l of OCEANOGRAPHY

OREGON STATE UNIVERSITY

GEOPHYSICAL LIBRARY SUBROUTINES (*GLIB)

by

Ken Keeling
Michael Gemperle
and
Gerald Connard

Office of Naval Research
Contract N00014-67-A-0369-0007
Project NR 083-102

November 1975 Reference 75-20

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Reference # 75-20	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) GEOPHYSICAL LIBRARY SUBROUTINES (* GLIB)		5. TYPE OF REPORT & PERIOD COVERED Technical 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ken Keeling Michael Gemperle and Gerald Connard		8. CONTRACT OR GRANT NUMBER(s) N00014-67-A-0369-0007
9. PERFORMING ORGANIZATION NAME AND ADDRESS School of Oceanography Oregon State University Corvallis, Oregon 97331		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 083-102
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Ocean Science & Technology Division Arlington, Virginia 22217		12. REPORT DATE November 1975 13. NUMBER OF PAGES 44
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Geophysical Computer Subroutines

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report documents the various subroutines in general use by the Oregon State University Geophysics group. Included are subroutines for plotting and date and time conversion and arithmetic functions and control functions.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

School of Oceanography
Oregon State University
Corvallis, Oregon 97331

John V. Byrne
Dean

GEOPHYSICAL LIBRARY SUBROUTINES
(*GLIB)

by

Ken Keeling
Michael Gemperle
and
Gerald Connard

Office of Naval Research
Contract N00014-67-A-0369-0007
Project NR 083-102

International Decade of
Oceanography
NAZCA Plate Program
Contract GX28675

NOVEMBER, 1975

Reference Number 75-20

GTR 751101

TABLE OF CONTENTS

GEOPHYSICAL LIBRARY SUBROUTINES

I. Introduction

This report describes the various subroutines and functions in general use by either the Geophysics Group or the present Geophysics Data Reduction and Plotting Package (Gemperle, et al., 1976) that are not available on the OS-3 Fortran Library (OS-3 is the operating system on the Control Data Corporation 3300 computer at Oregon State University). These subroutines are available to users of the CDC-3300 on the saved file *GLIB.

These subroutines may be classified into five general categories.

1. Plotting - general purpose plotter driver subroutines and mercator map projection subroutines.
2. Date and Time - general purpose date and time conversion from the conventional system of year, month, day and hour, minute, second (or tenths of minutes) to integer tenths of minutes since the beginning of the year and the inverses of the above. These subroutines are generally used in computations involving time differences, time output and plotting. Various formats are available.
3. Arithmetic Functions - standard Fortran Library functions not available on the OS-3 Fortran Library.
4. Control Functions - file and operating system control functions peculiar to the OS-3 operating system.
5. Misc Subroutines - subroutines used by more than one geophysics program. These subroutines perform specific geophysically oriented functions.

II. Characteristics Common throughout the Library

This library was written to run on a CDC 3300 computer operating under the Oregon State Open Shop Operating System (OS-3). As such, several of the subroutines are only applicable to this system, particularly those within the Control Function Group.

Because OS-3 is a time sharing system, these subroutines are usually called by programs which are normally run in an interactive mode on a teletype terminal in our offices.

These subroutines appear on the library because they are of general enough purpose under OS-3 to be used by more than one program. Most are written in CDC assembly language (COMPASS), though some were written in OS-3 Fortran IV when it was applicable to the subroutine.

There are a few conventions made in the documentation of this library. The "Type" of the parameters may be integer (I) or floating point (F.P.) for the passing of values or single (A4) or double (A8) word variables used to pass Binary Coded Decimal (BCD) information. All function names appear in the parameter table as the output parameter for the function.

Users of these subroutines and functions should also refer to the CDC and OSU publications listed in the references relating to the OS-3 operating system, and FORTRAN and COMPASS languages on the OSU CDC-3300 computer.

III. Description of Individual Subroutines

A. Plotting Subroutines

The first four subroutines are the plotter drivers and are used to convert the pen movements in inches specified by the calling program calls into actual pen movement codes used by the plotter and for initialization and bookkeeping. There are no limit checks made on any of the plotter driver subroutine parameters in order to speed up execution, therefore, care should be used in the calling programs. The last two subroutines are used to generate mercator maps on a 30-inch incremental plotter.

Source listings for the first four subroutines are not included in the Appendix because of their length.

1. PLOTINT (X, Y, LUN)

PLOTINT is used to initialize the plotter subroutines and define the location of the origin of the axis. The parameters X and Y are with respect to wherever the pen is left by the operator and are assumed to be within six inches of the left edge of the plotter paper. The X coordinate is across the paper and the Y coordinate is in the long dimension of the paper. The normal call is: CALL PLOTINT (-6., 6., 10). This will cause the origin to be against the left edge of the plotter paper and at least six inches away from the last plot. This is normally followed by a CALL PLOT (1.0, 0., -3) to redefine the origin one inch from the left edge of the plotter paper. The logical unit, e.g. LUN 10, should have already been defined as a plotter with an OS-3 Control Mode "EQUIP" statement or Fortran call to the EQUIP subroutine (see Skinner, 1970).

Parameter	Type	Description
X	F.P.	The location of the X-coordinate of the axis origin in inches relative to where the pen was left by the operator.
Y	F.P.	The location of the Y-coordinate of the axis origin in inches relative to where the pen was left by the operator.
LUN	I	Number of the Logical Unit to which the pen movement codes generated by the plotter driver are sent

2. PLOT (X, Y, IPEN)

PLOT is used to move the pen to a new X, Y in inches with the pen either up or down. If the parameter 'IPEN' is negative, then a new origin is defined with respect to the last one defined by either PLOTINT or PLOT. Redefinition of the origin also causes any accumulated pen movements to be buffered to the plotter file. This means that the last call in a plotting program should be a call to PLOT with IPEN negative to insure that the last record is sent to the plotter.

Parameter	Type	Description
X	F.P.	The new X-coordinate in inches
Y	F.P.	The new Y-coordinate in inches
IPEN	I	IPEN = 2 pen down prior to moving
		IPEN = 3 pen up prior to moving
		IPEN = -3 last call to the plotter subroutines for this plot. This defines a new origin for subsequent plotting.

This is the basic pen movement subroutine. The algorithim for generating pen movement codes was taken from CACM Algorithim 162 (Stockton, 1963).

3. PLOTSYMB (X, Y, H, ARRAY, THETA, N)

PLOTSYMB is used to move the pen to the location of a centered symbol and draw that symbol or to move the pen to the lower left corner of a label and to write that label. Figure 1 gives the available centered symbols and the integer associated with each.

Parameter	Type	Description
X	F.P.	The new X-coordinate in inches
Y	F.P.	The new Y-coordinate in inches
H	F.P.	Height of character or centered symbol to be drawn in inches
ARRAY or ISYM	A_n or I	Real or INTEGER variable or ARRAY containing n BCD characters or an integer specifying the centered symbol to be drawn (see Fig.1)
THETA	F.P.	The angle measured in degrees from the X-axis in a counter-clockwise direction at which the centered symbol or label is to be written
N	I	N > 0 then N characters will be drawn, pen is lifted prior to moving to X, Y N = -1 the pen is lifted prior to moving to draw a centered symbol N = -2 the pen is dropped prior to moving to draw a centered symbol

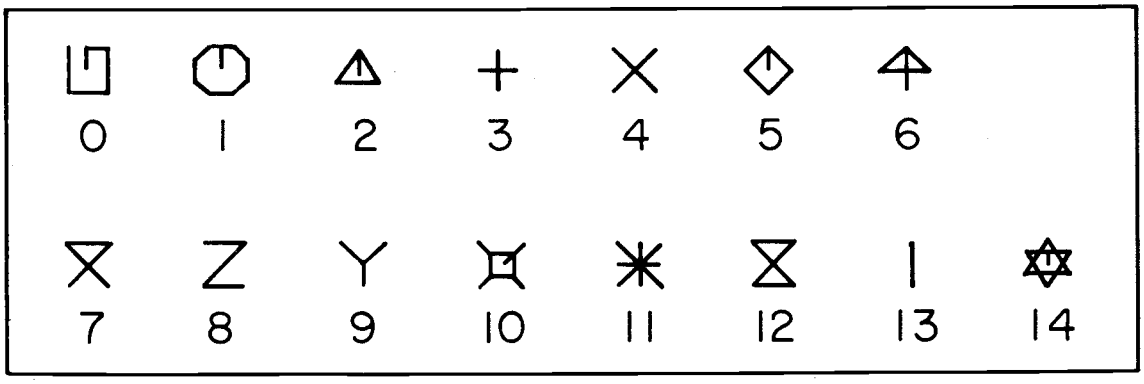

Figure 1 PLOTSYMB Centered Symbols

4. ROTATEXY

ROTATEXY is used to rotate the axis of the plot 90 degrees counter-clockwise resulting in the X-axis becoming parallel to the long dimension of the plotter paper. A call to this subroutine is irreversible and can be made only once during the running of a program. After a call to ROTATEXY a call to PLOTINT with the parameters (0., -30., LUN) should be made to reset the coordinate axis to the right edge of the plotter paper. If used, this subroutine should be the first plotter subroutine called or should only be made when the pen is at an origin. There are no parameters for this subroutine.

