************
C
      PARAMETER MAXDIM = 1300
      REAL KV,LAB
      INTEGER P1,P2,P3,P4,P5,P6,P7,P8,X1NR,X2NR,X3NR
      INTEGER X4NR,X5NR,X1MAX,T1,T2,T3,PRED,TRMTYP,TLEV
      CHARACTER LATIN*59,LTIN*59,SIZEGR*12,PRSIZE*3,HISTO*3,CHAR*17
      DIMENSION X1(22),X2(5),X3(15),X4(5),X5(5),KX1(68),KX2(18)
      DIMENSION XH1(MAXDIM),XH2(3),XH3(8,MAXDIM),YH1(MAXDIM),KODE(5)
      DIMENSION KX3(48),KX4(18),KX5(16),DATO1(3),DATO2(3),PRED(5)
      DIMENSION PSTR(2),XP(12),YP(12),KV(12),M(2),KK(6),TLEV(5)
      EQUIVALENCE (KX1(1),X1(1)),(KX1(67),P1),(KX1(68),P2)
      EQUIVALENCE(KX2(1),X2(1)),(KX2(16),P3),(KX2(17),P4),(KX2(18),X1NR)
      EQUIVALENCE(KX3(1),X3(1)),(KX3(46),P5),(KX3(47),P6),(KX3(48),X2NR)
      EQUIVALENCE(KX4(1),X4(1)),(KX4(16),P7),(KX4(17),P8),(KX4(18),X3NR)
      EQUIVALENCE (KX5(1),X5(1)),(KX5(16),X4NR)
      COMMON /TRM/ TRMTYP
      DATA M/7,8/
      CALL MGTTY(0,TRMTYP)
      OPEN(10,FILE = 'STATUS',ACCESS = 'RW')
      OPEN(11,FILE='STASJON',ACCESS='WX',RECL =68)
      OPEN(12,FILE='PREDATOR',ACCESS = 'WX',RECL=18)
      OPEN(13,FILE = 'PSTRGR',ACCESS = 'WX',RECL=48)
```

```
      OPEN(14,FILE='BYTTEDYR',ACCESS = 'WX',RECL=18)
      OPEN(15,FILE = 'BSTRGR',ACCESS = 'WX',RECL=16)
      OPEN(17,FILE = 'HJELP2',ACCESS = 'RW')
      OPEN(18,FILE = 'PLOTDATA',ACCESS='RW')
      OPEN(21,FILE ='DYREKODE',ACCESS='WX',RECL=10)
      OPEN(22,FILE = 'DYRENAVN',ACCESS = 'RW')
      OPEN(6,FILE = 'L-P-1',ACCESS='W')
C
C ************ START INPUT SECTION ************************************
C
      CALL ERPAGE
      WRITE(1,99)
   99 FORMAT(/,2X,'1 PERIODE :',17X,'3 LENGDEGRUPPE :',6X,'7 HISTOGRAM :
     &',//,4X,'dd mm åå dd mm åå',11X,'lll.l lll.l',11X,'(ja/nei)',/,3X,
     &'>',27X,'>',21X,'>',//,2X,'2 POLYGON :',17X,'4 PREDATOR :',11X,
     &'8 KOMMANDO :',/,17X,'aa',/,3X,'ant. hjørner >',15X,'aabbccddee',
     &14X,'1 - 7 = retting',
     &/,14X,'k gg.mm ggg.mm',3X,'>',24X,'8 = kjøring av tabell'
     &,/,4X,'hjørne  1>',42X,'9 = slutt',/,4X,'hjørne  2>',41X,'>',/,
     &4X'hjørne  3>',16X,'5 BYTTEDYRSTØRRELSE :',/,4X,'hjørne  4>',/,4X,
     &'hjørne  5>',18X,'(ja/nei)',/,4X,'hjørne  6>',17X,'>',/,4X,'hjørne
     &  7>',/,4X,'hjørne  8>',16X,'6 ARTSNIVÅ :',/,4X,'hjørne  9>',/,4X,
     &'hjørne 10>',18X,'aabbccddee',/,4X,'hjørne 11>',17X,'>',/,4X,
     &'hjørne 12>',//)
      ST = 0.
C TIMEPERIOD :
   10 CALL ABSADR(5,20)
      CALL BACKSP(17)
      CALL RECHA(17,CHAR,IST)
      READ(CHAR,151) DATO1(3),DATO1(2),DATO1(1),DATO2(3),DATO2(2),
     & DATO2(1)
  151 FORMAT(F2.0,5(1X,F2.0))
      IF(ST .EQ. 1.) GOTO 17
C AREA :
   11 CALL ABSADR(9,18)
      CALL BACKSP(2)
      CALL RECHA(2,CHAR,IST)
      READ(CHAR,152) NHJ
  152 FORMAT(I2)
      DO FOR I = 1,12
         CALL ABSADR(I+10,27)
         CALL BACKSP(14)
      ENDDO
      DO FOR I = 1,NHJ
         CALL ABSADR(I+10,13)
         CALL RECHA(14,CHAR,IST)
         READ(CHAR,153) KV(I),YP(I),XP(I)
  153 FORMAT(I1,F6.2,F7.2)
      ENDDO
      IF(ST .EQ. 1.) GOTO 17
C PREDATOR SIZEGROUP:
   12 CALL ABSADR(5,42)
      CALL BACKSP(11)
      CALL RECHA(11,CHAR,IST)
      READ(CHAR,154) PSTR(1),PSTR(2)
  154 FORMAT(F5.1,F6.1)
      IF(ST .EQ. 1.) GOTO 17
C PREDATOR SPECIES :
   13 CALL ABSADR(10,41)
      CALL BACKSP(10)
      CALL RECHA(10,CHAR,IST)
      READ(CHAR,103)(PRED(I),I = 1,5)
  103 FORMAT(5I2)
      IF(ST .EQ. 1.) GOTO 17
```

```
      C SPLIT PREY IN SIZEGROUPS OR NOT :
         14 CALL ABSADR(16,34)
            CALL BACKSP(3)
            CALL RECHA(3,PRSIZE,IST)
            IF(ST .EQ. 1.) GOTO 17
      C TAXONOMIC LEVEL :
         15 CALL ABSADR(21,41)
            CALL BACKSP(10)
            CALL RECHA(10,CHAR,IST)
            READ(CHAR,103)(TLEV(I),I = 1,5)
            IF(ST .EQ. 1.) GOTO 17
      C DECISION OF HISTOGRAM PLOT :
         16 CALL ABSADR(5,56)
            CALL BACKSP(3)
            CALL RECHA(3,HISTO,IST)
            IF(ST .EQ. 1.) GOTO 17
            ST = 1.
