

SPECIAL REPORT NO. 9

**Documentation of a Computer Program for
Multispecies Fish Populations**

by

WALTER J. HOGMAN

**Center for Great Lakes Studies
University of Wisconsin-Milwaukee
Milwaukee, Wisconsin 53201**

November 1969

TABLE OF CONTENTS

	Page
INTRODUCTION	1
COMPUTATIONS MADE	2
USE OF THE PROGRAM	3
DESCRIPTION OF ROUTINES AND VARIABLES	7
Subroutine SETZR	
Subroutine READER	
Subroutine MAGIC	
Subroutine FRQDIS	
Subroutine JUNE	
Subroutine REGRES	
Subroutine PRINT	
Subroutine BAKCAL	
Subroutine WTCAL	
ACKNOWLEDGEMENTS	14
MASTER FLOW CHART	15
APPENDIX	17

Documentation of a Computer Program for
Multispecies Fish Populations

This program performs the more common calculations of parameters of fish populations. It is designed with a simple, yet modern, digital computer logic. The central idea was to provide a program that can be used by a biologist with no computer background and at the same time keep the design general to allow for maximum flexibility.

These two objectives have been satisfied by making the entire program from subroutines, each of which perform a particular function; and by making all variables conform in name to the data they represent. A minimum of card handling has been achieved by building in looping procedures to handle multispecies problems in one run.

A diagnostic trace (independent of the computer error messages) has been built in which traces the flow of the program and prints values as they are computed. This allows the investigator to check that the data entered properly, were manipulated correctly, and to see where the program stopped if it stops after he has made a change within a subroutine.

To most effectively utilize the program, the user may wish to alter many of the arrays. This requires enough programming ability to understand the concept of array lengths, range of the DO loops, and how to change various class intervals and test statements. Methods of such internal manipulation are given following each variable described, or are described in the listing in the appendix. If changes are required in other routines to accommodate the initial change, these are indicated only once, after the description of the variable. Only the less obvious changes are explained.

COMPUTATIONS MADE

For each species or group the following calculations are made, the output of which is in three basic units.

Part I: Sexes are combined.

- a. Name of species.
- b. Average total length in millimeters.
- c. Average weight to hundredths of a kilogram.
- d. Percentage of females.
- e. Total number of this species or group.
- f. Average age (completed annuli).
- g. Coefficients of the body-scale equation.
- h. Coefficient of correlation of the body-scale data.
- i. Coefficients of the length-weight equation.
- j. Log of constant in the length-weight equation.

Part II: The calculated arrays are listed and plotted. The graph used is a normal X,Y plot available on most computer library files. A list of the graph is provided, so the user can load the version incorporated in this program if this is necessary.

- a. Frequency distribution of length printed with midpoints of specified class intervals.
- b. Frequency distribution of weight printed with midpoints of specified class intervals.
- c. Frequency distribution of length plotted against classes.
- d. Frequency distribution of weight plotted against classes.
- e. Length-weight values averaged by length classes and plotted against each other.
- f. Total length and scale diameter values averaged by length classes and plotted against each other.

Part III: The subroutine which calculates previous growth, generates and prints all values in Part III. The value of the length intercept (constant in body-scale equation) to use in the Lee-Lea growth

calculation formula is drawn from mainline (calculated in a previous routine). Sexes are sorted and males are processed first.

- a. A table is printed of average calculated lengths to each annulus, average length at capture for each age, number of each age, average age of group, weighted mean of length at capture, and number in group.
- b. Weighted means printed of average calculated length for all ages.
- c. Graph prepared of weighted mean of calculated length against year of life.
- d. A table is printed of average calculated weights to each annulus, average weight at capture for each age, number of each age, average age of group, weighted mean weight at capture, total number in group, and weighted mean weight at each annulus.
- e. Graph prepared of weighted mean weight against year of life.
- f. A table is printed of average increment in length by year of life, for each age.
- g. Weighted mean printed of average increment in length for all ages.
- h. Graph prepared of weighted average increment in length against year of life.
- i. Then a. -f. repeated for females only.
- j. Then a.-f. repeated for males and females combined.

USE OF THE PROGRAM

To use the listed program the data must be punched in the format set out. To increase the number of digits any variable may have, the READ statement in SUBROUTINE READER must be changed. The names associated with the columns of input are explained here; however, the

program utilizes these variables after leaving SUBROUTINE SETZR. Specifications given for the read variables refer only to the program as it is now. Each card may contain the following. The name of the variable that represents each type of data, follows each description.

<u>Columns</u>	<u>Datum held, suggested coding, and variable name</u>
1 - 8	Organisms name, any eight letters; first letter of genus, space, then first six letters of species name provides usable code. NAME1, NAME2
9	Sex of fish. Punch either M or F. SEX
10	Maturity, either I for immature or M for mature. MAT
11 - 13	Length, right justified. Maximum allowable is 999 mm. TLEN
14 - 17	Weight, right justified, rounded to nearest gram. Maximum allowable is 9999 grams. WT
18 - 20	Scale diameter (magnified image), right justified, to nearest millimeter. Maximum allowable 999 mm. KSCD
21 - 53	Annuli measurements (with same magnification as KSCD), three columns for each measurement, each right justified, to nearest mm. Eleven annuli maximum. KANN
54, 55	Age, right justified, eleven maximum. AGE
56	Code for possible following card with stomach contents of specimen. Leave blank if no stomach data follow. If so punch a one (1), and add user designed routine to program. JSTOM
57 - 59	Location code for specimen. Numbered stations up to three digits, right justified. LOC
60 - 65	Date of collection. Six digits allowed in groups of two for month, day, and year. For example 030469 indicates March 4, 1969; and 112357 means November 23, 1957. DATE
66 - 68	Type of capture gear code, coded by letters, numbers, or any combination. For example GN4 for 4 inch gill nets;

or, T21 for trawl with two inch cod end and of size (width) one. GEAR

- 69, 70 Depth of gear. Two digits in users units. IDPGR
- 71 - 73 Water temperature, three digits. First decimal included but decimal point not punched. For example 10.7°C punched as 107, C or F not punched but must be same throughout.
- 74, 75 Current, direction and speed coded, one column each using numerals. For example 88 means from north at 8 cm/sec; or 64 means from west at 4 cm/sec. Eight points on compass given digits 1 - 8 starting with 1 = northeast.
KCUR
- 76 Sea conditions, coded with users letter choice. KSEA
- 77 Sky conditions, coded with users letter choice. SKY
- 78 - 80 Time of capture, three digits to closest half hour. For example 103 equals 1030 hours, or 050 means 0500 hours.
TIME

If no data are available for some of the variables, leave spaces in appropriate columns. For most collections, all of the information in columns 56-80 will be the same for many specimens, so that each card can be duplicated from the previous card for these variables. The program presented here does nothing with the information in columns 56-80, but the reading and sorting capability is built in so that, with a user designed sort procedure, any type of calculation can be performed later. Having all specimen information on one card also facilitates manual sorting of cards for different runs if computer sort statements are not written, and allows the cards to be used directly as permanent data sheets.

If data from more than one species are being analyzed in a single run, the data for each must be followed by a card with *EOF in columns 1 - 4. The final species is followed with an END-OF-JOB card (@FIN on the Univac 1108).

A control card must precede each data deck. This card provides the options to growth calculations and the diagnostic trace. If growth calculations are wanted, punch a one (1) in column one. To skip over growth calculations punch a zero (0). If the diagnostic trace is wanted punch YES in columns 2 - 4. Punch NO in columns 3 and 4 if the trace is not wanted.

With a one (1) in column one, and NO in columns three and four, the program computes all of the parameters listed in the previous section, then checks for the *EOF or END-OF-JOB card. If an *EOF card follows the last data card, control returns to the top of mainline, all values are reset to zero, and new calculations are begun for the next species. See flow chart in appendix.

The looping technique employed thus provides an external sorting technique that any non-programmer can use to process different data in one run. A typical problem might be that growth and other calculations are wanted for a group of fish collected in gill nets and pound nets, and then make the same calculations with the data combined. Rather than attempt to manipulate the program internally by writing sorting statements, the user need only duplicate the data cards. He separates one set manually into gill net and pound net specimens, leaves one set intact, and then places a *EOF card between the three card stacks. If the three decks are each preceded with a card with a one (1) punched in column one, growth calculations will be made on the three decks in addition to the calculations described in Part I and II. A definite advantage of this type of sorting is that internal sorting by special statements is not necessary.

As a more complete example, suppose that a species is collected from two locations and that at each location fyke nets, pound nets and gill nets are used. A logical sequence of calculations might be to calculate everything first from each net at each location, then to pool all the data from the three nets at each location, then to pool all data from both locations. Three sets of data cards (a, b, and c) must be made: (a) is

left together, (b) is manually divided into two decks (b1,b2) by location, the third set (c) is manually divided first into two by location (c1,c2), then each is divided into three by net (c1 = ca,cb,cc) and (c2 = cd,ce,cf). Then an *EOF card is inserted between every group of data cards and each group is preceded with a card with a one (1) in column one and with a NO in columns 3 and 4. If the cards are presented to the computer with the six small decks, followed by the two next size, followed by the large pooled deck, the sequence of calculations mentioned above will be performed in one run. The proper sequence is ca, cb, cc, cd, ce, cf, b1, b2, a. Nine separate runs would be needed in the above example to avoid duplicating the cards and manual sorting would still be necessary. Similarly, other species caught at the same time at the same locations can follow the first species. The number of species or groups is not restricted provided that an *EOF card follows each group.

DESCRIPTION OF ROUTINES AND VARIABLES

The following section is provided mainly for the investigator who wishes to manipulate the program or to understand the variable names. The non-programmer may omit this section because nothing is given that further describes use of the program.