5. MERCMAP

MERCMAP is a general purpose subroutine used to generate an annotated Mercator grid on the plotter. It interactively asks the user for information concerning the latitude and longitude limits, scale in inches per degree of longitude and grid spacing in minutes as well as the data file name and plot label (Example 1, Page 7). Underlined characters are printed by the subroutine MERCMAP. MERCMAP will attempt to fit the grid on the plotter paper with the longitude lines parallel to the length of the paper. If the grid will not fit, the program will rotate the grid 90 degrees and try again. If the minimum dimension exceeds 26 inches, the program will ask for the map parameters again, enabling the user to split the map grid into sections or change the scale. MERCMAP calls the function MERIDPTS.

Subroutine MERCMAP contains the following declarative statements for the return of input values back to the calling program:

```
REAL  MINPTS, MAXPTS, LATINCH, LONGINCH, MAPSCALE
REAL  MSD60, MINLAT, MAXLAT, MINLONG, MAXLONG
COMMON  MINPTS, MAXPTS, LATINCH, LONGINCH, MAPSCALE
COMMON  MSD60, MINLAT, MAXLAT, MINLONG, MAXLONG
```

Parameter	Type	Description
MINPTS	F.P.	MERIDPTS (MINLAT)
MAXPTS	F.P.	MERIDPTS (MAXLAT)
LATINCH	F.P.	MAPSCALE* (MAXPTS-MINPTS)/60.
LONGINCH	F.P.	MAPSCALE* (MAXLONG-MINLONG)
MAPSCALE	F.P.	Inches per degree longitude
MSD60	F.P.	MAPSCALE/60.
MINLAT	F.P.	Decimal degrees for south edge of map
MAXLAT	F.P.	Decimal degrees for north edge of map
MINLONG	F.P.	Decimal degrees for west edge of map (0-360)
MAXLONG	F.P.	Decimal degrees for east edge of map (0-360)

```
LABEL PLOT FOR?
PETERSON

INPUT FILE NAME
Y69FEB

GIVE THE FOLLOWING:
   NLAT    SLAT    WLONG    ELONG   SCALE GRID
  3D00M -4D00M -90D00M -80D00M 4.000   60M

DO YOU WANT A RAPIDOGRAPH PEN - YES OR NO
NO
```

Example 1
MERCMAP OUTPUT

6. MERIDPTS (ALAT)

MERIDPTS is a real function which returns the value of the meridional parts equivalent to the latitude ALAT according to the Clark spheroid of 1866 from the formula (Bowditch, 1958, p. 1186):

$$M = a \log_{\varepsilon} 10 \log \tan (45° + \frac{L}{2}) - a(e^2 \sin L + \frac{e^4}{3} \sin^3 L + \frac{e^6}{5} \sin^5 L)$$

where

M is the number of meridional parts between the equator and the given latitude,

a is the equatorial radius of the earth in minutes of arc,

L is the latitude in radians,

f is the flattening of the earth, and

e is the ellipticity ($e = \sqrt{2f-f^2}$).

MERIDPTS is used by the mercator map plotting programs (Gemperle, et.al., 1976) to determine the location of data points to be plotted and by MERCMAP for drawing mercator grid. MERIDPTS must be declared a REAL function in the calling program.

Parameter	Type	Description
ALAT	F.P.	Latitude in decimal degrees
MERIDPTS	F.P.	OUTPUT - equivalent to M above

B. Date and Time Subroutines

The first five of these date and time subroutines comprise a package for the conversion of date and time to integer tenths of minutes from the beginning of the year and back in various formats. The sixth gets the current time from the computer in A8 Format and the seventh zeros the blanks in either A4 or A8 Format BCD variables.

1. LEAPYEAR (IYR)

LEAPYEAR initializes a month table to a non-leapyear and then checks the parameter IYR (a two digit year in the twentieth century) to see if it is a leapyear and, if so, modifies the month table to reflect the leapyear. The month table MTABLE is in LABELED COMMON for use by the rest of

the date and time subroutines in this package. The starting year IYR modified to become a four digit year is also saved in the same LABELED COMMON. This should be the first subroutine called when using these subroutines to initialize MTABLE.

Parameter	Type	Description
IYR	I	Integer year (two digit)
MTABLE	I	OUTPUT - Array dimensioned (12) in LABELED COMMON which contains the Julian day number of the last day of the preceeding month
ISYEAR	I	OUTPUT - Integer starting year (4 digit) follows MTABLE in the LABELED COMMON

2. NGDCTIME (YRMODAY, HRMN10)

The function NGDCTIME returns the integer tenths of minutes since the first of the year (ISYEAR). This function requires that the month table MTABLE has been initialized by a call to LEAPYEAR. A fatal error message is written if the parameters refer to a time later than February 28 of the year following ISYEAR initialized by LEAPYEAR.

Parameter	Type	Description
YRMODAY	I	Six digit year, month and day
HRMN10	I	Five digit hour, minute and tenths of minutes
NGDCTIME	I	OUTPUT - integer tenths of minutes since the first of the year specified by LEAPYEAR (ISYEAR)

3. ITIME (YRMODAY, HRMNSC)

This function is exactly the same as NGDCTIME except that the second parameter contains seconds instead of tenths of minutes. The value returned is still tenths of minutes since the first of the year. This function also requires that the month table MTABLE has been initialized by a call to LEAPYEAR. A fatal error message is written if the parameters

refer to a time later than February 28 of the year following ISYEAR initialized by LEAPYEAR.

Parameter	Type	Description
YRMODAY	I	Six digit integer year, month and day
HRMNSC	I	Six digit integer hour, minute and second
ITIME	I	OUTPUT - integer tenths of minutes since the first of the year specified by LEAPYEAR (ISYEAR)

4. IDTG (IYR, ITIME, YRMODAY, HRMN10)

This subroutine converts the two digit year (IYR) and time (ITIME) since the beginning of the year in tenths of minutes back to year, month, day (YRMODAY) and hour, minute and tenths of minutes (HRMN10). The parameter ITIME is usually the output from the function NGDCTIME or ITIME.

Parameter	Type	Description
IYR	I	Integer year (two digit)
ITIME	I	Integer tenths of minutes since the beginning of the year (output from ITIME or NGDCTIME)
YRMODA	I	Six digit integer year, month and day
HRMN10	I	Five digit integer hour, minute and tenths of minutes

5. ADTG (YR, ITIME, YRMODAY, HRMN10)

This subroutine is the same as IDTG except that the output is in BCD; useful, for example, for annotating plots. The output is BCD with leading zeros inserted by the function ZB (see following).

Parameter	Type	Description
YR	I	Integer year (two digit)
ITIME	I	Integer tenths of minutes since the beginning of the year (output from ITIME or NGDCTIME)
YRMODAY	A6	Six character year, month and day in BCD
HRMN10	A5	Five character hour, minute and tenths of minutes in BDC

6. ARMYTIME (X)

The subroutine ARMYTIME returns the time of day from the CDC 3300 clock in the real variable parameter "X" in BCD. The format is bbHHMMbb where b is a blank, H is an hour digit and M is a minute digit.

Parameter	Type	Description
X	A8	The time of day between 0000 and 2400 is returned in BCD

7. IZB (I) or ZB (A)

This subroutine with entry points for both a real BCD argument or an integer BCD argument, zeros the blanks of the argument. For example, it is used to change blanks to zeros in date and time BCD variables.

Parameter	Type	Description
I	A4	Integer BCD argument for the integer function IZB
A	A8	Real BCD argument for the real function ZB
IZB	A4	OUTPUT - blanks are zeroed
ZB	A8	OUTPUT - blanks are zeroed

C. Arithmetic Functions

These three arithmetic functions are standard Fortran Library functions but are not included on the OS-3 Fortran Library. We have, therefore, implemented them for the geophysics library.

1. AINT (X)

AINT returns the integer portion of a floating point number. It is equivalent to: AINT (X) = FLOAT (IFIX (X)) (McCracken, 1965).

Parameter	Type	Description
X	F.P.	The real argument
AINT	F.P.	OUTPUT - the sign times the largest integer less than or equal to the absolute value of X

2. ATAN2 (Y, X)

ATAN2 computes the value of the arctangent of Y/X returning a value between -PI and +PI. It simply extends the range of the single parametered ATAN function. The OS-3 Fortran Library function ATAN (CDC, 1965) is used to compute the arctangent in the range $-\pi/2$ to $+\pi/2$. ATAN2 determines the proper quadrant and modifies the angles accordingly (IBM, 1968).

Parameter	Type	Description
Y	F.P.	The ordinate
X	F.P.	The abscissa
ATAN2	F.P.	OUTPUT - Arctangent (Y/X) between -PI and +PI

3. AMOD (X, Y)

AMOD returns the value of X modulus Y for two real arguments. This is equivalent to AMOD (X, Y)=X - AINT (X/Y)*Y. This is a real valued, real parameter version of the MOD function (IBM, 1968).

Parameter	Type	Description
X	F.P.	The first parameter
Y	F.P.	The second parameter
AMOD	F.P.	Output - equivalent to X-AINT (X/Y)*Y

D. Control Functions

These functions interact with the OS-3 operating system and are peculiar to it. They do such things as UNEQUIP files, call the operating system, and zero core memory (also see Skinner, 1970).

1. FWSP (LUN)

The subroutine FWSP causes the present position of the specified LUN to advance one physical record. It is only legal when the LUN is EQUIPPED to an input device.

Parameter	Type	Description
LUN	I	Logical Unit Number of the input device to be forward spaced one physical record

2. ISTATUS (LUN)

This integer function returns the Logical Unit Number status to the calling program.