         17 CALL ABSADR(12,56)
            CALL BACKSP(1)
            CALL RECHA(1,CHAR,IST)
            READ(CHAR,155) KOMM
        155 FORMAT(I1)
            CALL ABSADR(23,75)
            CALL BACKSP(24)
            GOTO(10,11,12,13,14,15,16,18,8),KOMM
      C
      C ************** STOP OF INPUT SECTION *********************************
      C
         18 IF(TLEV(5) .GT. 0 ) THEN
               IF(PRSIZE .EQ. 'JA') THEN
                  KK(6) = NS = 6
               ELSE
                  NS = 5
               ENDIF
               DO FOR J = 1,5
                  KK(J) = J
               ENDDO
            ELSEIF(TLEV(4) .GT. 0 ) THEN
               IF(PRSIZE .EQ. 'JA') THEN
                  NS = 5
                  KK(5) = 6
               ELSE
                  NS = 4
               ENDIF
               DO FOR J = 1,4
                  KK(J) = J
               ENDDO
            ELSEIF(TLEV(3) .GT. 0 ) THEN
               IF(PRSIZE .EQ. 'JA') THEN
                  NS = 4
                  KK(4) = 6
               ELSE
                  NS = 3
               ENDIF
               DO FOR J = 1,3
                  KK(J) = J
               ENDDO
            ELSEIF(TLEV(2) .GT. 0 ) THEN
               IF(PRSIZE .EQ. 'JA') THEN
                  NS = 3
                  KK(3) = 6
               ELSE
```

```
              NS = 2
           ENDIF
           KK(1) = 1
           KK(2) = 2
        ELSEIF(TLEV(1) .GT. 0) THEN
           IF(PRSIZE .EQ. 'JA') THEN
              NS = 2
              KK(2) = 6
           ELSE
              NS = 1
           ENDIF
           KK(1) = 1
        ELSEIF(PRSIZE .EQ. 'JA') THEN
           NS = 1
           KK(1) = 6
        ELSE
            CALL ABSADR(23,53)
            WRITE(1,300)
    300     FORMAT(1H$,'FEIL OPPLØSNINGSNIVÅ')
            GOTO 17
        ENDIF
        T1 = T2 = T3 = 0
C CHECKS EACH RECORD IN THE STATION-FILE ON DATA AND POSITION
C TO SEE IF IT SHOULD BE INCLUDED. STATUS-FILE IS FIRST READ
C TO SEE HOW MANY RECORDS THE STATION FILE CONTAINS
        REWIND 17
        REWIND 10
        READ(10,104) X1MAX
    104 FORMAT(24X,I6)
        DO 7 X1NR = 0,X1MAX
           CALL RFILE(11,0,KX1(1),X1NR,68)
C CHECKS THE DATES :
C FIRST WE CHECKS THE YEAR :
           IF(X1(4).LT.DATO1(1).OR.X1(4).GT.DATO2(1)) GOTO 7
           XDATO = X1(5) + X1(6)/100.
           X1DATO = DATO1(2) + DATO1(3)/100.
           X2DATO = DATO2(2) + DATO2(3)/100.
           IF(XDATO .LT.X1DATO .OR. XDATO .GT. X2DATO) GOTO 7
C CHECKS IF IT WAS TAKEN INSIDE THE AREA :
           IF(X1(9).EQ.0..AND.X1(10).EQ.0..AND.X1(11).EQ.0..AND.X1(12).EQ.
     &     0. .AND.X1(13).EQ.0.) GOTO 7
           CALL INS(X1(10),X1(11),X1(12),X1(13),X1(9),XP,YP,KV,NHJ,IND)
           IF(IND .EQ. 0) GOTO 7
           T1 = T1 + 1
C CHECKS IF IT WERE TAKEN PREDATORS ON THE STATION :
           IF(P1 .LT. 0.) GOTO 7
           DO 6 X2NR = P1,P2
              CALL RFILE(12,0,KX2(1),X2NR,18)
C CHECKS IF THE ACTUAL PREDATORSPECIES WAS TAKEN IN THE AREA :
              DO FOR I = 1,5
                 IF(X2(I) .NE. PRED(I)) GOTO 6
              ENDDO
C NOW WE CHECKS THE PREDATORSPECIES-SIZEGROUP :
              DO 5 X3NR = P3,P4
                 CALL RFILE(13,0,KX3(1),X3NR,48)
C WHEN PREDATORSPECIES-SIZEGROUP WAS PRESENT THE DATA ARE SAVED ON
C HJELP2. IF NOT A NEW PREDATORSPEC.-SIZEGROUP IS READ. WE MUST CHECK IF
C IT WAS A SINGLE FISH WITH INDIVIDUAL LENGTH MEASUREMENT.