The variable names usually indicate their function. The information each contains, whether computed or read in, is given below in the order of occurrence in the program. The arguments brought into, and generated out of, each subroutine are indicated after the generalized description of the routine's function. The same routine can be called in different places if appropriate changes are made in the call arguments. Arguments from MAIN, indicates the variables that are coming to the subroutine with values other than zero. Arguments to MAIN, indicate the variables that are computed, or zeroed, in that routine.

SUBROUTINE SETZR

This is the first subroutine of the program. Its function is to set all arrays to zero so that sums can be accumulated to the variables.

CLEAR is a computer-specific routine available on most machines. A FORTRAN equivalent of CLEAR is provided with the listing so it can be loaded into any computer if necessary.

Arguments from MAIN: none

Arguments to MAIN: SLEN, SWT, SAGE, TALY, MALES, FEMALS, WIDTHL, WIDTHW, SUMSCD, GRPTOT, FREQL, FREQW, SMCLLEN, SUMCWT.

The only variables which receive non-zero values in this routine are:

WIDTHL This is the class width for length. The number of intervals and/or the size of the interval can be altered. Loops, print statements, and graphs in SUBROUTINE PRINT must be changed accordingly.

WIDTHW This is the class width for weight. Same comments.

These variables and arrays are returned to MAIN as zero. They are calculated by other subroutines but are given here because of first occurrence.

SLEN Sum of total lengths.

SWT Sum of weights.

SAGE Sum of ages

TALY Number of specimens read in as a group, i. e. , between *EOF cards.

MALES Number of males.

FEMALS Number of females.

SUMSCD Sum of scale diameters in each total length class.

FREQL Number of specimens in each total length class.

FREQW Number of specimens in each weight class.

SMCLLEN Sum of total lengths in each length class.

SUMCWT Sum of weight in each weight class.

SUBROUTINE READER

This is the only place the data are read. The method of data input and description of variables has been previously explained. Several variables are used to control flow of the program.

Arguments from MAIN: J

Arguments to MAIN: ISTOP, NFISH, ISKIP, ITRACE, plus all variables read in (see use of program section).

- J Subscript of the specimen being read. Maximum in present program is 1000 of a species or group. Must change MAIN dimension card to increase.
- ISTOP A counter used to control program flow based on the *EOF and END-OF-JOB cards.
- NFISH The species name is transferred to this variable so *EOF will not be printed later. It holds the first eight columns of information from the previous card read.
- ISKIP A control card variable that allows the user to skip growth calculations. Is read by a separate read statement.
- ITRACE A control card variable that allows the user to check data input and prints error messages while the program is running.

SUBROUTINE MAGIC

This is the summing routine. Each specimen has some of its data accumulated to others in the group.

Arguments from MAIN: SEX, TLEN, WT, KSCD, J, AGE, ITRACE

Arguments to MAIN: MALES, FEMALS, TALY, SLEN, SWT, WIDTHL, SUMSCD, SMCLN, SUMCWT, GRPTOT, SAGE.

All arguments previously described.

SUBROUTINE FRQDIS

Frequency distributions are made for length and weight based on specified class widths. Weight conversion from grams to kilograms is optional.

From MAIN: TLEN, WT, J, WIDTHL, WIDTHW, ITRACE.

To MAIN: FREQL, FREQW.

All arguments previously described.

SUBROUTINE JUNE

Averages are computed from sums made in SUBROUTINE MAGIC.

From MAIN: SLEN, SWT, FEMALS, TALY, SUMSCD, GRPTOT,

SMCLEN, SUMCWT, SAGE.

To MAIN: PRFEML, ASCD, ALEN, AWT, ACLEN, ACWT, AAGE.

PRFEML Percentage females in group.

ASCD Average scale diameters in each total length class.

ALEN Average total length.

AWT Average weight.

ACLEN Average total length in each length class.

ACWT Average weight in each length class.

ASGE Average age (completed annuli).

SUBROUTINE REGRES

Constants of the body-scale equation and coefficient of correlation are computed. The body-scale relationship is assumed to be linear, expressible in the form $Y = a + bX$, where Y equals total length, a is the intercept on the total length axis, b is the slope of the regression line, and X is the scale diameter.

Constants of the length-weight equation are calculated. The length-weight relationship is assumed to be non-linear of the form $W = aL^n$, where W is weight, a is a constant, L is total length, and n is the slope of the line. The length-weight equation is computed logarithmically and the constant converted to a real number.

From MAIN: TLEN, WT, KSCD, TALY, ITRACE

To MAIN: A, B, R, C, Z, AA

A Constant of body-scale equation.

B Slope of body-scale equation.

R	Coefficient of correlation for body-scale data.
C	Constant of length-weight equation, log form.
Z	Slope (exponent) of length-weight equation.
AA	Anti-log of C, the real number of the constant in the length-weight equation.

SUBROUTINE PRINT

This is the main printing routine. All values come from MAIN. The graph used is a normal X,Y plot available on most computer library files. Only calculated lengths and weights and associated sums are printed elsewhere.

SUBROUTINE BAKCAL

This routine performs most of the ordinary growth calculations for the species based on annuli measurements. The growth computation formula utilized is the Lee-Lea,

$$L_n = a + \frac{(L_t - a)}{S_t} S_n$$

where L_n equals length of fish at annulus n, L_t equals fish total length, a equals the total length intercept of the body-scale equation, S_t equals the total scale diameter, and S_n equals the scale diameter at annulus n.

If weight by age is desired it can be computed from length at age by calling a separate subroutine within BAKCAL. The coefficients of the length-weight equation are brought in for the group, a table printed, and a graph made of the weighted mean weight for each group.

Statistical packages can be inserted in BAKCAL. The extent and type of package has been left entirely as a user option.

From MAIN: TLEN, KSCD, AGE, KANN, A, TALY, J, SEX, NFISH,
AA, Z, ITRACE.

To MAIN: Nothing.

KK	Counter to control looping.
LOOP	Counter to control looping, used with KK.
N	Taly converted to fixed point.
M	Maximum age encountered for males, females, combined.
LA	Age of specimen converted to fixed point.
SCD	Scale diameter in floating point.
ANN	Scale diameter at each annulus in floating point.
BKCLN	Calculated length at each annulus.
SMLN	Two dimensional array to sum and store calculated lengths.
SMLNCP	Sum of lengths at capture for each age.
NUM	Number of specimens of each age.
AVLNCP	Average length at capture.
AVLN	Average length at each annulus for all ages.
SMAGE	Sum of ages within group.
GSLNCP	Grand sum of lengths at capture for group
TOTFSH	Number of fish being processed as group, e.g., males only.
WMAGE	Weighted mean age for group (completed annuli).
WMLNCP	Weighted mean of length at capture.
KTOTFH	Fixed point form of TOTFSH.
NUMAN	Number of specimens for which growth was calculated for each particular annulus.
GSMLN	Grand sum of calculated lengths for each annulus.
WTMLN	Weighted mean of calculated length by annuli.
IANN	Numerical progression to print one to 11, ages.
YEAR	Same as IANN but in floating form

After printing and graphing the above, SUBROUTINE WTCAL is called. Return from WTCAL is to BAKCAL and the following are computed.

AVINL	Average increment of length by age and annuli.
NUMAN2	Equivalent to NUMAN.
SMINC	Sum of increments of length.
WAVIN	Weighted average increments of length.

SUBROUTINE WTCAL

This routine uses the average calculated lengths, length at capture, and weighted mean lengths of SUBROUTINE BAKCAL and converts them to weights. The constants used in the equation

$$Y = aX^n$$

are drawn from MAIN through BAKCAL and are specific for the species or group. The average weight at capture is arithmetically weighted. WTCAL is called from BAKCAL and is a part of the triple loop which calculates for males, females, then combines them. To skip over WTCAL requires insertion of one card in the deck. See list of program for card type and position.

From BAKCAL: C, Z, AVLNCP, AVLN, WTMLN, YEAR, M, NUM,
KK, LOOP, NFISH, WMAGE, KTOTFH, ITRACE

To BAKCAL: Nothing.