Parameter	Type	Description
LUN	I	Logical Unit Number to which the returned status applies
ISTATUS	I	Bits are returned as defined below

The upper three octal digits contain LUN status as follow:

BITS	STATUS
40000000	File protected
20000000	Load point
10000000	End of data
04000000	End of file
02000000	Destructive read
01000000	Binary record
00400000	Abnormal/Unavailable
00200000	Seek error
00100000	Saved file

The lower four bits contain the hardware type as follow:

0000	Unit not equipped
0001	File
0010	Line printer
0011	Card punch
0100	Card reader
0101	Mag tape
0110	Teletype
0111	Plotter
1000	Null
1001	TV
1010	Random access file
1011	Task

3. RELEASE (LUN)

RELEASE deletes all of the data associated with the Logical Unit Number (LUN). The Logical Unit (LUN) should be equipped to a file or an output device such as a card punch, line printer or plotter. This function is useful for killing the data going to the plotter or line printer prior to unequipping the LUN.

Parameter	Type	Description
LUN	I	Logical Unit Number of the file or output device from which the data is to be released

4. RESET

RESET does an OS-3 Control Mode "UNEQUIP" to all logical units between 0 and 49 inclusive. It is often used by programs with internal calls to the EQUIP and UNEQUIP subroutines. There are no parameters for this subroutine.

5. MI

MI stores the address of the last Fortran call to MI in location 13_8 enabling a user to resume his program at the point of the last call to MI by entering OS-3 Control Mode and typing "MI". It is often used by calling programs to recover from normally fatal errors in the typing of data file names. There are no parameters to this subroutine.

6. ZAP (IPAGE)

This subroutine zeros a page (2048 words) of core on the CDC 3300 running under the OS-3 operating system. There are sixteen pages of core (32,768 words) normally available to a user running Fortran, numbered 0-15.

Parameter	Type	Description
IPAGE	I	The page to be zeroed normally 0-15

7. SBJP

SBJP returns the user to OS-3 control mode, effectively terminating the execution of the job. It is equivalent to "CALL EXIT" except that the "END OF FORTRAN EXECUTION" message is not sent to the standard output device. There are no parameters and the user is left in OS-3 Control Mode. This subroutine is often used to abort the users job on a fatal error.

E. Misc Subroutines

These subroutines preform a specific function for geophysics programs. The first converts the revolutions of the R/V YAQUINA's engines to speed in knots and the second accesses the Mathews table (*MATAB) of corrections for the velocity of sound in water.

1. SPEED (ISREV, IPREV, NDATE, NTIME)

This subroutine computes the speed in knots of the R/V YAQUINA given the starboard and port engine revolutions. The date and time parameters are used for error messages. A new speed function will be generated for the R/V WECOMA in early 1976.

Parameter	Type	Description
ISREV	I	Starboard engine revolutions
IPREV	I	Port engine revolutions
NDATE	I	Integer date for output with error messages
NTIME	I	Integer time for output with error messages
SPEED	F.P.	Output - speed returned between 0.0 and 10.3 knots

2. GETMATAB (MZ, ITAB(2), IEND)

GETMATAB reads the Mathews zone table (Bialec, 1966) for the area "MZ" from the Random Access File (RAF) "*MATAB", and puts it in the integer

array "ITAB". If the area "MZ" is not on the RAF, "*MATAB", then an error message is written and the user is aborted. Presently Mathews zones 23-25 and 41-45 are included on the saved public random access file "*MATAB". The table *MATAB contains corrections to the velocity of sound assuming an initial velocity of 1500 meters per second (See appendix B). A sample program (ZORK) for reducing fathoms to corrected meters is included on page 43.

Parameter	Type	Description
MZ	I	The Mathews zone to be read from the RAF
ITAB	I	OUTPUT - The array (dimensioned (60)) to return the Mathews zone table
IEND	I	OUTPUT - The integer length of the mathews zone table "MZ"

IV. Acknowledgments

This work was supported by the Office of Naval Research through contract N00014-67-A-0369-0007 under project NR 083-102, the National Science Foundation International Decade of Oceanography NAZCA Plate Program Contract GX28675, and the OSU School of Oceanography Industrial Cooperative Program.

V. References

Bialek, Eugene L., 1966, Handbook of Oceanographic Tables, U.S. Naval Oceanographic Office, Special Pub. 68.

Bowditch, Nathaniel, 1958, American Practical Navigator, U.S. Navy Hydrographic Office Pub. No. 9, 1524 p.

Control Data Corporation, 1965, Computer Systems Library Routines, Control Data Corporation Pub. No. 60058100, 33p.

Control Data Corporation, 1967, Compatible Compass, Control Data Corporation Pub. No. 60174000, A, 62 p.

Control Data Corporation, 1969, FORTRAN Reference Manual, Control Data Corporation Pub. No. 60057600, Rev. B, 116 p.

Gemperle, Michael, G. Connard and K. Keeling, 1976 (in preparation), Processing & Display of Marine Geophysical Data, Oregon State University School of Oceanography.

International Business Machines, 1968, IBM System/360 Fortran IV Library Subprograms, Form C28-6596-4, 66 p.

McCracken, Daniel D., 1965, A Guide to Fortran IV Programming, John Wiley & Sons, Inc., 151 p.

Massie, Water W. 1973. FORTRAN REFERENCE MANUAL. Oregon State University Computer Center Pub. CCM-73-04, August 1973.

Skinner, David. 1970. OS-3 REFERENCE MANUAL. Oregon State University Computer Center Pub. CCM-70-8. 59 p.

Stockton, Fred G., 1963, Algorithm 162 XYMOVE PLOTTING Communications of the ACM, vol. 6(4):161.