                 IF(X3(3) .EQ. 0.) GOTO 1
                 IF(PSTR(1) .LE. X3(2) .AND. PSTR(2) .GE. X3(3)) GOTO 2
                 GOTO 5
      1          IF(X3(2) .GE. PSTR(1).AND.X3(2).LE.PSTR(2)) GOTO 2
                 GOTO 5
      2          T2 = T2 + 1
```

```
                    WRITE(17) X3(12),X3(13),X3(14)
C IF THE PRED.SPEC.SIZEGROUP HAD EMPTY STOMACHS A NEW PRED.SPEC.SIZE-
C GROUP-RECORD IS READ.
                    IF(P5 .LT. 0) GOTO 5
                    DO 4 X4NR = P5,P6
                       CALL RFILE(14,0,KX4(1),X4NR,18)
                       DO 3 X5NR = P7,P8
                          CALL RFILE(15,0,KX5(1),X5NR,16)
C WE MUST CORRECT THE STOMACH CONTENT IF SOME FISH HAD REGURGITATED
                          IF(X3(12) .NE. 0. .AND. X3(13) .NE. 0.) THEN
                             DO FOR I = 4,5
                                X5(I) = (1. + (X3(13)/X3(12)))*X5(I)
                             ENDDO
                          ENDIF
                          IF(T3 .EQ. MAXDIM) THEN
                             CALL SORTX(KK,NS,0,N1,XH3,8,T3)
                             CALL SAME(KK,NS,M,2,XH3,8,T3)
                             IF((MAXDIM - T3) .LT. 1) THEN
                                CALL ABSADR(23,53)
                                WRITE(1,302)
  302                           FORMAT(1H$,'MAXDIM FOR LITEN      ')
                                GOTO 17
                             ENDIF
                          ENDIF
                          T3 = T3 + 1
                          DO FOR I = 1,5
                             XH3(I,T3) = X4(I)
                          ENDDO
C THE TWO NEXT LINES MUST BE CORRECTED WHEN MORE SCALES ARE CREATED:
                          XH3(6,T3) = X5(2)
                          XH3(7,T3) = X5(4)
                          XH3(8,T3) = X5(5)
    3                  CONTINUE
    4               CONTINUE
    5            CONTINUE
    6         CONTINUE
    7 CONTINUE
C
C WRITES THE TABLE-HEADING :
      WRITE(6,200)
  200 FORMAT(1H1,'  INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY.'/
     &'  PROJECT: BESTANDSBEREGNINGER MED FLERARTSMODELLER.'//)
      CALL NAME(PRED,LATIN,1)
      WRITE(6,201) (PRED(I),I=1,5),LATIN
  201 FORMAT('  SPECIES: '5J2 '  'A,//)
      WRITE(6,202)DATO1(3),DATO1(2),DATO2(3),DATO2(2),DATO1(1),DATO2(1),
     &     PSTR(1),PSTR(2)
  202 FORMAT('  TIMEPERIOD:'I3'/'I2' -'I3'/'I2' FOR THE YEARS:'I3' -'I3,
     &//,'  SIZEGROUP(CM):'F6.1' -'F6.1,//)
      WRITE(6,203) ((YP(I),XP(I),KV(I)),I=1,NHJ)
  203 FORMAT('  VERTICES OF SAMPLED AREA :',/,20(2F10.2,F3.0,/))
      WRITE(6,204) T1
  204 FORMAT('  NUMBER OF STATIONS SAMPLED IN GIVEN',/,
     &'  AREA AND TIME PERIOD'17X':',I4,/)
C COMPUTES TOTAL NUMBER OF STOMACHS :
C READS DATA FROM HJEL2-FILE
      XNM = 0.
      XNT = 0.
      XNS = 0.
      PRT = 0.
      REWIND 17
      DO FOR K = 1,T2
```

```
          READ(17)(XH2(I),I = 1,3)
C TOTAL NUMBER OF STOMACHS ,XNM :
          XNM = XNM + XH2(1) + XH2(2) + XH2(3)
C NUMBER OF STOMACHS EMPTY,XNT :
          XNT = XNT + XH2(3)
C NUMBER OF STOMACHS REGURGITATED, XNS :
          XNS = XNS + XH2(2)
      ENDDO
      WRITE(6,205) XNM
  205 FORMAT('  TOTAL NUMBER OF STOMACHS',13X,':'I4,/)
      WRITE(6,206) XNT
  206 FORMAT('  TOTAL NUMBER OF STOMACHS EMPTY',7X,':'I4,/)
      WRITE(6,213) XNS
  213 FORMAT('  TOTAL NUMBER OF STOMACHS REGURGITATED:'I4,/)
      IF(XNM .EQ. 0.) THEN
         PRT = -9.
      ELSE
         PRT = (XNT/XNM)*100.
      ENDIF
      WRITE(6,207) PRT
  207 FORMAT('  PERCENTAGE OF STOMACHS EMPTY',9X,':',F5.1,/)
      WRITE(6,208)(TLEV(J),J = 1,5)
  208 FORMAT('  TAXONOMIC LEVEL : ',5J2)
C BLANKS ALL COLUMNS IN XH3 ACCORDING TO THE CHOSEN TAXONOMIC LEVEL :
      IF(TLEV(5) .EQ. 0) THEN
         DO FOR J = 1,T3
            XH3(5,J) = 0.
         ENDDO
      ENDIF
      IF(TLEV(4) .EQ. 0) THEN
         DO FOR J = 1,T3
            XH3(4,J) = 0.
         ENDDO
      ENDIF
      IF(TLEV(3) .EQ. 0) THEN
         DO FOR J = 1,T3
            XH3(3,J) = 0.
         ENDDO
      ENDIF
      IF(TLEV(2) .EQ. 0) THEN
         DO FOR J = 1,T3
            XH3(2,J) = 0.