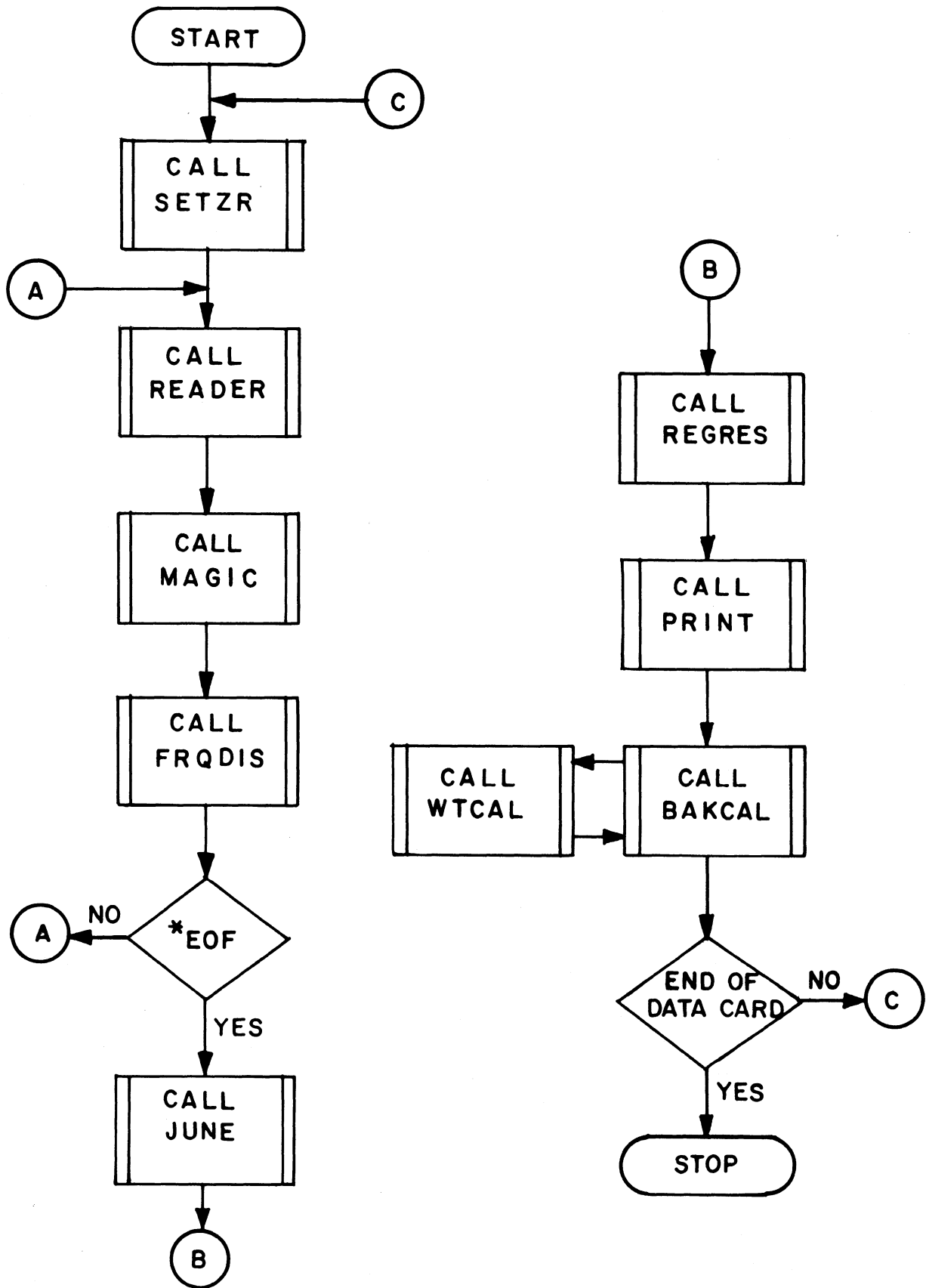
- MNWTCP The calculated weight from the average length at capture for each age.
- MNWT Calculated weight from weighted mean length at each annulus.
- MNCLWT Calculated weight of the average lengths at each annulus.
- KSMWTC Sum of the weights at capture, weighted by number at each age.
- JMWTC Weighted mean weight at capture.
- AVEWT MNWT in floating form, changed for GRAPH.

Control returns to WTCAL and increments of length are computed. After the males are processed (first time through) the females are processed, then males and females are combined.

ACKNOWLEDGEMENTS

Leonard P. Levine assisted with some of the logic, A. M. Beeton reviewed the manuscript, and Edward H. Brown, Jr. gave the author permission to review his calculation routine. Mitch Blank provided the graphing program.

Financial assistance was provided in part by the Bureau of Commercial Fisheries, Anadromous Fish Program, contract number 14-17-0004-329.



MASTER FLOW CHART

APPENDIX

0ASG,A ELECTRICAL*GRAPHACK

@FOR,ASI .MAIN

C
 C MASTER MAINLINE FISH PROGRAM BEGINS
 C THIS PROGRAM HAS AN EXTERNAL CALL TO A SUBROUTINE
 C GRAPH (XARRAY, YARRAY, LX, LY, 'NORMAL') WHICH CAUSES A
 C PAGEPLOT OF VARIOUS SORTS OF DATA TO BE MADE. IN THIS
 C SUBROUTINE XARRAY IS THE ARRAY TO BE PLOTTED ON THE X AXIS,
 C YARRAY IS THE ARRAY TO BE PLOTTED ON THE Y AXIS, LX IS THE
 C LENGTH OF THE PLOT ON THE X AXIS, LY IS THE SAME FOR Y, AND
 C THE LITERAL 'NORMAL' IS ONE OF THE VARIATIONS OF GRAPH.
 C THE CALLS FOR GRAPH ARE IN SUBROUTINES PRINT AND BAKCAL.
 C THIS PROGRAM HAS ONE SUBROUTINE, CLEAR, WHICH IS WRITTEN
 C IN UNIVAC 1108 ASSEMBLER FOR SPEED IMPROVEMENT. THIS
 C SUBROUTINE CAN BE WRITTEN IN FORTRAN.

PROGRAMMED BY WALTER J. HOGMAN
 CENTER FOR GREAT LAKES STUDIES
 UNIV. OF WIS. - MIL.
 MILWAUKEE, WISCONSIN

DIMENSION NAME1(1000), NAME2(1000), SEX(1000), MAT(1000),
 1TLEN(1000), WT(1000), KSCD(1000), KANN(1000, 11), AGE(1000),
 2JSTOM(1000), LOC(1000), DATE(1000), GEAR(1000), IDPGR(1000),
 3WATEM(1000), KCUR(1000), KSEA(1000), SKY(1000), TIME(1000),
 4WIDTHL(50), FREQW(50), WIDTHW(50), ASCD(50), SUMSCD(50),
 5GRPTOT(50), SMCLN(50), ACLN(50), SUMCWT(50), ACWT(50),
 6NFISH(2), FREQL(50)

ITRACE IS A DIAGNOSTIC USED THROUGHOUT TO TRACE THE FLOW OF THE
 PROGRAM. ONCE THE PROGRAM IS CORRECT, OR ALL CHANGES HAVE BEEN
 MADE, CHANGE ITRACE ON THE DATA CONTROL CARD, E.G. PUNCH A
 NO IN COMUMNS 2 AND 3.

COMMON ITRACE

10 J = 0
 IF(ITRACE.EQ.'YES')PRINT 20
 20 FORMAT (' NOW IN MAIN')
 CALL SETZR
 1(SLEN, SWT, SAGE, TALLY, MALES, FEMALS, WIDTHL, WIDTHW, SUMSCD,
 2GRPTOT, FREQL, FREQW, SMCLN, SUMCWT)
 30 J = J + 1
 IF(ITRACE.EQ.'YES')PRINT 20
 CALL READER
 1(NAME1, NAME2, SEX, MAT, TLEN, WT, KSCD, KANN, AGE, JSTOM,
 2LOC, DATE, GEAR, IDPGR, WATEM, KCUR, KSEA, SKY, TIME, ISTOP,
 3J, NFISH, ISKIP)
 GO TO (40,50,50),ISTOP
 40 IF(ITRACE.EQ.'YES')PRINT 20
 CALL MAGIC
 1(SEX, TLEN, WT, MALES, FEMALS, TALLY, SLEN, SWT, WIDTHL, SUMSCD,
 2KSCD, SMCLN, SUMCWT, GRPTOT, J, AGE, SAGE)
 IF(ITRACE.EQ.'YES')PRINT 20
 CALL FRQDIS
 1(TLEN, WT, FREQL, WIDTHL, FREQW, WIDTHW, J)

```

GO TO 30
50 IF(I TRACE.EQ.'YES')PRINT 20
   CALL JUNE
   1(SLEN, SWT, FEMALS, TALY, ALEN, AWT, PRFEML, SUMSCD, ASCD,
   2GRPTOT, SMCLN, ACLEN, SUMCWT, ACWT, SAGE, AAGE
   IF(I TRACE.EQ.'YES')PRINT 20
   CALL REGRES
   1(TLEN, WT, KSCD, TALY, A, B, R, C, Z, AA
C
C   TO SKIP OVER SUBROUTINE PRINT, INSERT A CARD 'GO TO 60', HERE
C
   IF(I TRACE.EQ.'YES')PRINT 20
   CALL PRINT
   1(NFISH, ALEN, AWT, PRFEML, TALY, FREQL, WIDTHL, FREQW, WIDTHW,
   2ASCD, A, B, R, C, Z, AA, ACLEN, ACWT, AAGE
C
C   IF BACK CALCULATIONS OF GROWTH ARE WANTED, ISKIP MUST BE
C   EQUAL TO 1 ON A CARD IMMEDIATELY BEFORE THE SPECIES DATA
C   CARDS. TO SKIP OVER GROWTH CALCULATIONS, PUNCH A 0 (ZERO).
C   THE 1 OR THE ZERO MUST BE FOLLOWED BY A COMMA.
C
   IF(ISKIP) 61, 61, 60
60 CALL BAKCAL
   1(TLEN, KSCD, AGE, KANN, A, TALY, J, SEX, NFISH, C, Z
61 GO TO (10, 10, 70), ISTOP
70 IF(I TRACE.EQ.'YES')PRINT 80
80 FORMAT (' ENTIRE PROGRAM TRACED AND OVER')
   END

```

@FOR,ASI SUBSETZR

```

C
C   THIS ROUTINE SETS THE FREQUENCY DISTRIBUTION ARRAYS TO ZERO,
C   ESTABLISHES THE CLASS LIMITS FOR THE FREQ. DIST., AND SETS
C   SUM TOTALS TO ZERO.
C

```

```

SUBROUTINE SETZR
1(SLEN, SWT, SAGE, TALY, MALES, FEMALS, WIDTHL, WIDTHW, SUMSCD,
2GRPTOT, FREQL, FREQW, SMCLN, SUMCWT
DIMENSION WIDTHL(50), WIDTHW(50), FREQL(50), FREQW(50),
1SUMSCD(50), GRPTOT(50), SMCLN(50), SUMCWT(50)
COMMON I TRACE
IF(I TRACE.EQ.'YES')PRINT 10
10   FORMAT (' BEGIN SETZR')
   CALL CLEAR (FREQL, 50, 0.)
   CALL CLEAR (FREQW, 50, 0.)
   CALL CLEAR (SUMSCD, 50, 0.)
   CALL CLEAR (SMCLN, 50, 0.)
   CALL CLEAR (SUMCWT, 50, 0.)
   CALL CLEAR (GRPTOT, 50, 0.)
   WIDTHL(1) = 0.