APPENDIX A

```
OS3 FORTRAN  VERSION 3.13          11/05/75  2243

1             SUBROUTINE  MERCMAP
2          C  THIS SUBROUTINE DRAWS A MERCATOR GRID ON THE PLOTTER.  IT
3          C  QUERRIES THE USER FOR THE MAP PARAMETERS AND LEAVES THEM IN
4          C  COMMON FOR THE CALLING PROGRAM.
5          C  SOUTHERN LATITUDES AND WESTERN LONGITUDES ARE NEGATIVE.
6          C  MERCMAP WILL WORK ACROSS BOTH THE DATELINE AND GREENWICH
7          C  MERIDIAN.
8                COMMON  MINPTS, MAXPTS, LATINCH, LONGINCH, MAPSCALE
9                COMMON  MSDEG, MINLAT, MAXLAT, MINLONG, MAXLONG
10               REAL    MINPTS, MAXPTS, LATINCH, LONGINCH, MAPSCALE
11               REAL    MSDEG, MINLAT, MAXLAT, MINLONG, MAXLONG
12               REAL    LAT, MIN, MERIDPTS
13               DIMENSION  BCD(8), ABCD(10)
14               DIMENSION IALABEL(2), FORM(4), FORMR(3)
15               EQUIVALENCE (ALABEL,IALABEL)
16               EQUIVALENCE (BCD,ABCD)
17               INTEGER  PENUP, PENDOWN
18         C  FOR RAPIDOGRAPH PEN ON THE PLOTTER
19               FORMR (1) = 8H MEDIUM
20               FORMR (2) = 8HRAPIDOGR
21               FORMR (3) = 8HAPH PEN.
22         C  FOR PLOTS LONGER THAN 60 INCHES.
23               FORM (1) = 8HFOLLOWIN
24               FORM (2) = 8HG PLOT
25               FORM (4) = 8H INCHES.
26         C  CODES FOR THE PLOT DRIVER SUBROUTINES
27               PENDOWN = 2
28               PENUP = 3
29         C  PUT DATE AND TIME INTO THE LABEL ARRAY
30               CALL DATE (ABCD(9))
31               CALL ARMYTIME (ABCD(10))
32         C  DON'T GET ABORTED BECAUSE LUNS WERE LEFT EQUIPPED
33               CALL RESET
34               CALL EQUIP (17,8HLP      )
35               WRITE (61,105)
36         C  LABEL THE PLOT AND LP SO YOU CAN GET IT BACK
37           105 FORMAT (/0LABEL PLOT FOR^/)
38               READ (60,104) BCD
39               WRITE (17,107) ABCD
40         C  GET THE DATA FILE
41               WRITE (61,108)
42               WRITE (17,108)
43           108 FORMAT (/0INPUT FILE NAME/)
44               READ (60,104) ANAME
45               WRITE (17,107) ANAME
46               CALL EQUIP (1,ANAME)
47               CALL EQUIP (10,8HPLOT    )
48            10 WRITE (61,100)
49               WRITE (17,100)
50         C  LONGS WEST ARE NEGATIVE AND LATS SOUTH ARE NEGATIVE
```

```
      OS3 FORTRAN  VERSION 3.13 MERCMAP   11/05/75  2243

51          100 FORMAT (≠0GIVE THE FOLLOWING:≠,/,
52             1        ≠   NLAT   SLAT   WLONG   ELONG  SCALE GRID≠,/,
53             2        ≠≠   D  M   D  M    D  M    D  M   .       M≠)
54              READ (60,101) MAXLAT, MAXLATM, MINLAT, MINLATM,
55             1              MINLONG, MINLONGM, MAXLONG, MAXLONGM,
56             2              MAPSCALE, DEC, MIN
57          101 FORMAT (F3.0,X,I2,X,F3.0,X,I2,X,F4.0,X,I2,X,F4.0,X,I2,
58             1        X,F2.0,X,F3.3,F4.0)
59          C  THIS IS A FUNNY WAY TO GET MAPSCALE, BUT IT IGNORES THE
60          C  DECIMAL POINT, AS IT MAY NOT BE THERE.
61              MAPSCALE = MAPSCALE + DEC
62              WRITE (17,109) MAXLAT, MAXLATM, MINLAT,
63             1               MINLATM, MINLONG, MINLONGM,
64             2               MAXLONG, MAXLONGM, MAPSCALE,
65             3               MIN
66          109 FORMAT(≠+≠,F3.0,≠D≠,I2,≠M≠,F3.0,≠D≠,I2,≠M≠,F4.0,≠D≠,I2,
67             1       ≠M≠,F4.0,≠D≠,I2,≠M≠,F6.3,F4.0)
68          C  MAKE DECIMAL DEGREES OUT OF THE DEGREES AND MINUTES.
69              MINLAT = MINLAT + SIGNF (FLOAT(MINLATM),MINLAT) / 60.
70              MAXLAT = MAXLAT + SIGNF (FLOAT(MAXLATM),MAXLAT) / 60.
71              MINLONG = MINLONG + SIGNF (FLOAT(MINLONGM),MINLONG) / 60.
72              MAXLONG = MAXLONG + SIGNF (FLOAT (MAXLONGM),MAXLONG) / 60.
73              IF (MAXLAT .LE. MINLAT) GO TO 19
74          C  MAKE LONGS 0 TO 360 TO THE EAST
75              IF (MINLONG .LT. 0.) MINLONG = 360. + MINLONG
76              IF (MAXLONG .LT. 0.) MAXLONG = 360. + MAXLONG
77          C  TAKE CARE OF CROSSING GREENWICH - MAKE LONGS GO TO 720 DEGS.
78              IF (MINLONG .LT. MAXLONG) GO TO 12
79              MAXLONG = MAXLONG + 360.
80          C  CALCULATE MAP VARIABLES
81             12 MIN = MIN / 60.
82              MSD60 = MAPSCALE / 60.
83              LONGINCH = MAPSCALE *  (MAXLONG - MINLONG)
84              MINPTS = MERIOPTS (MINLAT)
85              MAXPTS = MERIOPTS (MAXLAT)
86              LATINCH = MSD60 * (MAXPTS - MINPTS)
87          C  TAKE CARE OF FELT TIP PENS
88             15 WRITE (61,117)
89              WRITE (17,117)
90            117 FORMAT (≠0DO YOU WANT A RAPIDOGRAPH PEN - YES OR NO≠)
91              READ (60,114) IANS
92            114 FORMAT (A4)
93              WRITE (17,116) IANS
94            116 FORMAT (X,A4)
95              IF (IANS .EQ. 4HNO  ) GO TO 17
96              IF (IANS .EQ. 4HYES ) GO TO 16
97              WRITE (61,115)
98              WRITE (17,115)
99            115 FORMAT (≠0ILLEGABLE RESPONSE - TRY AGAIN≠)
100             GO TO 15
```

```
      OS3 FORTRAN  VERSION 3.13 MERCMAP   11/05/75  2243

101          16 CALL FORMS (10, FORMR)
102        C  INITIALIZE THE PLOTTER
103           17 CALL PLOTINT (-6., 6., 10)
104              CALL PLOT (1., 1., -3)
105              CALL PLOTSYMB (0., -2., .21, ABCD, 0., 80)
106        C  INITIALIZE PEN POSITIONS
107              YORG = 0.
108              XORG = 0.
109        C  IF MAP WILL FIT WITH LONGITUDES PARALLEL TO THE PLOTTER
110        C  PAPER ROLL GO TO 25
111              IF (LONGINCH .LE. 27.) GO TO 25
112        C  IF MAP WILL FIT WITH LATITUDES PARALLEL TO THE PLOTTER
113        C  PAPER ROLL GO TO 20
114              IF (LATINCH .LE. 27.) GO TO 20
115        C  SORRY ABOUT THAT -- GUESS AGAIN
116              WRITE (61,113)LATINCH,LONGINCH
117              WRITE (17,113) LATINCH, LONGINCH
118            113 FORMAT(≠ MAP IS ≠,F7.1,≠ BY ≠,F7.1)
119              WRITE (61,102)
120              WRITE (17,102)
121            102 FORMAT(≠- ONE DIMENSION MUST BE LE 27 IN. -- TRY AGAIN≠)
122              GO TO 10
123             19 WRITE (61,110)
124              WRITE (17,110)
125            110 FORMAT (≠0ERROR IN LATITUDES -- TRY AGAIN≠)
126              GO TO 10
127        C  ROTATE THE X-Y AXIS 90 DEGREES COUNTER CLOCK-WISE
128             20 CALL PLOT (0., 0., -3)
129              CALL ROTATEXY
130              YORG = -30.
131              XORG = 0.
132        C  CALCULATE THE LENGTH OF THE PLOT AND LET THE OPERATORS
133        C  KNOW IF GREATER THAN 60 INCHES.
134             25 AMAX = LONGINCH
135                IF (LATINCH .GT. LONGINCH) AMAX = LATINCH
136                ALENGTH = AMAX + 16
137                ENCODE (8,111,FORM(3)) ALENGTH
138            111 FORMAT (F8.0)
139                IF (ALENGTH .GT. 60.) CALL FORMS (10,FORM)
140        C  GET THE ORIGIN FOR THE MAP GRID
141                CALL PLOT (XORG, YORG, -3)
142                CALL PLOT (0., 1., -3)
143        C  LETS LABEL IT AGAIN WITH INFORMATION CONCERNING THE MAP,
144        C  AREA, DATA, ETC.
145                WRITE (61,103)
146                WRITE (17,103)
147            103 FORMAT (≠0GIVE THE PLOT LABEL -- UP TO 80 CHARS.≠)
148                READ (60,104) (ABCD(I),I=1,10)
149                WRITE (17,107) (ABCD(I),I=1,10)
150            107 FORMAT (≠ ≠,11A8)
```

```
OS3 FORTRAN  VERSION 3.13 MERCMAP   11/05/75  2243

151         104 FORMAT (10A8)
152             YLINE = 0.
153             CALL PLOTSYMB (0.,-1.,.21,BCD,0.,80)
154       C  FROM HERE TO L.00040 PLOT AND ANNOTATE THE LONGITUDES
155             GO TO 31
156          30 YLINE = YLINE + MAPSCALE * MIN
157       C  PLOT A LONGITUDE FROM NORTH TO SOUTH
158          31 CALL PLOT (YLINE, LATINCH, PENUP)
159             CALL PLOT (YLINE, 0., PENDOWN)
160             TEMP = MINLONG + YLINE / MAPSCALE
161             IF (TEMP .GT. 180.) TEMP = TEMP - 360.
162             ITEMP1 = TEMP + SIGNF (.01, TEMP)
163             ITEMP2 = ABS(TEMP - ITEMP1) * 60. + .01
164             ENCODE (8,106,ALABEL) ITEMP1, ITEMP2
165             IF (.NOT. ITEMP2) IALABEL(2) = IALABEL(2) + 60B
166         106 FORMAT (I5,X,I2)
167             CALL PLOTSYMB (YLINE-.5,-.25,.14,ALABEL,0.,8)
168             YLINE = YLINE + MAPSCALE * MIN
169             IF (MIN .EQ. 0.0) YLINE = LONGINCH
170             IF (YLINE - LONGINCH .GT. .1) GO TO 40
171       C  PLOT A LONGITUDE FROM SOUTH TO NORTH
172             CALL PLOT (YLINE, 0.,PENUP)
173             CALL PLOT (YLINE,LATINCH, PENDOWN)
174             TEMP = MINLONG + YLINE / MAPSCALE
175             IF (TEMP .GT. 180.) TEMP = TEMP - 360.
176             ITEMP1 = TEMP + SIGNF (.01, TEMP)
177             ITEMP2 = ABS(TEMP - ITEMP1) * 60. + .01
178             ENCODE (8,106,ALABEL) ITEMP1, ITEMP2
179             IF (.NOT. ITEMP2) IALABEL(2) = IALABEL(2) + 60B
180             CALL PLOTSYMB (YLINE-.5,LATINCH+.25,.14,ALABEL,0.,8)
181             IF (YLINE - LONGINCH .LT. -.1) GO TO 30
182       C  FROM HERE TO L.00060 PLOT AND ANNOTATE THE LATITUDES
183          40 CALL PLOT (LONGINCH, 0., PENUP)
184             LAT = MINLAT
185             GO TO 51
186          50 LAT = LAT + MIN
187          51 XLINE = MSD60 * (MERIDPTS (LAT) - MINPTS)
188       C  PLOT A LATITUDE FROM EAST TO WEST
189             CALL PLOT (LONGINCH, XLINE, PENUP)
190             CALL PLOT (0.,XLINE, PENDOWN)
191             ITEMP1 = LAT + SIGNF(.01,LAT)
192             ITEMP2 = ABS(LAT - ITEMP1) * 60. + .01
193             ENCODE (8,106,ALABEL) ITEMP1, ITEMP2
194             IF (.NOT. ITEMP2) IALABEL(2) = IALABEL(2) + 60B
195             IF (.NOT. ITEMP1 .AND. .NOT. ITEMP2) ALABEL = 8H EQUATOR
196             CALL PLOTSYMB (-1.,XLINE,.14,ALABEL,0.,8)
197             LAT = LAT + MIN
198             IF (MIN .EQ. 0.0) LAT = MAXLAT
199             IF (LAT - MAXLAT .GT. .1) GO TO 60
200             XLINE = MSD60 * (MERIDPTS (LAT) - MINPTS)
```

```
        OS3 FORTRAN  VERSION 3.13 MERCMAP   11/05/75  2243

  201       C  PLOT A LATITUDE FROM WEST TO EAST
  202             CALL PLOT (0.,XLINE, PENUP)
  203             CALL PLOT (LONGINCH, XLINE, PENDOWN)
  204             ITEMP1 = LAT + SIGNF(.01,LAT)
  205             ITEMP2 = ABS(LAT - ITEMP1) * 60. + .01
  206             ENCODE (8,106,ALABEL) ITEMP1, ITEMP2
  207             IF (.NOT. ITEMP2) IALABEL(2) = IALABEL(2) + 60B
  208             IF (.NOT. ITEMP1 .AND. .NOT. ITEMP2) ALABEL = 8H EQUATOR
  209             CALL PLOT SYMB (LONGINCH,XLINE,.14,ALABEL,0.,8)
  210             IF (LAT - MAXLAT .LT. -.1) GO TO 50
  211          60 CONTINUE
  212             RETURN
  213             END
LENGTH OF SUBPROGRAM      01746
LENGTH OF COMMON          00024
```

```
      OS3 FORTRAN  VERSION 3.13          11/05/75  2242

    1            FUNCTION MERIDPTS(LAT)
    2      C  THIS FUNCTION RETURNS A VALUE FOR THE MERIDIONAL PARTS
    3      C  EQUIVALENT TO THE LATITIDE ACCORDING TO THE CLARK SPHEROID OF
    4      C  1866.  (BOWDITCH 1958, PAGE 1186).
    5            REAL MERIDPTS, LAT
    6            RADLAT = LAT / 57.29578
    7            SINE = SIN(RADLAT)
    8            MERIDPTS = 7915.704468 * ALOG10(TANF(.785398+RADLAT/2.))
    9           1           -23.268932 * SINE
   10           2           -0.0525000 * SINE * SINE * SINE
   11            RETURN
   12            END
LENGTH OF SUBPROGRAM      00076
```

```
      OS3 FORTRAN  VERSION 3.13          11/05/75  2239

    1             SUBROUTINE LEAPYEAR(YR)
    2       C  THIS SUBROUTINE CHANGES THE MONTH TABLE FOR LEAPYEARS
    3       C  INITIALIZES MONTH TABLE TO THE NORMAL YEAR SO THAT IT
    4       C  IS RECALLABLE.
    5             COMMON /MTABLE/ MTABLE(12), ISYEAR
    6             INTEGER  YEAR, YR
    7             MTABLE (1) = 0
    8             MTABLE (2) = 31
    9             MTABLE (3) = 59
   10             MTABLE (4) = 90
   11             MTABLE (5) = 120
   12             MTABLE (6) = 151
   13             MTABLE (7) = 181
   14             MTABLE (8) = 212
   15             MTABLE (9) = 243
   16             MTABLE (10) = 273
   17             MTABLE (11) = 304
   18             MTABLE (12) = 334
   19       C  ONLY 2 DIGIT YEAR IN CALL
   20             YEAR = YR + 1900
   21       C  ISYEAR IS STARTING YEAR USED BY NGDCTIME, ITIME, ADTG, IDTG.
   22             ISYEAR = YR
   23             IF (MOD (YEAR, 4)) RETURN
   24             IF (.NOT. MOD (YEAR, 100)) GO TO 30
   25       C  ADD 1 TO MONTHS AFTER JAN FOR LEAPYEARS
   26          10 DO 20 I = 3, 12
   27          20 MTABLE (I) = MTABLE (I) + 1
   28             RETURN
   29          30 IF (.NOT. MOD (YEAR, 400)) GO TO 10
   30             RETURN
   31             END
LENGTH OF SUBPROGRAM      00127
LENGTH OF COMMON MTABLE   00015
```

```
      OS3 FORTRAN  VERSION 3.13          11/05/75  2239

    1            FUNCTION NGDCTIME(YRMODAY, HRMIN10)
    2       C  ENTER WITH YEAR MONTH DAY AND HOUR MINUTE * 10.  EXIT WITH
    3       C  INTEGER (MIN * 10) SINCE BEGINING OF YEAR.  IF YEAR CHANGES
    4       C  FROM START OF YEAR NGDCTIME IS FROM FIRST YEAR.
    5       C  ISYEAR IS THE STARTING YEAR, IT IS SET BY LEAPYEAR.
    6       C  IF LEAP YEAR IS NOT CALLED, IE FOR ONE TIME PROGRAMS ISYEAR
    7       C  DEFAULTS TO 1999 NOT A LEAP YEAR
    8            COMMON /MTABLE/ MTABLE(12), ISYEAR
    9            DATA (MTABLE = 0, 31, 59, 90, 120, 151, 181, 212, 243,
   10           1                273, 304, 334)
   11            DATA  (ISYEAR = 99)
   12            INTEGER YRMODAY, HRMIN10, YR, DAY, HR
   13       C  EXTRACT THE YEAR, MONTH AND DAY
   14            YR = YRMODAY / 10000
   15            MON = (YRMODAY - YR * 10000) / 100
   16            DAY = MOD (YRMODAY, 100)
   17       C  EXTRACT THE HOURS AND TENTHS OF MINUTES
   18            HR = HRMIN10 / 1000
   19            MIN = (HRMIN10 - HR * 1000)
   20       C  COMPUTE TENTHS OF MINUTES SINCE FIRST OF YEAR
   21            NGDCTIME = ((MTABLE(MON) + DAY-1) * 24 + HR) * 600 + MIN
   22       C  FOR THE SECOND YEAR
   23            IF (YR .GT. ISYEAR) NGDCTIME = NGDCTIME + (MTABLE(12)+31) * 14400
   24       C  CHECK FOR LATER THAN FEB 28, 23:59:50 OF SECOND YEAR
   25            IF (NGDCTIME .LT. 6119998) RETURN
   26            WRITE (61, 100)
   27          100 FORMAT (≠-TIME LATER THAN FEB 28, 23:59:50 OF SECOND YR≠)
   28            CALL SBJP
   29            END
LENGTH OF SUBPROGRAM      00135
LENGTH OF COMMON MTABLE   00015
```

```
      OS3 FORTRAN  VERSION 3.13          11/05/75  2239

    1            FUNCTION ITIME(YRMODAY, HRMINSEC)
    2       C  ENTER WITH YEAR MONTH DAY AND HOUR MINUTE SECOND.  EXIT WITH
    3       C  INTEGER (MIN * 10) SINCE BEGINING OF YEAR.  IF YEAR CHANGES
    4       C  FROM START OF YEAR ITIME IS FROM FIRST YEAR.
    5       C  ISYEAR IS THE STARTING YEAR, IT IS SET BY LEAPYEAR.
    6       C  IF LEAP YEAR IS NOT CALLED, IE FOR ONE TIME PROGRAMS ISYEAR
    7       C  DEFAULTS TO 1999 NOT A LEAPYEAR.
    8            COMMON /MTABLE/ MTABLE(12), ISYEAR
    9            DATA (MTABLE = 0, 31, 59, 90, 120, 151, 181, 212, 243,
   10           1                273, 304, 334)
   11            DATA  (ISYEAR = 99)
   12            INTEGER YRMODAY, HRMINSEC, YR, DAY, HR, SEC
   13       C  EXTRACT YEAR, MONTH AND DAY
   14            YR = YRMODAY / 10000
   15            MON = (YRMODAY - YR * 10000) / 100
   16            DAY = MOD (YRMODAY, 100)
   17       C  EXTRACT HOURS, MINUTES AND SECONDS
   18            HR = HRMINSEC / 10000
   19            MIN = (HRMINSEC - HR * 10000) / 100
   20            SEC = MOD (HRMINSEC, 100)
   21       C  COMPUTE TIME IN TENTHS OF MINUTES SINCE FIRST OF YEAR
   22            ITIME = MIN * 10 + (SEC * 100 / 60 + 5)/ 10
   23            ITIME = ((MTABLE(MON) + DAY-1) * 24 + HR) * 600 + ITIME
   24       C  FOR THE SECOND YEAR
   25            IF (YR .GT. ISYEAR) ITIME = ITIME + (MTABLE(12) + 31) * 14400
   26       C  CHECK FOR LATER THAN FEB 28, 23:59:50 OF SECOND YEAR