         ENDDO
      ENDIF
C CREATES HEADING FOR THE SECOND PART OF THE TABLE :
      WRITE(6,209)
  209 FORMAT(1H1,2X'NUMBER'9X'TAXON'21X'SIZE'7X'WEIGHT'3X'WEIGHT'1X'NUMB
     &ER'1X'NUMBER'2X'WEIGHT'2X,/,3X'OF'49X'GRAMS'12X'PER'12X'PER'4X,/,
     &1X'CATEGORIES'33X'GROUP'5X'PER PRED.'2X'%'4X'PRED.'4X'%'4X'PREY IN
     &D.',/)
C SORTS XH3(I,J) WITH INCREASING PREYSPECIESCODE
C AND WITH INCREASING SIZEGROUPS FOR EACH PREYSPECIESCODE :
      CALL SORTX(KK,NS,0,N1,XH3,8,T3)
C COMBINES ALL LINES IN XH3(I,J) WHERE PREYSPECIESCODE AND SIZE-GROUP
C OF THAT PREYSPECIESCODE ARE EQUAL :
      CALL SAME(KK,NS,M,2,XH3,8,T3)
C COMPUTES TOTAL NUMBER AND TOTAL WEIGHT IN ALL CATEGORIES :
      XW = 0.
      XN = 0.
      IF(HISTO .EQ. 'JA') WRITE(18) T3
      DO FOR J = 1,T3
         XW = XW + XH3(7,J)
         XN = XN + XH3(8,J)
      ENDDO
```

```
          IF(XW .EQ. 0. .OR. XN .EQ. 0.) THEN
             WRITE(6,211)
  211        FORMAT('  NOT POSSIBLE TO MAKE THE TABLE'/)
             CALL ABSADR(23,53)
             WRITE(1,301)
  301        FORMAT(1H$,'TABELL FERDIG         ')
             GOTO 17
          ENDIF
C PRINTS OUT THE RESULTS ON THE LINE-PRINTER :
          DO FOR J = 1,T3
             XPW = (XH3(7,J)/XW)*100.
             XPN = (XH3(8,J)/XN)*100.
             IF(XH3(8,J) .EQ. 0.) THEN
                XVPN = -9.
                XH3(8,J) = -9.
             ELSE
                XVPN = XH3(7,J)/XH3(8,J)
             ENDIF
             IF(XNM .EQ. 0.) THEN
                XNPF = -9.
                XWPF = -9.
             ELSE
                XNPF = XH3(8,J)/XNM
                XWPF = XH3(7,J)/XNM
             ENDIF
C FINDS THE LATIN-NAME THAT BELONGS TO THE SPECIESCODE
C IF A TAXONOMIC LEVEL IS SPECIFIED :
             IF(TLEV(1) .EQ. 0) THEN
                DO FOR I = 1,59
                   LATIN(I:I) = ' '
                ENDDO
                GOTO 19
             ENDIF
             DO FOR K = 1,5
                KODE(K) = XH3(K,J)
             ENDDO
             CALL NAME(KODE,LATIN,1)
             IF(J.EQ.1) THEN
                LTIN = LATIN
             ELSEIF(LATIN .EQ. LTIN) THEN
                DO FOR I = 1,59
                   LATIN(I:I) = ' '
                ENDDO
             ELSE
                LTIN = LATIN
             ENDIF
   19        IF( PRSIZE .EQ. 'JA') THEN
                CALL SIZE(XH3(6,J),SIZEGR)
             ELSE
                SIZEGR = '           '
             ENDIF
             WRITE(6,210) J,LATIN,SIZEGR,XWPF,XPW,XNPF,XPN,XVPN
  210 FORMAT(I6,5X,A29,A12,F10.2,F6.1,F7.2,F7.1,F10.2)
             IF(HISTO .EQ. 'JA') THEN
                IF(XPW .LT. 0.) XPW = 0.
                WRITE(18) XPW
             ENDIF
          ENDDO
          WRITE(6,212)
  212 FORMAT(//'  NOTE : WHEN A VARIABLE IS NEGATIVE IT IS NOT POSSIBLE
     &TO COMPUTE'/)
          CALL ABSADR(23,53)
          WRITE(1,301)
          GOTO 17
    8 CALL ERPAGE
          STOP ' GOODBYE, YOU WILL GET A PRINTOUT ON THE LINE-PRINTER'
          END
```

```
C ** SORTX-AM **
C
C**********************************************************************
C                                                                     *
C       ROUTINE TO SORT A REAL-ARRAY, X(I,J). ONE MAY CHOOSE THE      *
C       SIGNIFICANCE OF EACH VARIABLE(COLUMN) IN THE ARRAY ARBITRARILY *
C                                                                     *
C    GUNNAR HELLE, INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY       *
C    MODIFIED BY T. WESTGARD 23/6-81                                  *
C**********************************************************************
      SUBROUTINE SORTX(KK,NS,N,N1,X,L,L1)
C
C EXPLANATION OF THE PARAMETERS :
C
C KK(J)        = NUMBER OF THE VARIABLES(COLUMNS) THE ARRAY SHOULD BE
C                SORTED AFTER. KK(1) MOST KK(NS) LEAST SIGNIFICANT.
C NS           = NUMBER OF COLUMNS TO BE SORTED AFTER . MAX. VALUE IS 10
C N            = THE VARIBLE TELLS WHETHER TO SORT IN INCREASING( = 0)
C                OR DECREASING( = -1) ORDER.
C N1           = TELLS HOW MANY CHANGES OF ORDER THAT ARE MADE
C X(L,L1)      = ARRAY TO BE SORTED.