```

```

WIDTHW(1) = 0.
DO 20 I=2,50
WIDTHL(I) = WIDTHL(I - 1) + 20.0
20 WIDTHW(I) = WIDTHW(I - 1) + .1
SLEN = 0.
SWT = 0.
SAGE = 0.
TALY = 0.
MALES = 0.
FEMALS = 0.
IF(ITRACE.EQ.'YES')PRINT 30
30   FORMAT (' FIRST RETURN')
RETURN
END

```

@FOR,ASI SUBREADER

C
C THIS ROUTINE READS THE DATA CARDS AND ASSIGNS A SUBSCRIPT TO
C EACH SPECIMEN. ALL THE DATA FOR THAT SPECIMEN IS ASSIGNED THE
C SAME SUBSCRIPT. IF MORE THAN ONE SPECIES IS RUN AT ONCE, THE
C TEST CARD FOR THE END OF A RUN FOR A SPECIES IS *EOF. IF
C ONLY ONE SPECIES IS RUN THE @FIN CARD (END-OFJOB-CARD) IS TESTED.
C

SUBROUTINE READER

```

1(NAME1, NAME2, SEX, MAT, TLEN, WT, KSCD, KANN, AGE, JSTOM,
2LOC, DATE, GEAR, IDPGR, WATEM, KCUR, KSEA, SKY, TIME, ISTOP,
3J, NFISH, ISKIP
  DIMENSION NAME1(1000), NAME2(1000), SEX(1000), MAT(1000),
  1TLEN(1000), WT(1000), KSCD(1000), KANN(1000, 11), AGE(1000),
  2JSTOM(1000), LOC(1000), DATE(1000), GEAR(1000), IDPGR(1000),
  3WATEM(1000), KCUR(1000), KSEA(1000), SKY(1000), TIME(1000),
  4NFISH(2)
  COMMON ITRACE
  IF(ITRACE.EQ.'YES')PRINT 10
10   FORMAT (' BEGIN READER')
  IF(J-1) 11, 11, 12
11 READ 21, ISKIP, ITRACE
21 FORMAT (I1, A3)
12 READ(5, 20, END = 70, ERR = 13)
  1NAME1(J), NAME2(J), SEX(J), MAT(J), TLEN(J), WT(J), KSCD(J),
  2(KANN(J, I), I = 1, 11), AGE(J), JSTOM(J), LOC(J), DATE(J),
  3GEAR(J), IDPGR(J), WATEM(J), KCUR(J), KSEA(J), SKY(J),
  4TIME(J)
20 FORMAT (2A4, A1, A1, F3.0, F4.0, I3, 11I3, F2.0, I1, A3, 3I2,
  1A3, I2, F3.0, I2, A1, A1, F3.0)
  GO TO 15
13 PRINT 14
14 FORMAT (1HD, 5X, 'ERROR ON CARD, NOT ALL READ')
15 ISTOP = 1
  IF(NAME1(J).EQ.'*EOF')GO TO 50

```

C
C
C
C
C
C
C

SPECIMEN NAME IS BEING HELD SO *EOF IS NOT PRINTED LATER.

NFISH(1) = NAME1(J)
NFISH(2) = NAME2(J)

THIS IS A DIAGNOSTIC PRINT TO CHECK THAT SPECIMEN DATA
ENTERED PROPERLY.

IF(ITRACE.EQ.'YES')PRINT 30,
1NAME1(J), NAME2(J), SEX(J), MAT(J), TLEN(J), WT(J), KSCD(J),
2(KANN(J, I), I = 1, 11), AGE(J), JSTOM(J), LOC(J), DATE(J),
3GEAR(J), IDPGR(J), WATEM(J), KCUR(J), KSEA(J), SKY(J),
4TIME(J)

30 FORMAT (1X, 2A4, 2X, A1, 2X, A1, 2X, F4.0, 2X, F5.0, 2X, I3,
12X, 11(2X,I3), 2X, F3.0, /, 2X, I1, 2X, A3, 2X, 3I2, 2X, A3, 2X,
2I2, 2X, F4.0, 2X, I2, 2X, A1, 2X, A1, 2X, F4.0)

IF(ITRACE.EQ.'YES')PRINT 40

40 FORMAT (' FIRST RETURN')
RETURN

50 ISTOP = 2

IF(ITRACE.EQ.'YES')PRINT 60

60 FORMAT (' SECOND RETURN')
RETURN

70 ISTOP = 3

IF(ITRACE.EQ.'YES')PRINT 80

80 FORMAT (' THIRD RETURN')
RETURN
END

@FOR,ASI SUBMAGIC

C
C
C
C
C
C

THIS ROUTINE SUMS EACH SPECIMENS TOTAL LENGTH, WEIGHT, AND SCALE
DIAMETER INTO SUBSCRIPTED ARRAYS BASED ON THE CLASS WIDTH OF
TOTAL LENGTH. A SEPARATE TALLY IS MAINTAINED FOR EACH CLASS
INTERVAL. MALES AND FEMALES ARE COUNTED.

SUBROUTINE MAGIC

1(SEX, TLEN, WT, MALES, FEMALES, TALY, SLEN, SWT, WIDTHL, SUMSCD,
2KSCD, SMCLN, SUMCWT, GRPTOT, J, AGE, SAGE)
DIMENSION SUMSCD(50), GRPTOT(50), WIDTHL(50), SMCLN(50),
1SUMCWT(50), SEX(1), TLEN(1), WT(1), KSCD(1), AGE(1)

COMMON ITRACE

IF(ITRACE.EQ.'YES')PRINT 10

10 FORMAT (' BEGIN MAGIC')

SLEN = SLEN + TLEN(J)

SWT = SWT + WT(J)

SAGE = SAGE + AGE(J)

TALY = TALY + 1.

DO 20 I=1,50

```

IF (TLEN(J) .GE. WIDTHL(I) .AND. TLEN(J) .LT. WIDTHL(I+1))
1      GRPTOT(I) = GRPTOT(I) + 1.
IF (TLEN(J) .GE. WIDTHL(I) .AND. TLEN(J) .LT. WIDTHL(I + 1))
1      SMCLN(I) = SMCLN(I) + TLEN(J)
IF (TLEN(J) .GE. WIDTHL(I) .AND. TLEN(J) .LT. WIDTHL(I + 1))
1      SUMCWT(I) = SUMCWT(I) + WT(J)
IF (TLEN(J) .GE. WIDTHL(I) .AND. TLEN(J) .LT. WIDTHL(I + 1))
1      SUMSCD(I) = SUMSCD(I) + FLOAT(KSCD(J))
20 CONTINUE
IF (SEX(J) .EQ. 'M') MALES = MALES + 1.
IF (SEX(J) .EQ. 'F') FEMALS = FEMALS + 1.
IF (ITRACE.EQ.'YES')PRINT 30
30      FORMAT (' FIRST RETURN')
RETURN
END

```

@FOR,ASI SUBFRQDIS

C
C THIS ROUTINE ASSIGNS THE LENGTH AND WEIGHT DATA TO APPROPRIATE
C INTERVALS. LENGTHS ARE ASSIGNED TO 50 CLASSES EACH 20 MILLIMETERS
C WIDE. WEIGHTS ARE ASSIGNED TO 50 CLASSES EACH 100 GRAMS WIDE.
C

SUBROUTINE FRQDIS

1(TLEN, WT, FREQL, WIDTHL, FREQW, WIDTHW, J)
DIMENSION FREQL(50), WIDTHL(50), FREQW(50), WIDTHW(50),

1TLEN(1), WT(1)

COMMON ITRACE

IF (ITRACE.EQ.'YES')PRINT 10

10 FORMAT (' BEGIN FRQDIS')

C
C THE LENGTHS AND WEIGHTS WITHIN A CLASS ARE COUNTED BUT NOT
C SUMMED. THE CLASS LIMITS CAN BE CHANGED BY CHANGING A CARD IN
C SUBROUTINE SETZR.
C

DO 20 I=1,50

IF (TLEN(J) .LT. WIDTHL(I) .AND. WT(J) .LT. WIDTHW(I)) RETURN

IF (TLEN(J) .GE. WIDTHL(I) .AND. TLEN(J) .LT. WIDTHL(I + 1))

1 FREQL(I) = FREQL(I) + 1.

IF (WT(J)/1000. .GE. WIDTHW(I) .AND. WT(J)/1000. .LT.

1WIDTHW(I+1)) FREQW(I) = FREQW(I) + 1.

20 CONTINUE

IF (ITRACE.EQ.'YES')PRINT 30

30 FORMAT (' FIRST RETURN')

RETURN

END

@FOR,ASI SUBJUNE

C
C THIS ROUTINE PERFORMS THE CALCULATIONS TO YIELD ALL THE
C AVERAGE AND PERCENTAGE VALUES.
C

```

SUBROUTINE JUNE
1(SLEN, SWT, FEMALS, TALY, ALEN, AWT, PRFEML, SUMSCD, ASCD,
2GRPTOT, SMCLN, ACLN, SUMCWT, ACWT, SAGE, AAGE
3DIMENSION ASCD(50), SUMSCD(50), GRPTOT(50), ACLN(50), ACWT(50),
4SMCLN(50), SUMCWT(50)
COMMON ITRACE
IF(ITRACE.EQ.'YES')PRINT 10
10      FORMAT (' BEGIN JUNE')
ALEN = SLEN / TALY
AWT = SWT / TALY
PRFEML = (FEMALS / TALY) * 100.
DO 20 I=1,50
ACLN(I) = SMCLN(I) / GRPTOT(I)
ACWT(I) = SUMCWT(I) / GRPTOT(I)
AAGE = SAGE / TALY
20 ASCD(I) = SUMSCD(I)/GRPTOT(I)
IF(ITRACE.EQ.'YES')PRINT 30
30      FORMAT (' FIRST RETURN')
RETURN
END

```

@FOR,ASI SUBREGRES

C
C THIS ROUTINE COMPUTES THE LEAST SQUARE REGRESSION COEFFICIENTS
C OF THE BODY-SCALE EQUATION BASED ON THE AVERAGES WITHIN THE
C CLASS INTERVALS. THE COEFFICIENT OF CORRELATION IS COMPUTED
C FOR THE BODY-SCALE REGRESSION EQUATION. THEN THE LENGTH-
C WEIGHT EQUATION COEFFICIENTS ARE COMPUTED IN LOG FORM.
C THE LOG OF THE CONSTANT IS CONVERTED TO THE REAL NUMBER.
C ALL VALUES ARE SENT TO SUBROUTINE PRINT AND BAKCAL.
C

```

SUBROUTINE REGRES
1(TLEN, WT, KSCD, TALY, A, B, R, C, Z, AA
2DIMENSION TLEN(1000), WT(1000), KSCD(1000)
3COMMON ITRACE
IF(ITRACE.EQ.'YES')PRINT 10
10      FORMAT (' BEGIN REGRES')
SUMX = 0.
SUMY = 0.
SUMXSQ = 0.
SUMYSQ = 0.
SUMXPY = 0.
SUMLGX = 0.
SUMLGY = 0.
SMLXPY = 0.
SMLXSQ = 0.