   27            IF (ITIME .LT. 6119998) RETURN
   28            WRITE (61, 100)
   29          100 FORMAT (≠-TIME LATER THAN FEB 28, 23:59:50 OF SECOND YR≠)
   30            CALL SBJP
   31            END
LENGTH OF SUBPROGRAM      00164
LENGTH OF COMMON MTABLE   00015
```

```
      OS3 FORTRAN  VERSION 3.13          11/05/75  2240

    1            SUBROUTINE IDTG (YR, TIME, YRMODAY, T)
    2      C  ENTER WITH YEAR, E.G. 72, AND TIME IN TENTHS OF MINUTES
    3      C  SINCE BEGINNING OF THE YEAR (OUTPUT FROM NGDCTIME OR ITIME).
    4      C  EXIT WITH YR MO DAY, HR MIN10.
    5            COMMON/MTABLE/MTABLE(12),ISYEAR
    6            INTEGER T, YR, TIME, YRMODAY
    7            T=(MTABLE(12)+31)*14400
    8      C  T IS TENTHS OF MINUTES IN THE YEAR
    9            IF(TIME.LT.T)GOTO2
   10      C  FOR THE SECOND YEAR
   11            YR=ISYEAR+1
   12            TIME=TIME-T
   13          2 T=TIME/14400
   14      C  T IS INTEGER DAYS
   15            DO 4 I=1,12
   16            IF(T.LT.MTABLE(I))GOTO3
   17          4 CONTINUE
   18            I=13
   19      C  I IS THE MONTH
   20          3 I=I-1
   21            IDAY=T-MTABLE(I)+1
   22            YRMODAY=YR*10000+I*100+IDAY
   23      C  NEW T IS REMAINING TENTHS OF MINUTES OF DAY
   24            T=TIME-T*14400
   25      C  COMPUTE HOURS AND MINUTES * 10
   26            IHR=T/600
   27            MNT=T-IHR*600
   28            T=IHR*1000+MNT
   29            RETURN
   30            END
LENGTH OF SUBPROGRAM      00153
LENGTH OF COMMON MTABLE   00015
```

```
      OS3 FORTRAN  VERSION 3.13          11/05/75  2240

    1            SUBROUTINE ADTG(YR,TIME,YRMODAY,HRMNT)
    2       C  ENTER WITH YEAR, E.G. 72, AND TIME IN TENTHS OF MINUTES
    3       C  SINCE BEGINNING OF THE YEAR (OUTPUT FROM NGDCTIME OR ITIME).
    4       C  EXIT WITH YR MO DAY, HR MIN AND TENTHS.  OUTPUT IS IN ≠A≠
    5       C  FORMAT WITH LEADING ZEROS.
    6            COMMON /MTABLE/ MTABLE (12), ISYEAR
    7            INTEGER T, YR, TIME
    8            T=(MTABLE(12)+31)*14400
    9       C  T IS TENTHS OF MINUTES IN THE YEAR
   10            IF(TIME.LT.T)GOTO2
   11       C  FOR THE SECOND YEAR
   12            YR=ISYEAR+1
   13            TIME=TIME-T
   14          2 T=TIME/14400
   15       C  T IS INTEGER DAYS
   16            DO 4 I=1,12
   17            IF(T.LT.MTABLE(I))GOTO3
   18          4 CONTINUE
   19            I=13
   20       C  I IS THE MONTH, J IS YRMODA
   21          3 I=I-1
   22            IDAY=T-MTABLE(I)+1
   23            J=YR*10000+I*100+IDAY
   24       C  NEW T IS REMAINING TENTHS OF MINUTES OF DAY
   25            T=TIME-T*14400
   26       C  COMPUTE HOURS AND MINUTES * 10
   27            IHR=T/600
   28            MNT=T-IHR*600
   29            T=IHR*1000+MNT
   30       C  CHANGE T TO HOURS MINUTES AND TENTHS OF MINUTES WITH
   31       C  LEADING ZEROS - ≠A5≠ FORMAT
   32            ENCODE(8,100,HRMNT)T
   33        100 FORMAT (I5,3X)
   34       C  ZB ZEROES THE BLANKS IN THE REAL ARGUMENT
   35            HRMNT = ZB (HRMNT)
   36       C  CHANGE J TO YEAR MONTH DAY IN ≠A6≠ FORMAT
   37            ENCODE(8,101,YRMODAY)J
   38        101 FORMAT(I6,2X)
   39            RETURN
   40            END
LENGTH OF SUBPROGRAM      00176
LENGTH OF COMMON MTABLE   00015
```

```
LENGTH OF  PRG         00026
                                    1                IDENT     ARMYTIME
                00000 P             2                ENTRY     ARMYTIME
                                    3   ********************************************************************************
                                    4   ***                                                                          *
                                    5   ***  THIS SUBROUTINE RETURNS THE CURRENT 24 HOUR TIME IN ≠A8≠                 *
                                    6   ***  FORMAT TO THE CALLING PROGRAM.                                           *
                                    7   ***                                                                          *
                                    8   ***  THE CALLING SEQUENCE IS:                                                *
                                    9   ***                                                                          *
                                   10   ***       CALL ARMYTIME (X)                                                  *
                                   11   ***                                                                          *
                                   12   ***  THE FORMAT OF THE RETURNED VALUE IS:                                    *
                                   13   ***  ≠  HHMM  ≠ WHERE HH IS THE HOURS AND MM IS THE MINUTES                  *
                                   14   ***                                                                          *
                                   15   ********************************************************************************
                00001              16   X1        EQU       1
                00000 P            17   ARMYTIME  EQU       *                    ENTRY POINT
00000  01077777                    18             UJP       **
00001  54100000 P                   19             LDI       ARMYTIME,X1          FOR THE RETURN
00002  53010037                    20             TMQ       37B                  GET THE HOURS FROM RF 37
00003  17700037                    21             ANQ       37B                  SAVE THE BOTTOM 5 BITS
00004  14400000                    22             ENA,S     0                    CLEAR A
00005  51000024 P                  23             DVA       TEN                  CONVERT TO DECIMAL
00006  42000112 P    00022 2       24             SACH      ATIME+2
00007  43000113 P    00022 3       25             SQCH      ATIME+3
00010  53010022                    26             TMQ       22B                  GET MILLISECS FROM RF 22
00011  14400000                    27             ENA,S     0
00012  51000025 P                  28             DVA       P60000               DIVIDE BY 60000
00013  13077747                    29             SHAQ      -24                  GOT THE MINUTES
00014  51000024 P                  30             DVA       TEN                  CONVERT TO DECIMAL
00015  42000114 P    00023 0       31             SACH      ATIME+4
00016  43000115 P    00023 1       32             SQCH      ATIME+5
00017  25000022 P                  33             LOAQ      *+3                  GET THE WHOLE TIME
00020  45500000                    34             STAQ,I    0,X1                 SEND IT BACK
00021  01100001                    35             UJP       1,X1                 RETURN
                00110 P    00022 0 36   ATIME     EQU,C     *
00022  60600000                    37             OCT       60600000,00006060
00024  00000012                    38   TEN       OCT       00000012
00025  00165140                    39   P60000    OCT       165140
                                   40             END

 NO LINES WITH ERRORS
```

```
ASSEMBLER/OS3  V1.0 11/10/75 1558  PAGE   1     ZB

LENGTH OF  PRG        00023

             00007 P
             00000 P


             00001
    00000  01077777
    00001  54100000 P
    00002  25500000
    00003  45000020 P
    00004  37500022 P
    00005  36000020 P
    00006  01100001
    00007  01077777
    00010  54100007 P
    00011  25500000
    00012  45000020 P
    00013  13000030
    00014  37500022 P
    00015  36000021 P
    00016  13000030
    00017  01000004 P
    00020
    00021
    00022  60606060


 NO LINES WITH ERRORS
```

```
 1            IDENT     ZB
 2            ENTRY     ZB                    ENTRY FOR REAL ARGUMENTS
 3            ENTRY     IZB                   ENTRY FOR INTEGER ARGUMENTS
 ********************************************************************************
 5  ***                                                                         *
 6  ***  THIS FUNCTION ZEROS OUT THE BLANKS (60B#S) OF THE ARGUMENT             *
 7  ***  IZB FOR AN INTEGER ARGUMENT AND ZB FOR A REAL (DOUBLE WORD)            *
 8  ***  ARGUMENT.                                                              *
 9  ***                                                                         *
10  ***  CALLING SEQUENCE:                                                      *
11  ***                                                                         *
12  ***       R = ZB (A)                                                        *
13  ***       J = IZB (I)                                                       *
14  ***                                                                         *
 ********************************************************************************
16  X1        EQU       1
17  IZB       UJP       **                    ENTRY FOR INTEGER ARGUMENTS
18            LDI       IZB,X1                FOR RETURN FROM IZB
19            LDA,I     0,X1                  GET THE ARGUMENT
20            STA       TEMP
21  FINISH    LPA       BLANKS                AND (ARG, BLANKS)
22            SCA       TEMP                  EOR (AND (ARG, BLANKS), ARG)
23            UJP       1,X1                  RETURN
24  ZB        UJP       **                    ENTRY FOR REAL ARGUMENTS
25            LDI       ZB,X1                 FOR RETURN FROM ZB
26            LDAQ,I    0,X1                  GET THE ARGUMENT
27            STAQ      TEMP
28            SHAQ      24
29            LPA       BLANKS                AND (ARG, BLANKS)
30            SCA       TEMP2                 EOR (AND(ARG, BLANKS), ARG)
31            SHAQ      24
32            UJP       FINISH
33  TEMP      BSS       1
34  TEMP2     BSS       1
35  BLANKS    OCT       60606060
36            END
```

```
ASSEMBLER/OS3  V1.0 11/06/75 1414   PAGE    1     AINT
LENGTH OF  PRG          00066
                    00000 P                 IDENT     AINT
                                            ENTRY     AINT
                                  ***********************************************************************
                                  ***                                                                   *
                                  ***   THIS FUNCTION TRUNCATES THE FRACTION PORTION OF A FP           *
                                  ***   ARGUMENT BY SHIFTING THE FRACTION PORTION OFF THE END          *
                                  ***   RETURNING THE INTEGER PORTION.                                  *
                                  ***                                                                   *
                                  ***   THE CALLING SEQUENCE IS:                                        *
                                  ***                                                                   *
                                  ***       Y = AINT (X)                                                *
                                  ***                                                                   *
                                  ***********************************************************************
                    00001           X1        EQU       1                       INDEX 1
                    00002           X2        EQU       2                       INDEX 2
      00000  01077777               AINT      UJP       **                      ENTRY POINT
      00001  54100000 P                       LDI       AINT,X1                 FOR RETURN
      00002  25500000                         LDAQ,I    0,X1                    GET ARGUMENT
      00003  03300030 P                       AZJ,LT    NEGARG                  CHECK FOR NEGATIVE
      00004  00700006 P                       RTJ       SUB                     FOR POS NUM
      00005  01100001                         UJP       1,X1                    RETURN
      00006  01077777               SUB       UJP       **                      FOR POS NUMS
      00007  45000044 P                       STAQ      TEMP
      00010  61000052 P                       FSB       ONE                     CHECK FOR .LT. 1
      00011  03300036 P                       AZJ,LT    LTONE                   GO GET ZERO
      00012  25000044 P                       LDAQ      TEMP
      00013  61000050 P                       FSB       MAX                     CHECK FOR .GT. 2**43
      00014  03200040 P                       AZJ,GE    ERROR                   NUM TOO LARGE
      00015  25000044 P                       LDAQ      TEMP
      00016  12077763                         SHA       -12
      00017  37000047 P                       LPA       MASK                    SAVE 2 OCT DIGITS
      00020  53600000                         TAI       X2
      00021  16477777                         XOA,S     77777B
      00022  40000046 P                       STA       ITEMP
      00023  20000044 P                       LDA       TEMP
      00024  13277733                         SHAQ      -44B,X2                 SHIFT TO DECIMAL PT
      00025  54200046 P                       LDI       ITEMP,X2
      00026  13200044                         SHAQ      44B,X2                  SHIFT BACK
      00027  01400006 P                       UJP,I     SUB                     RETURN
      00030  16477777               NEGARG    XOA,S     77777B
      00031  16577777                         XOQ,S     77777B
      00032  00700006 P                       RTJ       SUB                     TREAT AS POSITIVE
      00033  16477777                         XOA,S     77777B
      00034  16577777                         XOQ,S     77777B                  COMPLEMENT
      00035  01100001                         UJP       1,X1                    RETURN
      00036  25000054 P             LTONE     LDAQ      ZERO
      00037  01100001                         UJP       1,X1                    RETURN
      00040  14600056 P             ERROR     ENA       EMESS
      00041  14700010                         ENQ       8
      00042  76000075                         WRITE     61
      00043  77620000                         SBJP      ABORT
      00044                         TEMP      BSS       2
      00046                         ITEMP     BSS       1
      00047  00000077               MASK      OCT       00000077
      00050  20434000               MAX       OCT       20434000,00000000
      00052  20014000               ONE       OCT       20014000,00000000
      00054  00000000               ZERO      OCT       00000000,00000000
      00056  40255151               EMESS     BCD       8,-ERROR IN AINT X > 2**43 ABORT
                                              END

   NO LINES WITH ERRORS
```

```
ASSEMBLER/OS3  V1.0 11/06/75 1414  PAGE   1     ATAN2
LENGTH OF  PRG        00066
                                   ATAN2    COSY/     01                          10/28/75  1633
                                            IDENT     ATAN2
          00000 P                           ENTRY     ATAN2
                                            EXT       ATAN
                                   ***********************************************************************
                                   ***  THIS COMPUTES THE VALUE OF ATAN (Y/X) RETURNING A VALUE BETWEEN   *
                                   ***  -PI AND +PI.  IT EXTENDS THE RANGE OF THE REGULAR ATAN FUNCTION.  *
                                   ***                                                                    *
                                   ***  REFERENCE:  IBM SYSTEM 360 FORTRAN IV LIBRARY SUBPROGRAMS         *
                                   ***                                                                    *
                                   ***  THE CALLING SEQUENCE IS:                                          *