C L            = NUMBER OF COLUMNS IN X
C L1           = NUMBER OF LINES IN X
C
      DIMENSION X(L,L1),K(10),KK(NS)
      DO FOR JX = 1,10
         K(JX) = 0
      ENDDO
      N1=0
      IF(L1-1)13,13,1
    1 DO FOR JX = 1,NS
         K(JX) = KK(JX)
      ENDDO
      I=1
    2 I=I*2
      IF(I.LE.L1)GO TO 2
      M1=(I-1)/2
      DO 12 M=1,M1,2
      M2=M1-M+1
      M3=L1-M2
      DO 11 J=1,M3
      DO 10 I=1,J,M2
      I1=J-I+1
      DO 7 NN=1,10
      II=K(NN)
      IF(II)7,7,3
    3 IF(II-L)4,4,7
    4 Y=X(II,I1+M2)-X(II,I1)
      IF(N)5,6,6
    5 Y=-Y
    6 IF(Y)8,7,11
    7 CONTINUE
      GO TO 11
    8 DO 9 NN=1,L
      Y=X(NN,I1+M2)
      X(NN,I1+M2)=X(NN,I1)
      X(NN,I1)=Y
    9 CONTINUE
      N1=N1+1
   10 CONTINUE
   11 CONTINUE
   12 CONTINUE
   13 RETURN
      END
```

```
C ** NAVN-TRANS-AM **
C
C************************************************************************
C                                                                       *
C       THE FILE "QEDDYRENAVN" CONTAINS THE RECORDS :                   *
C                                                                       *
C                    1. 10-DIGIT SPECIES NODC-CODE                      *
C                    2. LATIN NAME                                      *
C                    3. NORWEGIAN NAME                                  *
C                    4. ENGLISH NAME                                    *
C                                                                       *
C       THE SPECIESCODES ARE STORED IN INCREASING ORDER.                *
C       FROM THIS FILE TWO FILES ARE PRODUCED :                         *
C                                                                       *
C                    A. A TABLE WITH THE SPECIESCODES(SORTED)           *
C                       AND A BYTEPOINTER TO FILE B. AND THE            *
C                       LENGTH OF EACH NAME IN NR. OF BYTES.            *
C                    B. THE ACTUAL NAMES STORED IN BYTES(ASC-II)        *
C                                                                       *
C     TRUND WESTGÅRD, INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY      *
C                                                                       *
C************************************************************************
      INTEGER ELEN
      DOUBLE INTEGER IBYTE,N
      DIMENSION K(10),KODE(5)
      CHARACTER LATIN*59,NORSK*31,ENG*31
      EQUIVALENCE (K(1),KODE(1)),(K(6),IBYTE)
      EQUIVALENCE (K(8),LLEN),(K(9),NLEN),(K(10),ELEN)
      OPEN(20,FILE = 'QEDDYRENAVN',ACCESS = 'RW')
      OPEN(21,FILE = 'DYREKODE',ACCESS = 'WX',RECL = 10)
      OPEN(22,FILE = 'DYRENAVN',ACCESS = 'RW')
      OPEN(6,FILE ='L-P-1',ACCESS = 'W')
      IBYTE = 0
      IREC = -1
      N = 0
      DO FOR I = 0,10000,1
      N = N + 1
      READ(20,100,END = 1,ERR = 2)(KODE(J),J = 1,5)
  100 FORMAT(5I2)
      N = N + 1
      READ(20,101,ERR = 2) LATIN
  101 FORMAT(A)
      N = N + 1
      READ(20,101,ERR = 2) NORSK
      N = N + 1
      READ(20,101,ERR = 2) ENG
      CALL STRSEG(59,LATIN,LLEN)
      CALL STRSEG(31,NORSK,NLEN)
      CALL STRSEG(31,ENG,ELEN)
      CALL WFILE(21,0,K(1),I,10)
      IREC = IREC + 1
      WRITE(22) (LATIN(J:J),J = 1,LLEN,1)
      WRITE(22) (NORSK(J:J),J = 1,NLEN,1)
      WRITE(22) (ENG(J:J),J = 1,ELEN,1)
      IBYTE = IBYTE + LLEN + NLEN + ELEN
      ENDDO
    1 WRITE(1,102) IREC
  102 FORMAT(' ANTALL KODER : ',I6)
      GOTO 3
    2 WRITE(1,103) N
  103 FORMAT(' ERROR WHEN READING LINE :',I6)
    3 STOP
      END
```

```
C ** STRSEG-AM **
C
C*************************************************************************
C                                                                        *
C       ROUTINE TO FIND LENGHT OF NONE-BLANK FIELD IN A                  *
C       CHARACTERFIELD.                                                  *
C                                                                        *
C     AAGE FOTLAND, INSTITUTE OF MARINE RESEARCH, BERGEN,NORWAY          *
C                                                                        *
C*************************************************************************
C
      SUBROUTINE STRSEG(M,STRING,LENGTH)
      CHARACTER STRING
C     STRING = STRING(-1:-1)
      DO FOR I = M,1,-1
      IF(STRING(I:I) .NE. ' ') THEN
      LENGTH = I
      RETURN
      ENDIF
      ENDDO
      LENGTH = 0
C     LENGTH = -1
      RETURN
      END
```

```
C ** NAVN-AM **
C
C*************************************************************************
C                                                                        *
C       THE ROUTINE IS A BINARY-SEARCH ROUTINE THAT FIRST SEEKS          *
C       THE CODE GIVEN IN FILE A. CREATED IN "NAVN-TRANS-AM"             *
C       THEN THE ACTUAL NAME IS READ IN FILE B. IF IT EXIST.             *
C       IF THE CODE OR THE NAME IS NOT FOUND AN ERROR MESSAGE            *
C       IS GIVEN ON THE SCREEN AND THE ACTUAL CODE IS RETURNED           *
C       AS THE NAME TO THE CALLING PROGRAM.                              *
C                                                                        *
C     TROND WESTGARD, INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY       *
C                                                                        *
C*************************************************************************
C
C    KODE = 10-DIGIT SPECIESCODE.
C    NAVN = CHARACTERVARIABLE THAT RETURNS THE NAME IF CODE AND NAME IS
C           FOUND, IF NOT THE CODE IS RETURNED AS THE NAME.
C    TYPE = TYPE OF NAME THE USER WANT.