```

```

C
C   BEGIN BODY-SCALE CALCULATIONS.
C
  N = TALY
  DO 20 I=1,N
  IF(TLEN(I).LT.1.)GO TO 20
  X = KSCD(I)
  Y = TLEN(I)
  IF(ITRACE.EQ.'YES')PRINT 70, X,Y
  SUMX = SUMX + X
  SUMY = SUMY + Y
  SUMXSQ = SUMXSQ + X * X
  SUMYSQ = SUMYSQ + Y * Y
  SUMXPY = SUMXPY + X * Y
20 CONTINUE
  IF(ITRACE.EQ.'YES')PRINT 30
30   FORMAT (' LENGTH-SCALE DIAMETER VALUES HAVE BEEN COMPUTED')
  IF(ITRACE.EQ.'YES')PRINT 70, SUMX,SUMY,SUMXSQ,
1SUMYSQ, SUMXPY
C
C   COMPUTE BODY-SCALE COEFFICIENT OF CORRELATION.
C
  A = ((SUMY * SUMXSQ) - (SUMX * SUMXPY))/((TALY * SUMXSQ) -
1(SUMX **2))
  B = ((TALY * SUMXPY) - (SUMX * SUMY))/((TALY * SUMXSQ) -
1(SUMX **2))
  R = ((TALY * SUMXPY) - (SUMX * SUMY))/SQRT(((TALY * SUMXSQ) -
1SUMX **2) * ((TALY * SUMYSQ) - (SUMY **2)))
  IF(ITRACE.EQ.'YES')PRINT 40
40   FORMAT (' LEN-SCALE CONSTANTS AND COEF. OF CORR. COMP. ')
C
C   BEGIN LENGTH-WEIGHT CALCULATIONS.
C
  DO 50 I=1,N
  IF(TLEN(I).LT.1.)GO TO 50
  X = ALOG10 (TLEN(I))
  Y = ALOG10 (WT(I))
  IF(ITRACE.EQ.'YES')PRINT 70, X,Y
  SUMLGX = SUMLGX + X
  SUMLGY = SUMLGY + Y
  SMLXSQ = SMLXSQ + X * X
  SMLXPY = SMLXPY + X * Y
50 CONTINUE
  IF(ITRACE.EQ.'YES')PRINT 60
60   FORMAT(' LEN-WT VALUES HAVE BEEN COMPUTED')
  IF(ITRACE.EQ.'YES')PRINT 70, SUMLGX,SUMLGY,
1SUMXPY, SMLXSQ
70 FORMAT (1X, 5(1X, F14.4))
C
C   COMPUTE LENGTH-WEIGHT COEFFICIENTS.
C
  AA = ((SUMLGY * SMLXSQ) - (SUMLGX * SMLXPY))/((TALY * SMLXSQ) -
1(SUMLGX **2))
  Z = (SUMLGY - (TALY * AA)) / SUMLGX
  C = 10. **AA

```

```

      IF (ITRACE.EQ.'YES')PRINT 80
80    FORMAT(' CONSTANTS OF LEN-WT EQUATION COMPUTED')
      IF (ITRACE.EQ.'YES')PRINT 90
90    FORMAT (' FIRST RETURN')
      RETURN
END

```

```

@FOR,ASI    SUBPRINT

```

```

C
C    THIS ROUTINE IS THE MAIN PRINTING ROUTINE.  THE AVERAGES,
C    TOTALS, FREQUENCY DISTRIBUTION, EQUATION COEFFICIENTS,
C    AND GRAPHS FOR A SINGLE SPECIES ARE PRINTED.
C
      SUBROUTINE PRINT
      1(NFISH, ALEN, AWT, PRFEML, TALY, FREQL, WIDTHL, FREQW, WIDTHW,
      2ASCD, A, B, R, C, Z, AA, ACLEN, ACWT, AAGE )
      DIMENSION NFISH(2), FREQL(50), WIDTHL(50), FREQW(50),
      1WIDTHW(50), ASCD(50), ACLEN(50), ACWT(50)
      COMMON ITRACE
      IF (ITRACE.EQ.'YES')PRINT 10
10    FORMAT (' BEGIN PRINT')
      PRINT 20
20    FORMAT (1H1,35X,'ALL VALUES GIVEN HERE ARE FOR SEXES COMBINED',
      1//// )
      PRINT 30, (NFISH(K),K=1,2),ALEN,AWT,PRFEML,TALY
30    FORMAT(10X2A4, 3X'SPECIES'// 10XF5.1, 3X'AVERAGE TOTAL LENGTH'
      1//8X,F7.2,3X'AVERAGE WEIGHT'//10XF5.1,3X'PERCENTAGE FEMALES',
      2//10XF5.1,3X'TOTAL NUMBER OF THIS SPECIES' )
      PRINT 40, AAGE
40    FORMAT (/, 10X, F5.1, 3X, 'AVERAGE AGE (COMPLETED ANNULI)' )
      PRINT 50, A,B,R
50    FORMAT(/,14X, 'Y = ',F8.2, ' + ',F8.4, ' X ', 10X,
      1'REGRESSION EQUATION OF TOTAL LENGTH ON SCALE DIAMETERS.',
      2//, 9X,F6.3, 3X, 'COEFFICIENT OF CORRELATION OF BODY-SCALE '
      3'DATA' )
      PRINT 60, C,Z
60    FORMAT (1H0,13X, 'W = ',F6.4,1X,'TIMES L TO THE',F8.4,10X
      1'LENGTH - WEIGHT EQUATION' )
      PRINT 70, AA
70    FORMAT (1H0,6X,F8.4,3X, 'LOG OF C IN LEN.-WT. EQUATION')
      PRINT 80
80    FORMAT(1H1, 20X'MIDPOINT OF CLASS INTERVAL FOR LENGTH AND WEIGHT
      1' )
      PRINT 90
90    FORMAT(1H0,9X,'CLASS MM.',5X'LENGTH FREQ.', 20X'CLASS KGRAMS',
      15X'WEIGHT FREQ.' )
      PRINT 100
100   FORMAT(10X'-----',5X'-----',20X'-----',
      15X'-----' )
      DO 110 I=1,49

```

```

ACWT(I) = ACWT(I)/1000.
WIDTHL(I) = (WIDTHL(I) + WIDTHL(I + 1))/2.
WIDTHW(I) = (WIDTHW(I) + WIDTHW(I + 1))/2.
110 PRINT 120, WIDTHL(I),FREQ(L(I),WIDTHW(I),FREQW(I)
120 FORMAT(1H ,10X,F6.1, 12X, F5.1, 23X, F7.2, 13X, F5.1 )
    IF(ITRACE.EQ.'YES')PRINT 130
130     FORMAT(' END OF SPECIES PRINT. BEGIN GRAPHING FOR SPECIES')
    PRINT 140
140 FORMAT (1H1)
    CALL GRAPH (WIDTHL, FREQ(L, 50, 9, 6, 'NORMAL')
    PRINT 150, NFISH
150 FORMAT (1H0,3X,2A4,' FREQUENCY DISTRIBUTION OF NUMBERS WITHIN'
1° LENGTH CLASSES, LENGTH ON ABSCISSA, SEXES COMBINED' )
    PRINT 140
    CALL GRAPH (WIDTHW, FREQ(W, 50, 9, 6, 'NORMAL')
    PRINT 160, NFISH
160 FORMAT (1H0,3X,2A4,' FREQUENCY DISTRIBUTION OF NUMBERS WITHIN',
1° WEIGHT CLASSES, WEIGHT ON ABSCISSA, SEXES COMBINED' )
    PRINT 140
    CALL GRAPH (ACLEN, ACWT, 50, 9, 6, 'NORMAL')
    PRINT 170, NFISH
170 FORMAT (1H0,3X,2A4,' AVERAGE LENGTH IN CLASS AGAINST AVERAGE',
1° WEIGHT IN CLASS, LENGTH ON ABSCISSA, SEXES COMBINED' )
    PRINT 140
    CALL GRAPH (ASCD, ACLEN, 50, 9, 6, 'NORMAL')
    PRINT 180, NFISH
180 FORMAT (1H0,3X,2A4,' AVERAGE TOTAL LENGTH AGAINST AVERAGE',
1° SCALE DIAMETER, SCALE DIA. ON ABSCISSA, SEXES COMBINED' )
    IF(ITRACE.EQ.'YES')PRINT 190
190     FORMAT (' FIRST RETURN')
    RETURN
    END