                                   ***                                                                    *
                                   ***      A = ATAN2 (Y, X)                                              *
                                   ***                                                                    *
                                   ***********************************************************************
             00001                 X1       EQU       1
             00000 P               ATAN2    EQU       *                     ENTRY TO FUNCTION
00000  01077777                             UJP       **
00001  54100000 P                           LDI       ATAN2,X1
00002  25500001                             LDAQ,I    1,X1                  PICK UP X
00003  03100014 P                           AZJ,NE    CASE2                 IF (X) CASE2
00004  25500000                             LDAQ,I    0,X1                  GET Y
00005  03000042 P                           AZJ,EQ    YZERO                 UNDEFINED IF X = Y = 0.0
00006  25500000                    CASE1    LDAQ,I    0,X1                  ATAN2 = SIGN (Y) * PI / 2
00007  03200012 P                           AZJ,GE    PSIGN                 CHECK SIGN OF Y
00010  26000062 P                           LCAQ      PID2
00011  01100002                             UJP       2,X1                  DONE WITH CASE1 FOR SIGN (Y) NEG
00012  25000062 P                  PSIGN    LDAQ      PID2
00013  01100002                             UJP       2,X1                  DONE WITH CASE1 FOR SIGN (Y) POS
00014  25500000                    CASE2    LDAQ,I    0,X1                  GET Y
00015  63500001                             FDV,I     1,X1                  DIVIDE BY X
00016  45000056 P                           STAQ      YDX                   SAVE (Y/X) FOR LATER
00017  03200021 P                           AZJ,GE    *+2                   NEED THE ABS (YDX)
00020  26000056 P                           LCAQ      YDX                   FOR YDX NEG
00021  61000064 P                           FSB       TWOP24
00022  03200006 P                           AZJ,GE    CASE1                 IF (ABS(Y/X) .GE. 2**24) CASE1
00023  25500001                             LDAQ,I    1,X1                  GET X
00024  03300030 P                           AZJ,LT    CASE4                 CASE 4 IF (X) .LT. 0.0
00025  00777777 X                  CASE3    RTJ       ATAN                  ATAN2 = ATAN (Y/X)
00026  77000056 P                           77        YDX
00027  01100002                             UJP       2,X1                  DONE WITH CASE3
00030  00700025 X                  CASE4    RTJ       ATAN                  ATAN2 = ATAN (Y/X) + SIGN (Y) + P
00031  77000056 P                           77        YDX
00032  45000056 P                           STAQ      YDX                   YDX IS A TEMP
00033  25500000                             LDAQ,I    0,X1                  GET Y
00034  03200037 P                           AZJ,GE    PSIGN2
00035  26000060 P                           LCAQ      PI                    - PI
00036  01000040 P                           UJP       EXIT
00037  25000060 P                  PSIGN2   LDAQ      PI                    + PI
00040  60000056 P                  EXIT     FAD       YDX                   Y/X +- PI
00041  01100002                             UJP       2,X1                  RETURN
00042  14600050 P                  YZERO    ENA       EMESS
00043  14700006                             ENQ       6
00044  76000075                             WRITE     61                    ERROR MESSAGE
00045  14400000                             ENA,S     0
00046  14500000                             ENQ,S     0                     CLEAR AQ
00047  01100002                             UJP       2,X1                  RETURN WITH 0.0 FOR X=Y=0.0
00050  40255151                    EMESS    BCD       6,-ERROR IN ATAN2  Y=X=0.0
00055                              YDX      BSS       2
00060   20026220                   PI       OCT       20026220,77324774     =3.14159265
00062   20016220                   PID2     OCT       20016220,77324744     =3.14159265 / 2.
00064   20314000                   TWOP24   OCT       20314000,00000000     = 2. ** 24
                                            END

NO LINES WITH ERRORS
```

```
ASSEMBLER/OS3  V1.0 11/06/75 1417  PAGE   1     AMOD

LENGTH OF  PRG          00035
                                        IDENT     AMOD
           00000 P                      ENTRY     AMOD
                                        EXT       AINT
                                ***********************************************************************
                                ***                                                                  *
                                ***   AMOD (X,Y) = X - AINT(X/Y) * Y                                 *
                                ***                                                                  *
                                ***       CALLING SEQUENCE                                           *
                                ***                                                                  *
                                ***       Z = AMOD (X, Y)                                            *
                                ***                                                                  *
                                ***********************************************************************
             00001               X1     EQU       1
    00000  01077777              AMOD   UJP       **                  ENTRY POINT
    00001  54100000 P                   LDI       AMOD,X1             ADDRESS OF X INTO X1
    00002  25500001                     LDAQ,I    1,X1                GET Y
    00003  03000023 P                   AZJ,EQ    ERROR               CHECK FOR ZERO
    00004  45000021 P                   STAQ      TEMP                SAVE IT
    00005  25500000                     LDAQ,I    0,X1                X INTO (AQ)
    00006  03000017 P                   AZJ,EQ    ZEROX               FIX FOR WHEN X = 0.
    00007  63000021 P                   FDV       TEMP                DEVIDE BY Y
    00010  45000021 P                   STAQ      TEMP
    00011  00777777 X                   RTJ       AINT                TO TRUNCATE
    00012  77000021 P                   77        TEMP
    00013  54100000 P                   LDI       AMOD,X1             AINT EATS X1
    00014  62500001                     FMU,I     1,X1                TRUNCATED X/Y TIMES Y
    00015  16477777                     XOA,S     -0                  COMPLEMENT (A)
    00016  16577777                     XOQ,S     -0                  COMPLEMENT (Q)
    00017  63500000                ZEROX FAD,I     0,X1                SUBTRACK X/Y*Y FROM X
    00020  01100002                EXIT  UJP       2,X1                RETURN
    00021                          TEMP  BSS       2
    00023  14600027 P              ERROR ENA       EMESS
    00024  14700006                      ENQ       6
    00025  75000075                      WRITE     61
    00026  77620000                      SBJP                          ABORT
    00027  40255151                EMESS BCD       6,-ERROR IN AMOD Y=0 ABORT
                                         END

   NO LINES WITH ERRORS
```

```
ASSEMBLER/OS3  V1.0 11/06/75 1415  PAGE    1     FWSP

LENGTH OF  PRG         00007
                                          1                IDENT     FWSP
                00000 P                   2                ENTRY     FWSP
                                          3   ************************************************************************
                                          4   ***  THIS SUBROUTINE EXECUTES A FORWARD SPACE ON THE LUN              *
                                          5   ***                                                                  *
                                          6   ***  CALLING SEQUENCE:                                               *
                                          7   ***                                                                  *
                                          8   ***       RTJ       FWSP                                            *
                                          9   ***       CALL FWSP (LUN)                                           *
                                         10   ***                                                                  *
                                         11   ************************************************************************
                00001                    12    X1        EQU       1
    00000  01077777                      13    FWSP      UJP       **                  ENTRY POINT
    00001  54100000 P                    14              LDI       FWSP,X1             GET THE ADDRESS
    00002  20500000                      15              LDA,I     0,X1                GET THE LUN
    00003  44000005 P                    16              SWA       LUN
    00004  14700010                      17              ENQ       10B
    00005  72077777                      18    LUN       CNTL      **                  DO THE FWSP
    00006  01100001                      19              UJP       1,X1                RETURN
                                         20              END

     NO LINES WITH ERRORS
```

```
ASSEMBLER/OS3  V1.0 11/06/75 1418  PAGE    1     ISTATUS

LENGTH OF  PRG          00007
            00000 P
                                  IDENT     ISTATUS
                                  ENTRY     ISTATUS
                        ****************************************************************
                        ***                                                            *
                        ***  THIS FUNCTION RETURNS THE STATUS BITS OF THE LUN IN A     *
                        ***  BOTH THE LUN STATUS AND THE HARDWARE TYPE ARE AVAILABLE TO *
                        ***  BE MASKED OFF.                                            *
                        ***                                                            *
                        ***  CALLING SEQUENCE:                                         *
                        ***                                                            *
                        ***       I = ISTATUS (LUN)                                    *
                        ***                                                            *
                        ****************************************************************
             00001      X1        EQU       1
 00000  01077777        ISTATUS   UJP       **                  ENTRY POINT
 00001  54100000 P                LDI       ISTATUS,X1          FOR THE RETURN
 00002  20500000                  LDA,I     0,X1                GET THE LUN
 00003  44000005 P                SWA       LUN
 00004  14700000                  ENQ       0
 00005  72077777        LUN       CNTL      **                  GET THE STATUS
 00006  01100001                  UJP       1,X1                RETURN
                        ****************************************************************
                        ***                                                            *
                        ***  THE BIT ASSIGNMENTS RETURNED ARE:                         *
                        ***                                                            *
                        *                   40000000            FILE PROTECT           *
                        *                   20000000            LOAD POINT             *
                        *                   10000000            END OF DATA            *
                        *                   04000000            END OF FILE            *
                        *                   02000000            DESTRUCTIVE READ       *
                        *                   01000000            BIN RECORD READ        *
                        *                   00400000            ABNORMAL / UNAVAILABLE *
                        *                   00200000            SEEK ERROR             *
                        *                   00100000            SAVED FILE             *
                        ***                                                            *
                        ***  THE BOTTOM 4 BITS CONTAIN HARDWARE TYPE AS FOLLOW:        *
                        ***                                                            *
                        ***                 0000                UNIT NOT EQUIPPED      *
                        ***                 0001                FILE                   *
                        ***                 0010                LP                     *
                        ***                 0011                CP                     *
                        ***                 0100                CR                     *
                        ***                 0101                MT                     *
                        ***                 0110                TTY                    *
                        ***                 0111                PLOTTER                *
                        ***                 1000                NULL                   *
                        ***                 1001                TV                     *
                        ***                 1010                RAF                    *
                        ***                 1011                TASK                   *
                        ****************************************************************
                                  END

     NO LINES WITH ERRORS
```

```
ASSEMBLER/OS3  V1.3 11/06/75 1416  PAGE   1    RELEASE

LENGTH OF  PRG        00007
                                    1              IDENT     RELEASE
                00000 P             2              ENTRY     RELEASE
                                    3  ************************************************************************
                                    4  ***                                                                  *
                                    5  ***  THIS SUBROUTINE EXECUTES AN OS-3 CONTROL MODE RELEASE           *
                                    6  ***  ON THE SPECIFIED LUN.  IE THE LENGTH OF THE LUN BECOMES         *
                                    7  ***  ZERO.                                                          *
                                    8  ***                                                                  *
                                    9  ***  THE CALLING SEQUENCE IS:                                       *
                                   10  ***                                                                  *
                                   11  ***       CALL RELEASE (LUN)                                         *
                                   12  ***                                                                  *
                                   13  ************************************************************************
                00001              14  X1        EQU       1
00000  01477777                    15  RELEASE   UJP,I     **
00001  20400000 P                   16            LDA,I     RELEASE           GET THE LUN
00002  44000004 P                   17            SWA       LUN
00003  14700003                    18            ENQ       3
00004  72077777                    19  LUN       CNTL      **                RELEASE IT
00005  54100000 P                   20            LDI       RELEASE,1         FOR THE RETURN
00006  01100001                    21            UJP       1,1               RETURN
                                   22            END

 NO LINES WITH ERRORS
```