C                   1 = LATIN NAME
C                   2 = NORWEGIAN NAME
C                   3 = ENGLISH NAME
C
      SUBROUTINE NAME(KODE,NAVN,TYPE)
      DOUBLE INTEGER IBYTE
      INTEGER TYPE,HI
      DIMENSION K(10),KODE(5),KO(5),KOO(10)
      CHARACTER NAVN*59
      EQUIVALENCE (K(6),IBYTE)
      DO FOR I = 1,59
      NAVN(I:I) = ' '
      ENDDO
      LO = -1
C HI IS THE NUMBER OF SPECIES-CODES GIVEN FROM NAVN-TRANS-AM PLUS 1.
      HI = 1225
    1 MID = (LO + HI)/2 + MOD((LO+HI),2)
      CALL RFILE(21,0,K(1),MID,10)
      DO FOR I = 1,5
      IF(KODE(I) .GT. K(I)) GOTO 2
      IF(KODE(I) .LT. K(I)) GOTO 3
      ENDDO
      GOTO 5
```

```
      2 LO = MID
        GOTO 4
      3 HI = MID
      4 IF((HI - LO) .LT. 2) GOTO 9
        GOTO 1
C CODE FOUND :
      5 IF(TYPE - 2) 6,7,8
      6 IF(K(8).EQ. 0) GOTO 10
        CALL SETBT(22,IBYTE)
        READ(22) (NAVN(I:I),I = 1,K(8))
        RETURN
      7 IF(K(9) .EQ. 0) GOTO 10
        CALL SETBT(22,(IBYTE+K(8)))
        READ(22) (NAVN(I:I),I = 1,K(9))
        RETURN
      8 IF(K(10) .EQ. 0) GOTO 10
        CALL SETBT(22,(IBYTE+K(8)+K(9)))
        READ(22) (NAVN(I:I),I = 1,K(10))
        RETURN
C CODE OR NAME NOT FOUND :
      9 GOTO 11
     10 WRITE(6,101)(KODE(I),I = 1,5),TYPE
    101 FORMAT(' KODEN ',5I2,' HAR IKKE NOE TYPE',I2,' NAVN')
     11 DO FOR I = 1,5
        KO(I) = KODE(I)
        ENDDO
        DO FOR I = 1,5
        J = 2*I - 1
        KOO(J) = KO(I)/10
        KOO(J+1) = MOD(KO(I),10)
        NAVN(J:J) = CHAR(KOO(J)+48)
        NAVN(J+1:J+1) = CHAR(KOO(J+1)+48)
        ENDDO
        RETURN
        END
```

```
C ** SAME-AM **
C
C**********************************************************************
C                                                                     *
C       ROUTINE THAT COMBINES LINES IN AN ARRAY WHERE SPECIFIED       *
C       COLUMNS HAS EQUAL VALUES. THE LINES THAT ARE COMBINED WILL    *
C       GET THE VALUES OF OTHER SPECIFIED COLUMNS IN THE LINES THAT ARE *
C       COMBINED SUMMED.                                              *
C                                                                     *
C     TROND WESTGÅRD, INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY    *
C                                                                     *
C**********************************************************************
C
      SUBROUTINE SAME(N,NI,M,MI,X,IX,JX)
C
C N  = ARRAY THAT THE NUMBER OF THE COLUMNS THAT SHOULD BE
C      CHECKED FOR EQUALNESS.
C NI = THE NUMBER OF COLUMNS TO BE CHECKED FOR EQUALNESS.
C M  = ARRAY THAT TELLS WHICH COLUMNS WHICH ARE TO BE SUMMED IN THE
C      LINES THAT ARE COMBINED.
C MI = THE NUMBER OF COLUMNS WHERE VALUES SHOULD BE SUMMED.
C X  = ARRAY TO BE TREATED
C IX = NUMBER OF COLUMNS IN X
C JX = NUMBER OF LINES IN X
C
      DIMENSION N(1),M(1),X(IX,JX)
      IT = J = 0
    1 J = J + 1
    2 IF(J .GT. (JX - IT)) GOTO 4
```

```
      DO FOR I = 1,NI
      I1 = N(I)
      IF(X(I1,J+1) .NE. X(I1,J)) GOTO 1
      ENDDO
      DO FOR I = 1,MI,1
      I1 = M(I)
      X(I1,J) = X(I1,J) + X(I1,J+1)
      ENDDO
      IF((J+2) .GT. JX) GOTO 3
      DO FOR J1 = J+1,JX-IT
      DO FOR I1 = 1,IX
      X(I1,J1) = X(I1,J1+1)
      ENDDO
      ENDDO
    3 IT = IT + 1
      GOTO 2
    4 JX = JX - IT
      RETURN
      END
C ** INS-AM **
C
C*********************************************************************
C                                                                    *
C      ROUTINE THAT CHECKS IF A GEOGRAPHICAL POSITION IS OUTSIDE     *
C      OR INSIDE A POLYGON WITH N VERTICES. THE INSIDE ROUTINE FROM  *
C      "OCEANOGRAPHY EMR". IS USED.                                  *
C                                                                    *
C     TROND WESTGÅRD, INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY   *
C                                                                    *
C*********************************************************************
C
      SUBROUTINE INS(BG,BM,LG,LM,KVAD,X,Y,KV,N,IND)
C
C BG = LATITUDE(DEGREES)
C BM = LATITUDE(MINUTES)
C LG = LONGITUDE(DEGREES)
C LM = LONGITUDE(MINUTES)
C KVAD = QUADRANT THE POSITION BELONGS TO .
C        0(N AND E), 1(N AND W), 2(S AND E), 3(S AND W)
C X(I),Y(I) = VERTICES OF THE POLYGON (E.G.: 54.13,3.04)
C N = NUMBER OF VERTICES IN THE POLYGON
C KV(I) = QUADRANT THE VERTICES IS IN
C IND = INDICATOR TO TELL WHETHER THE POSITION IS OUTSIDE(0) OR
C       INSIDE(1) THE POLYGON.