```

@FOR,ASI SUBBAKCAL

```

C
C THIS ROUTINE PERFORMS ALL OF THE ORDINARY GROWTH CALCULATIONS
C FOR THE SPECIES. TWO TABLES (AVERAGE BACK CALCULATED GROWTH
C AND AVERAGE INCREMENT OF GROWTH) ARE COMPUTED AND PRINTED
C FOR MALES ALONE, THEN FEMALES ALONE, THEN MALES AND FEMALES
C COMBINED. A GRAPH OF THE WEIGHTED MEANS FOR EACH TABLE
C FOLLOWS THE TABLE. DIFFERENT GRAPHS OF THE SIZE BY AGE
C GROUP OR INCREMENT BY AGE GROUP CAN BE INSERTED USING THE
C ARRAYS ALREADY COMPUTED. TO SORT THE SPECIMENS BY LOCATION,
C METHOD OF CAPTURE, DATE OF CAPTURE, ETC. REQUIRES ONLY THE
C INSERTION OF THE PROPER IF STATEMENT AFTER THE MASTER DO LOOP.
C IF THE SEXES ARE TO BE CALCULATED SEPARATELY RETAIN THE SEX
C SORT STATEMENT. IF THE SEXES ARE TO BE COMBINED ONLY, FOLLOW
C OTHER COMMENT CARD INSTRUCTIONS.
C

```

SUBROUTINE BAKCAL

```

1(TLEN, KSCD, AGE, KANN, A, TALY, J, SEX, NFISH, C, Z
  DIMENSION TLEN(1000), KSCD(1000), AGE(1000), KANN(1000,11),
1SCD(1000), ANN(1000), BKCLN(11), SMLN(11,11), SMLNCP(11),
2NUM(11), AVLNCP(11), AVLN(11,11), NUMAN(11), GSMLN(11),
3WTMLN(11), IANN(11), AVINL(11,11), NUMAN2(11), SMINC(11),
4WAVIN(11), SEX(1000), NFISH(2), YEAR(20)
  COMMON ITRACE
  IF(ITRACE.EQ.'YES')PRINT 10
10   FORMAT (' BEGIN BACK CALCULATIONS FOR ALL FISH OF SPECIES')
  KK = 0

C
C
C   CHANGE LOOP = END IF SEXES ARE TO BE COMBINED ONLY.

  LOOP = 'NO'
20  CALL CLEAR (SCD, 500, 0.)
  CALL CLEAR (ANN, 500, 0.)
  CALL CLEAR (BKCLN, 11, 0.)
  CALL CLEAR (SMLNCP, 11, 0.)
  CALL CLEAR (SMINC, 11, 0.)
  CALL CLEAR (NUM, 11, 0)
  CALL CLEAR (AVLNCP, 11, 0.)
  CALL CLEAR (NUMAN, 11, 0)
  CALL CLEAR (GSMLN, 11, 0.)
  CALL CLEAR (WTMLN, 11, 0.)
  CALL CLEAR (IANN, 11, 0)
  CALL CLEAR (NUMAN2, 11, 0)
  CALL CLEAR (WAVIN, 11, 0.)
  DO 30 I=1,11
  DO 30 II=1,11
  AVLN(I, II) = 0.
  AVINL(I, II) = 0.
30  SMLN(I, II) = 0.
  N = TALY
  M = 0
  SMAGE = 0.
  GSLNCP = 0.
  TOTFSH = 0.
  WMAGE = 0.
  WMLNCP = 0.

C
C
C   THIS IS THE MASTER DO LOOP.

  DO 60 J=1,N

C
C
C   INSERT VARIOUS SORTING STATEMENTS HERE.

  IF(SEX(J).EQ.'F'.AND.LOOP.EQ.'NO')GO TO 60
  IF(SEX(J).EQ.'M'.AND.LOOP.EQ.'YES')GO TO 60
  LA = AGE(J)
  SCD(J) = KSCD(J)
  DO 40 I=1,LA
  ANN(I) = KANN(J, I)
40  BKCLN(I) = 0.
  DO 50 K=1,LA

```

```

C   MAIN BACK CALCULATION STATEMENT AND ARRAY POSTING WHILE SUMMING.
C
  BKCLN(K) = (IANN(K) * (ITLEN(J) - A) / SCD(J)) + A
50  SMLN(LA, K) = SMLN(LA, K) + BKCLN(K)
  SMLNCP(LA) = SMLNCP(LA) + TLEN(J)
  NUM(LA) = NUM(LA) + 1
  IF (LA .GT. M) M = LA
60  CONTINUE
  IF (ITRACE.EQ.'YES')PRINT 70
70   FORMAT (' MID-FILE DIAGNOSTIC TRACE')
  DO 100 I=1,M
  IF(NUM(I)) 100,100,80
80  AVLNCP(I) = SMLNCP(I)/NUM(I)
  DO 90 L=1,I

C
C   AVERAGE CALCULATED GROWTH COMPUTED BY AGE.
C
  90  AVLN(I, L) = SMLN(I, L) / NUM(I)
100  CONTINUE
  DO 120 I=1,M
  IF(NUM(I)) 120,120,110
110  SMAGE = SMAGE + I * NUM(I)
  GSLNCP = GSLNCP + SMLNCP(I)
  TOTFSH = TOTFSH + FLOAT(NUM(I))
120  CONTINUE
  WMAGE = SMAGE / TOTFSH
  WMLNCP = GSLNCP / TOTFSH
  KTOTFH = TOTFSH
  DO 140 I=1,M
  NUMAN(I) = 0
  DO 140 L=I,M
  GSMLN(I) = GSMLN(I) + SMLN(L, I)
  NUMAN(I) = NUMAN(I) + NUM(L)
  IF(L-M) 140,130,140

C
C   WEIGHTED MEAN OF TOTAL LENGTH BY YEAR COMPUTED.
C
130  WTMLN(I) = GSMLN(I) / NUMAN(I)
140  CONTINUE
  IF (ITRACE.EQ.'YES')PRINT 70
  IANN(1) = 1
  DO 150 K=2,11
150  IANN(K) = IANN(K - 1) + 1
  PRINT 160, (IANN(K),K=1,11)
160  FORMAT (1H1,//////////,15X, 'AVERAGE BACK-CALCULATED LENGTH ',
  1'TO EACH ANNULUS AND AVERAGE LENGTH AT CAPTURE', 1H0, 16X,
  2'AGE NUMBER CAPTURE', 9(6X,I1), 2(5X,I2) )
  PRINT 170
170  FORMAT (1H+, 14X, 97(1H_) )
  DO 200 I=1,M
  IF(NUM(I)) 200,200,180
180  PRINT 190, I,NUM(I),AVLNCP(I),(AVLN(I,L),L=1,M)
190  FORMAT (1H0,16X,I2, 3X,I4, 2X,F6.1, 3X,F6.1, 10(1X,F6.1) )
200  CONTINUE
  PRINT 170

```

```

PRINT 210, WMAGE,KTOTFH,WMLNCP,(WTMLN(I),I=1,M)
210 FORMAT (1H0, 14X, F5.1, 2X, I4, 2X, F6.1, 2X, 11F7.1 )
PRINT 220
220 FORMAT (1H0, 16X, 'WEIGHTED MEANS' )
PRINT 230
230 FORMAT (///)
IF(LOOP.EQ.'NO')PRINT 240, NFISH
240 FORMAT ( /,1H0, 14X, 'MALES ONLY OF', 1X(2A4) )
IF(LOOP.EQ.'YES'.AND.KK.EQ.1)PRINT 250, NFISH
250 FORMAT ( /, 1H0, 14X, 'FEMALES ONLY OF ', 2A4 )
IF(LOOP.EQ.'END'.AND.KK.EQ.2)PRINT 260, NFISH
260 FORMAT (/, 1H0, 14X, 'MALES AND FEMALES COMBINED OF ',2A4 )
DO 270 I=1,11
270 YEAR(I) = IANN(I)
PRINT 280
280 FORMAT (1H1)
CALL GRAPH (YEAR, WTMLN, 11, 9, 6, 'NORMAL')
PRINT 290, NFISH
290 FORMAT (1H ,22X, 2A4, ' MEAN OF BACK CALCULATED GROWTH AGAINST',
1' YEAR OF LIFE (YEARS ON ABSCISSA)' )
IF(LOOP.EQ.'NO')PRINT 240, NFISH
IF(LOOP.EQ.'YES'.AND.KK.EQ.1)PRINT 250, NFISH
IF(LOOP.EQ.'END'.AND.KK.EQ.2)PRINT 260, NFISH

```

C
C
C
C

```

TO SKIP OVER SUBROUTINE WTCAL, FOLLOW THESE COMMENT CARDS
WITH A CARD GO TO 300

```

```

CALL WTCAL

```

```

1(C, Z, AVLNCP, AVLN, WTMLN, YEAR, M, NUM, KK, LOOP, NFISH,
2 WMAGE, KTOTFH )

```

```

300 IF(NUM(1)) 320,320,310
310 AVINL(1, 1) = AVLN(1, 1)
320 DO 350 I=2,M
IF(NUM(I)) 350,350,330
330 AVINL(I, 1) = AVLN(I, 1)
DO 340 K=2,I

```

C
C
C

```

INCREMENTS OF GROWTH BY AGE AND YEAR COMPUTED.

```

```

340 AVINL(I,K) = AVLN(I,K) - AVLN(I, K-1)
350 CONTINUE
IF(ITRACE.EQ.'YES')PRINT 70
DO 370 I=1,M
NUMAN2(I) = 0
DO 370 L=I,M
SMINC(I) = SMINC(I) + AVINL(L, I) * NUM(L)
NUMAN2(I) = NUMAN2(I) + NUM(L)
IF(L-M) 370,360,370

```

C
C
C

```

WEIGHTED MEAN INCREMENT BY YEAR COMPUTED.