LENGTH OF PRG 00021

```
                                    1              IDENT     RESET
                00000 P             2              ENTRY     RESET
                                    3  ************************************************************************
                                    4  ***                                                                   *
                                    5  ***  RESET UNEQUIPS ALL LUNS 0 THROUGH 49 AND RETURNS                 *
                                    6  ***  THERE ARE NO PARAMETERS.                                         *
                                    7  ***                                                                   *
                                    8  ***  CALLING SEQUENCE:                                               *
                                    9  ***                                                                   *
                                   10  ***        CALL RESET                                                 *
                                   11  ***                                                                   *
                                   12  ************************************************************************
                00001              13  X1        EQU       1
                00002              14  X2        EQU       2
00000  01077777                    15  RESET     UJP       **              ENTRY POINT
00001  14200000                    16            ENI       0,X2            LUN COUNTER
00002  14100002                    17  LOOP1     ENI       2,X1            FOR UNEQUIP
00003  71200000                    18            XREQ      0,X2            UNEQUIP (X2)
00004  04100000                    19            ISE       0,X1            SKIP IF OK
00005  01000007 P                  20            UJP       *+2             SKIP IF ERROR
00006  01000015 P                  21            UJP       OK              NO ERRORS
00007  04100001                    22            ISE       1,X1            CHECK FOR UNIT NOT EQUIPPED
00010  01000012 P                  23            UJP       *+2             FP BUT NOT SAVED
00011  01000015 P                  24            UJP       OK              UNIT NOT EQUIPPED
00012  14100004                    25            ENI       4,X1            FOR RFP
00013  71200000                    26            XREQ      0,X2            RFP
00014  01000002 P                  27            UJP       LOOP1           TRY AGAIN
00015  15200001                    28  OK        INI       1,X2            FOR NEXT LUN
00016  04200062                    29            ISE       50,X2           DONE WITH LOOP1
00017  01000002 P                  30            UJP       LOOP1           NO DO NEXT LUN
00020  01400000 P                  31            UJP,I     RESET           RETURN
                                   32            END
```