C
      DIMENSION X(N),Y(N),KV(N),X1(20),Y1(20),KV1(20)
      REAL BG,BM,LG,LM,KVAD,KV,M,B,L,KV1
C "INTERNAL ARRAY" IS PUT EQUAL TO INPUT ARRAY :
      DO FOR I = 1,N
      X1(I) = X(I)
      Y1(I) = Y(I)
      KV1(I) = KV(I)
      ENDDO
C CONVERTS THE POSITION TO ONE REAL NUMBER :
      B = BG + (BM/100.)
      L = LG + (LM/100.)
C SOUTHERLY LATITUDE NEGATIV, WESTHERLY LONGITUDE NEGATIVE :
      IF(KVAD .GT. 1.)  B = -B
      IF(KVAD .EQ. 1. .OR. KVAD .EQ. 3.) L = -L
      DO FOR I = 1,N
      IF(KV1(I) .GT. 1.) Y1(I) = -Y1(I)
      IF(KV1(I) .EQ. 1. .OR. KV1(I) .EQ. 3.) X1(I) = -X1(I)
      ENDDO
      IND = 999
      CALL INSIDE(L,B,X1,Y1,N,IND)
      RETURN
      END
```

```
C ** SIZE-AM **
C
C*********************************************************************
C                                                                   *
C  ROUTINE TO FIND A SIZEGROUP WHEN A SIZECODE IS GIVEN             *
C                                                                   *
C TROND WESTGARD, INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY      *
C                                                                   *
C*********************************************************************
C
      SUBROUTINE SIZE(CODE,SIZEGR)
      DIMENSION SI(37)
      INTEGER HI
      CHARACTER C(37)*12,SIZEGR*12
      DATA SI/-1.,0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,15.,20.,25.,30.,
     > 40.,50.,70.,100.,150.,200.,250.,300.,400.,500.,700.,1000.,
     > 1500.,2000.,2500.,3000.,4000.,5000.,7000.,10000.,99999./
      DATA C/'     EGG     ','     NAUPLII   ',
     >                 '  .01-.019CM','  .02-.029CM','  .03-.039CM
     >',         '  .04-.049CM','  .05-.059CM','  .06-.069CM','  .07-.079CM
     >',         '  .08-.089CM','  .09-.099CM','  .1 -.14  CM','  .15-.19 CM
     >',         '  .2 -.24  CM','  .25-.29 CM','  .3 -.39  CM','  .4 -.49  CM
     >',         '  .5 -.69  CM','  .7 -.99  CM','  1. -1.4  CM','  1.5-1.9  CM
     >',         '  2. -2.4  CM','  2.5-2.9  CM','  3. -3.9  CM','  4. -4.9  CM
     >',         '  5. -6.9  CM','  7. -9.9  CM',' 10. -14.  CM',' 15. -19.  CM
     >',         ' 20. -24.  CM',' 25. -29.  CM',' 30. -39.  CM',' 40. -49.  CM
     >',         ' 50. -69.  CM',' 70. -99.  CM',' 100.-149.CM','  UNKNOWN
     >'/
      LO = 0
      HI = 38
    1 MID = (LO+HI)/2 + MOD((LO+HI),2)
      IF( CODE .GT. SI(MID)) GOTO 2
      IF( CODE .LT. SI(MID)) GOTO 3
      GOTO 5
    2 LO = MID
      GOTO 4
    3 HI = MID
    4 IF((HI - LO) .LT. 2) GOTO 6
      GOTO 1
    5 SIZEGR = C(MID)
      GOTO 7
    6 SIZEGR = '   *****    '
    7 RETURN
      END
```

7. Sample input/output

The communication with TAB-POLY-AM is done on one "page" on the screen and the user has always by a quick glance on the screen the full overview over the last conditions given to the program. From the command position in field 8 the user is free to change any of the seven fields that gives the input to the program or to produce a table. The actual screen page is given in Fig. 2.

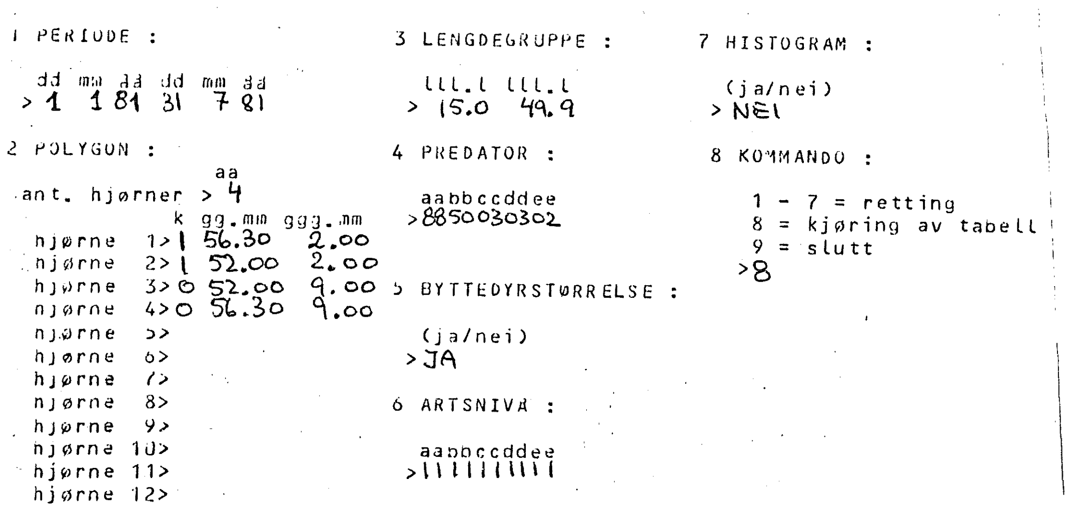

Fig. 2. The user communication for the program TAB-POLY-AM.