```

```

360 WAVIN(I) = SMINC(I) / NUMAN2(I)
370 CONTINUE
IF(ITRACE.EQ.'YES')PRINT 70
WAVIN(1) = WTMLN(1)

```

```

      PRINT 380, (IANN(K),K=1,11)
380 FORMAT (1H1, //,23X, 'AVERAGE INCREMENT OF GROWTH',
1 ' DURING EACH YEAR OF LIFE',/, 1H0, 14X, 'AGE NUMBER',
22X,9(6X,I1), 2(5X,I2))
      PRINT 390
390 FORMAT (1H+, 14X, 9(1H_) )
      DO 420 I=1,M
      IF(NUM(I)) 420,420,400
400 PRINT 410, I,NUM(I),(AVINL(I,L),L=1,M)
410 FORMAT (1H0, 14X, I2, 3X,I4, 1X, F10.1, 10F7.1 )
420 CONTINUE
      PRINT 390
      PRINT 430, WMAGE,KTOTFH,(WAVIN(I),I=1,M)
430 FORMAT (1H0, 12X, F5.1, 2X,I4, 1X,F10.1, 10F7.1 )
      PRINT 440
440 FORMAT (1H0, 14X, 'WEIGHTED MEANS' )
      PRINT 230
      IF(LOOP.EQ.'NO')PRINT 240, NFISH
      IF(LOOP.EQ.'YES'.AND.KK.EQ.1)PRINT 250, NFISH
      IF(LOOP.EQ.'END'.AND.KK.EQ.2)PRINT 260, NFISH
      PRINT 280
      CALL GRAPH (YEAR, WAVIN, 11, 9, 6, 'NORMAL')
      PRINT 450, NFISH
450 FORMAT (1H , 22X, 2A4, ' MEAN INCREMENT OF GROWTH AGAINST',
1 ' YEARS (YEARS ON ABSCISSA)')
      IF(LOOP.EQ.'NO')PRINT 240, NFISH
      IF(LOOP.EQ.'YES'.AND.KK.EQ.1)PRINT 250, NFISH
      IF(LOOP.EQ.'END'.AND.KK.EQ.2)PRINT 260, NFISH
C
C   CHANGE 1 TO 3 IF SEXES ARE TO BE COMBINED.
C
      KK = KK + 1
      LOOP = 'YES'
      IF(KK-1) 20,20,460
460 LOOP = 'END'
      IF(ITRACE.EQ.'YES')PRINT 70
      IF(KK-2) 20,20,470
470 IF(ITRACE.EQ.'YES')PRINT 480
480     FORMAT (' FIRST RETURN')
      RETURN
      END

```

```

@ASM,ASI   CLEAR
.   TO USE CLEAR DO THE FOLLOWING
.   CALL CLEAR(ARRAY,N,FILL)
.   ARRAY IS THE ARRAY TO BE CLEARED
.   N IS THE NUMBER OF ELEMENTS TO BE CLEARED
.   FILL IS THE NUMBER TO WHICH ALL THE ELEMENTS OF ARRAY ARE SET.
S(1),CLEAR* LA 12,0,11
LA 13,2,11

```

```

LR 65,*1,11
LXI,15 12,1
BT 12,0,*13
J 4,11

```

THE CLEAR ROUTINE USED IN THIS PROGRAM IS MACHINE LANGUAGE. THE FOLLOWING SUBROUTINE IS THE FORTRAN EQUIVALENT OF CLEAR. IT SHOULD BE USED ONLY IF THE USERS COMPUTER DOES NOT HAVE A CLEAR FUNCTION BUILT INTO IT.

```

SUBROUTINE CLEAR (ARRAY, ICOUNT, X)
DO 71 I = 1, ICOUNT
71 ARRAY (I) = X
RETURN
END

```

END

@FOR,ASI SUBWTCAL

C
C
C
C
C

THIS SUBROUTINE CALCULATES FISH WEIGHT AT THE END OF EACH YEAR OF LIFE BASED ON AVERAGE LENGTH AND COMPUTED CONSTANTS OF THE LENGTH-WEIGHT EQUATION. WEIGHT IS EXPRESSED IN GRAMS

```

SUBROUTINE WTCAL
1(C, Z, AVLNCP, AVLN, WTMLN, YEAR, M, NUM, KK, LOOP, NFISH,
2 WMAGE, KTOTFH )
DIMENSION AVLNCP(11), AVLN(11,11), WTMLN(11), MNWTCP(11),
1MNCLWT(11,11), MNWT(11), YEAR(11), NUM(11), NFISH(2), AVEWT(11),
2IYEAR(11)
COMMON ITRACE
IF(ITRACE.EQ.'YES')PRINT 10
10 FORMAT (' BEGIN WTCAL')
IF(ITRACE.EQ.'YES')PRINT 20, C,Z,AVLNCP,WTMLN,AVLN
20 FORMAT (10X, F8.4, ' CONSTANT',//, 10X, F8.4, ' EXPONENT', //
110X, 11F8.1, 3X, ' AVLNCP', //, 10X, 11F8.1, 3X, ' WTMLN',
2//, 10X, 11(11F6.1//),3X, ' AVLN' )
IF(ITRACE.EQ.'YES')PRINT 30, M,NUM,YEAR, KK, LOOP, NFISH
30 FORMAT (10X, I4, ' M', //, 10X, 11I4, ' NUM', //, 10X,
111F4.1, ' YEARS', //, 10X, I4, ' KK', //, 10X, A4, ' LOOP',
2//, 10X, 2A4, ' NFISH' )
DO 50 I=1,M
MNWTCP(I) = 0
MNWT(I) = 0
DO 40 K=1,I
40 MNCLWT(I, K) = ,0
50 CONTINUE
KSMWTC = 0

```

C

```

C      ALL LENGTH VALUES ARE NOW COMPUTED TO WEIGHT VALUES.
C
      DO 80 I=1,M
      MNWT(I) = C * WTMLN(I) ** Z
      IF(NUM(I)) 80,80,60
60     MNWTC(I) = C * AVLNCP(I) ** Z
      KSMWTC = KSMWTC + (MNWTC(I) * NUM(I))
      DO 70 K=1,I
70     MNCLWT(I, K) = C * AVLN(I, K) ** Z
80     CONTINUE
      JMWTC = KSMWTC / KTOTFH
      IF(ITRACE.EQ.'YES')PRINT 90
90     FORMAT (5X, 'MID-FILE DIAGNOSTIC TRACE')
      DO 100 I=1,11
      AVEWT(I) = MNWT(I)
100    IYEAR(I) = YEAR(I)
      PRINT 110, (IYEAR(K),K=1,11)
110    FORMAT (1H1, '//////////', 18X, 'AVERAGE CALCULATED WEIGHT AT',
1      ' EACH ANNULUS AND AVERAGE WEIGHT AT CAPTURE', /,
21H0, 16X, 'AGE NUMBER CAPTURE', 9(6X,11), 2(5X,12) )
      PRINT 120
120    FORMAT (1H+, 14X, 97(1H_) )
      DO 150 I=1,M
      IF(NUM(I)) 150,150,130
130    PRINT 140, I,NUM(I),MNWTC(I),(MNCLWT(I,L),L=1,M)
140    FORMAT (1H0, 16X, I2, 3X, I4, 2X, I6, 1X, 11(1X, I6) )
150    CONTINUE
      PRINT 120
      PRINT 160, WMAGE,KTOTFH,JMWTC,(MNWT(I),I=1,M)
160    FORMAT (1H0, 14X, F5.1, 2X, I4, 2X, I6, 1X, 11(1X, I6) )
      PRINT 170
170    FORMAT (1H0, 16X, 'WEIGHTED MEANS' )
      PRINT 180
180    FORMAT (///)
      IF(LOOP.EQ.'NO')PRINT 190, NFISH
190    FORMAT ( /, 1H0, 14X, 'MALES ONLY OF', 1X(2A4) )
      IF(LOOP.EQ.'YES'.AND.KK.EQ.1)PRINT 200, NFISH
200    FORMAT ( /, 1H0, 14X, 'FEMALES ONLY OF ', 2A4 )
      IF(LOOP.EQ.'END'.AND.KK.EQ.2)PRINT 210, NFISH
210    FORMAT (/, 1H0, 14X, 'MALES AND FEMALES COMBINED OF ', 2A4 )
      IF(ITRACE.EQ.'YES')PRINT 90
      PRINT 220
C
C      GRAPH MADE OF AVERAGE CALCULATED WEIGHT AGAINST YEAR OF LIFE.
C
220    FORMAT (1H1)
      CALL GRAPH (YEAR, AVEWT, 11, 9, 6, 'NORMAL')
      PRINT 230, NFISH
230    FORMAT (1H , 22X, 2A4, ' WEIGHTED MEAN OF CALCULATED '
1      ' WEIGHT AGAINST YEAR OF LIFE (YEARS ON ABSCISSA)' )
      IF(LOOP.EQ.'NO')PRINT 190, NFISH
      IF(LOOP.EQ.'YES'.AND.KK.EQ.1)PRINT 200, NFISH
      IF(LOOP.EQ.'END'.AND.KK.EQ.2)PRINT 210, NFISH
      RETURN
      END