A sample output is shown in Table 1. The data is from mackerel stomachs analysed by Sigbjørn Mehl at the Institute of Marine Research, Bergen.

8. Implementation

The programs has been implemented on the ND-100 computer at the Institute of Marine Research, Bergen, Norway. This computer has a 16-bits wordlength and one real number takes 3 words, one integer 1 word and one ASC-II character (byte) 0.5 words. Therefore it is two bytes in one integer. The screenhandling routines uses the monitor-calls available on ND-100 and are

INSTITUTE OF MARINE RESEARCH, BERGEN, NORWAY.
PROJECT: BESTANDSBEREGNINGER MED FLERARTSMODELLER.

SPECIES: 8850030302 SCOMBER SCOMBRUS

TIMEPERIOD: 1/ 1 - 31/ 7 FOR THE YEARS: 81 - 81

SIZEGROUP(CM): 15.0 - 49.9

VERTICES OF SAMPLED AREA :

56.30	2.00	1.
52.00	2.00	1.
52.00	9.00	0.
56.30	9.00	0.

NUMBER OF STATIONS SAMPLED IN GIVEN AREA AND TIME PERIOD : 26

TOTAL NUMBER OF STOMACHS : 275

TOTAL NUMBER OF STOMACHS EMPTY : 21

TOTAL NUMBER OF STOMACHS REGURGITATED: 2

PERCENTAGE OF STOMACHS EMPTY : 7.6

TAXONOMIC LEVEL : 1111111111

NUMBER OF CATEGORIES	TAXON	SIZE GROUP	WEIGHT GRAMS PER PRED.	WEIGHT %	NUMBER PER PRED.	NUMBER %	WEIGHT PER PREY IND.
1	PHAEOPHYCEAE FUCALES	UNKNOWN	0.00	0.0	0.00	0.0	0.01
2	PHYSOPHORA HYDROSTATICA	UNKNOWN	0.00	0.0	0.02	0.0	0.12
3	NEREIS PELAGICA	7. -9.9 CM	0.02	0.4	0.00	0.0	6.20
4		UNKNOWN	0.00	0.0	0.00	0.0	0.07
5	ALLOTEUTHIS SUBULATA	UNKNOWN	0.00	0.0	0.00	0.0	0.12
6	COPEPODA	.1 -.14 CM	0.01	0.3	58.29	26.6	0.00
7		.3 -.39 CM	0.00	0.0	0.00	0.0	0.00
8	COPEPODA CALANOIDA	.2 -.24 CM	0.04	0.7	72.73	33.2	0.00
9	CALANUS FINMARCHICUS	.3 -.39 CM	0.04	0.8	43.27	19.7	0.00
10	TEMORIDAE	.1 -.14 CM	0.01	0.1	25.45	11.6	0.00
11	MYSIDA	.7 -.99 CM	0.00	0.0	0.00	0.0	0.00
12	AMPHIPODA	.5 -.69 CM	0.00	0.0	0.00	0.0	0.01
13	PARATHEMISTO	.7 -.99 CM	0.00	0.0	0.00	0.0	0.02
14		1. -1.4 CM	0.00	0.0	0.00	0.0	0.02
15	EUPHAUSIIDAE	1. -1.4 CM	0.01	0.1	0.48	0.2	0.01
16		1.5-1.9 CM	0.00	0.1	0.13	0.1	0.03
17		UNKNOWN	0.01	0.2	0.10	0.0	0.11
18	MEGANYCTIPHANES NORVEGICA	1.5-1.9 CM	0.00	0.0	0.01	0.0	0.04
19		2. -2.4 CM	0.01	0.2	0.14	0.1	0.07
20		2.5-2.9 CM	1.27	22.7	9.16	4.2	0.14
21		3. -3.9 CM	0.85	15.2	4.57	2.1	0.19
22		UNKNOWN	0.39	6.9	3.27	1.5	0.12
23	OPHIUROIDEA	UNKNOWN	0.00	0.1	0.01	0.0	0.39
24	TELEOSTEI	UNKNOWN	1.18	21.0	0.41	0.2	2.87
25	AMMODYTIDAE	2. -2.4 CM	0.00	0.0	0.01	0.0	0.02
26		3. -3.9 CM	0.02	0.4	0.11	0.0	0.23
27		4. -4.9 CM	0.08	1.5	0.24	0.1	0.34
28		5. -6.9 CM	0.22	3.9	0.28	0.1	0.80
29		7. -9.9 CM	0.25	4.5	0.13	0.1	1.93
30		UNKNOWN	0.32	5.6	0.17	0.1	1.82
31	HYPEROPLUS LANCEOLATUS	7. -9.9 CM	0.08	1.5	0.02	0.0	3.87
32		10. -14. CM	0.60	10.8	0.10	0.0	5.92
33		15. -19. CM	0.17	3.0	0.01	0.0	15.47
34	AVES	UNKNOWN	0.00	0.0	0.00	0.0	0.01
35	9999999999	UNKNOWN	0.00	0.0	-0.03	0.0	-9.00

NOTE : WHEN A VARIABLE IS NEGATIVE IT IS NOT POSSIBLE TO COMPUTE

Table 1. Sample output from the program TAB-POLY-AM. Data from mackerel in the North Sea.

therefore computer specific. With this and differences in the FORTRAN syntax on different computers in mind, it should be possible to implement the programs on other computers.

9. Acknowledgements

I want to thank my colleagues at the Institute of Marine Research for invaluable assistance. I am especially indebted to Gunnar Helle for his help and advice. Per G. Fadnes and Aage Fotland have also helped in many ways.