```

```

@FOR,ASI  SUBGRAPH
C
C   THIS GRAPHING ROUTINE IS USED THROUGHOUT FSHPAK.  IF THE USERS
C   MACHINE DOES NOT HAVE A LIBRARY GRAPH ROUTINE, THIS ONE CAN
C   BE LOADED AS LISTED HEREIN.  BOTH SCALE AND GRAPH MUST
C   BE LOADED.
C
C   SUBROUTINE GRAPH(X, Y, N, LLX, LLY, KNTRL)
C       BY MITCH BLANK, 11/68
C
C       X AND Y ARE THE ARRAYS TO BE GRAPHED
C       N IS THE NUMBER OF POINTS TO BE GRAPHED
C       LLX IS THE LENGTH OF THE X AXIS IN INCHES(INTEGER)
C       LLY IS THE LENGTH OF THE Y AXIS IN INCHES(INTEGER)
C       KONTRL IS THE SCALE CONTROL LITERAL
C           'USER' FOR USER SUPLIED SCALE
C           'DX=DY' FOR EQUAL X AND Y SCALING FACTOR
C           'NORMAL' FOR NORMAL SCALING
C
C       GLOBAL DEFINERS
C       DIMENSION X(1), Y(1)
C       LOCAL DEFINERS
C       DIMENSION LINE(103), XSCALE(11)
C       REAL HIGH, LOW, INVDLX
C       INTEGER WIDTH, EDGE, PLACE, RIGHT
C       INTEGER DECODE(3) /'NORMAL', 'USER', 'DX=DY'/
C       EQUIVALENCE (LINE(1), EDGE)
C       DATA LIT1/5H-----/, LIT2/5H----+/, LIT3/5H.    /
101 FORMAT (' *** ERROR IN CALL OF SUBROUTINE GRAPH ***')
102 FORMAT (10X, 103A1)
103 FORMAT (1HD)
104 FORMAT (1X, G9.2, 103A1)
105 FORMAT (10X, 2H.+21A5)
106 FORMAT (7X, 11(G9.2, 1X))
107 FORMAT (' *** ERROR: ', A6, ' IS NOT A VALID GRAPH CALL LITERAL
$**')
C
C       DO 180 J = 0, 2
C       IF ( KNTRL .EQ. DECODE(J + 1)) GO TO 182
180 CONTINUE
GO TO 910
182 KONTRL = J
LX = LLX
LY = LLY
185 WRITE (6, 103)
C       CHECK THE VALIDITY OF THE CALL
190 IF (LX .LE. 0 .OR. LY .LE. 0 .OR. N .LE. 0 .OR. LX .GT. 10)
$ GO TO 900

```

```

C          SCALE THE X AND Y AXIS
          IF (KONTRL .EQ. 1) GO TO 200
          CALL SCALE (X, N, FLOAT(LX), XMIN, DX)
          CALL SCALE (Y, N, FLOAT(LY), YMIN, DY)
          IF (KONTRL .EQ. 0) GO TO 210
          DX = MAX(DX, DY)
          DY = DX
          GO TO 210
200 XMIN = X(N + 1)
          YMIN = Y(N + 1)
          DX = X(N + 2)
          DY = Y(N + 2)
          IF (DX .LE. 0.0 .OR. DY .LE. 0.0) GO TO 900
C          INITIALIZE VARIOUS THINGS
210 DELTAY = DY / 6.
          HDELY = DELTAY * .5
          DELTAX = DX * .1
          INVDLX = 10. / DX
          LINES = LY * 6 + 1
          WIDTH = LX * 10 + 2
          RIGHT = WIDTH + 1
          NXSCL = LX + 1
C          SET UP THE X AXIS
          DO 250 J = 1, NXSCL
          XSCALE(J) = XMIN + DX * FLOAT(J-1)
          IF (ABS(XSCALE(J)) .LT. DELTAX) XSCALE(J) = 0.0
250 CONTINUE
C          PRINT OUT TOP AXIS
          WRITE (6, 106) (XSCALE(J), J = 1, NXSCL)
          WRITE (6, 105) (LIT1, LIT2, J = 1, LX), LIT3
C          MAIN ROUTINE, PRINT OUT THE GRAPH
          DO 600 L = 1, LINES
          YVALUE = YMIN + DELTAY * FLOAT (LINES - L)
          HIGH = YVALUE + HDELY
          LOW = YVALUE - HDELY
C          CLEAR THE OUTPUT LINE TO BLANKS
          DO 300 J = 2, WIDTH
300 LINE(J) = 1H
          DO 400 J = 1, N
          IF (Y(J) .GE. HIGH .OR. Y(J) .LT. LOW) GO TO 400
          PLACE = (X(J) - XMIN) * INVDLX + 2.5
          IF (PLACE .GE. 2 .AND. PLACE .LE. WIDTH) LINE(PLACE) = 1H*
400 CONTINUE
C          HERE COME DE JUDGE
          IF (MOD(L, 6) .EQ. 1) GO TO 500
          EDGE = 1H:
          LINE(RIGHT) = EDGE
          WRITE (6, 102) (LINE(J), J = 1, RIGHT)
          GO TO 600
500 EDGE = 1H+
          LINE(RIGHT) = EDGE
          IF (ABS(YVALUE) .LT. DELTAY) YVALUE = 0.0
          WRITE (6, 104) YVALUE, (LINE(J), J = 1, RIGHT)
600 CONTINUE
C          PRINT OUT BOTTOM AXIS

```

```

        WRITE (6, 105) (LIT1, LIT2, J = 1, LX), LIT3
        WRITE (6, 106) (XSCALE(J), J = 1, NXSC)
C      EXIT OF SUBROUTINE
800 WRITE (6, 103)
    RETURN
C      ERROR PRINTOUT ROUTINE
900 WRITE (6, 101)
    GO TO 800
910 WRITE(6, 107) KNTRL
    GO TO 800
C      SIMULATED GRAPH 2 ENTRY POINT
    ENTRY GRAPH2 (XL, YL, N, X, Y)
C
C      N IS THE NUMBER OF POINTS TO BE GRAPHED
C      X AND Y ARE THE ARRAYS TO BE GRAPHED
C      XL IS THE LENGTH OF THE X AXIS IN INCHES (FLOATING POINT)
C      YL IS THE LENGTH OF THE Y AXIS IN INCHES (FLOATING POINT)
C
    LX = XL
    LY = YL
    KONTRL = 0
    GO TO 185
    END
    SUBROUTINE XRANGE (ARRAY, N, XMIN, XMAX)
    DIMENSION ARRAY(1)
    XMIN=ARRAY(1)
    XMAX=XMIN
    DO 1 J=1, N
    IF (ARRAY(J).LT.XMIN)GOTO2
    IF (ARRAY(J).GT.XMAX)GOTO3
1    CONTINUE
    RETURN
2    XMIN=ARRAY(J)
    GOT01
3    XMAX=ARRAY(J)
    GOT01
    END
    SUBROUTINE SCALE (X, N, XL, XMIN, DX)
C    BY ROGER GERSONDE + MITCH BLANK
    DIMENSION X(1)
    LOGICAL FIRST
C    CALULATE XMIN + XMAX
    CALL XRANGE (X, N, XMIN, XMAX)
    FIRST = .TRUE.
70 RANGE = ABS(XMIN)
    IF(RANGE) 43, ,43
75 FIRST = .FALSE.
    RANGE      =(XMAX - XMIN)/XL
C    CHECK FOR NON-VARING DATA
41 IF(RANGE) 43,42,43
42 XMAX =      XMAX + 1.
    GO TO 75
C    FIND MAGNITUDE OF RANGE= POWER OF IT
43 POWER      =1.
    IF(RANGE-1.) 50,61,45

```

```
45 IF(RANGE-POWER) 60,61,46
46 POWER      =POWER*10.
   GO TO 45
50 POWER      =POWER*.1
   IF(RANGE-POWER) 50, 61, 61
60 POWER      =POWER*.1
C   MAKE POWER .GTR. RANGE
61 ADD        =POWER
62 IF(RANGE-POWER) 65,65,63
63 POWER      =POWER+ADD
   GO TO 62
65 IF (.NOT. FIRST) GO TO 72
66 IF(XMIN    ) 67,75 ,68
67 XMIN       =-POWER
   GO TO 75
68 XMIN       =POWER-ADD
   GO TO 75
72 DX         =POWER
   RETURN
   END
```

REQUEST FOR LITERATURE EXCHANGE

This report is being sent to you in the hope that literature exchange between you or your organization and this Center will be initiated and maintained. Reports or reprints which you may wish to designate for the Center Library, or inquiries concerning publications and exchanges, should be addressed to:

Center for Great Lakes Studies
Reference Library
University of Wisconsin-Milwaukee
Milwaukee, Wisconsin 53201

Publications designated for individuals within the Center should be addressed to those persons by name.

SPECIAL REPORT SERIES

Special Reports are issued from time to time, usually to provide accounts of work in progress, or completed, which are too detailed for acceptance by professional journals. This method of publication is, for example, convenient when it is desired to record and deposit collections of data in identifiable and recoverable form, or to fulfill the terms of contracts, or to make the details of a particular technique available to other potential users. This mode of reporting does not preclude later publication – usually abbreviated and selected – in professional journals; indeed, authors are urged to regard such publication as a necessary final step in the completion of a piece of literature. The following Special Reports have been issued to date:

- Mortimer, C. H. 1968. Internal waves and associated currents observed in Lake Michigan during the summer of 1963. Spec. Rept. No. 1; 24 p., 120 figs.
- Schenker, E. 1968. Effects of containerization on Great Lakes ports. Spec. Rept. No. 2; 45 p., 18 tables, 3 appendices.
- Abstracts, Eleventh Conference on Great Lakes Research, April 1968. Spec. Rept. No. 3; 83 p., 91 abstracts.
- Fee, E. J. 1968. Digital computer programs for the Defant method of seiche analysis. Spec. Rept. No. 4; 27 p., 4 appendices, 3 figs.
- Schenker, E. 1968. Future general cargo traffic and terminal requirements at the Port of Milwaukee. Spec. Rept. No. 5, 13 p.
- Fee, E. J. 1969. Digital computer programs for spectral analysis of time series. Spec. Rept. No. 6, 16 p., 3 appendices, 1 fig.

Gannon, J. E. 1969. Great Lakes plankton investigations: a bibliography. Spec. Rept. No. 7, 65 p. 2 appendices, 1 fig.

Gannon, J.E. and A. M. Beeton. 1969. Studies on the effects of dredged materials from selected Great Lakes harbors on plankton and benthos. Spec. Rept. No. 8, 80 p., 36 figs.