NO LINES WITH ERRORS

```
ASSEMBLER/OS3  V1.3 11/06/75 1416  PAGE   1    MI

LENGTH OF  PRG        00004
                                    1              IDENT     MI
                                    2              ENTRY     MI                    ENTRY POINT
                                    3  ********************************************************************
                                    4  ***                                                                *
                                    5  ***  THIS SUBROUTINE ALLOWS THE USER TO RETURN TO THE LAST LOCATION  *
                                    6  ***  IN A PROGRAM FROM WHICH MI WAS CALLED.                         *
                                    7  ***  CALLING SEQUENCE:                                             *
                                    8  ***                                                                *
                                    9  ***       CALL MI                                                  *
                                   10  ***                                                                *
                                   11  ***                                                                *
                                   12  ***  RETURN TO THE CALLING LOCATION IS BY TYPING CNTL A FOLLOWED BY *
                                   13  ***  #MI# AND A CARRIAGE RETURN.                                   *
                                   14  ***                                                                *
                                   15  ********************************************************************
00000  01077777                    16  MI        UJP       **                    ENTRY
00001  20000000 P                  17            LDA       MI                    GET CALLING ADDRESS
00002  40000013                    18            STA       13B                   IN 13B FOR A MANNUAL INTERRUPT
00003  01400000 P                  19            UJP,I     MI                    RETURN
                                   20            END

 NO LINES WITH ERRORS
```

```
ASSEMBLER/OS3  V1.0 11/06/75 1415  PAGE   1    ZAP

LENGTH OF  PRG        00007
                                    1                IDENT     ZAP
             00000 P                2                ENTRY     ZAP
                                    3   ************************************************************************
                                    4   ***                                                                  *
                                    5   ***  THIS SUBROUTINE ZEROES A PAGE (2048 WORDS) OF CORE               *
                                    6   ***                                                                  *
                                    7   ***                                                                  *
                                    8   ***  CALLING SEQUENCE:                                               *
                                    9   ***                                                                  *
                                   10   ***       CALL ZAP (I)                                               *
                                   11   ***  WHERE I IS THE PAGE NUMBER (0 - 15 FOR LOWER CORE).              *
                                   12   ***                                                                  *
                                   13   ************************************************************************
             00001                 14    X1        EQU       1
  00000   01477777                 15    ZAP       UJP,I     **
  00001   20400000 P               16              LDA,I     ZAP                 GET THE PAGE NUMBER
  00002   44000004 P               17              SWA       PAGE
  00003   14100006                 18              ENI       6,X1                FOR THE ZAP
  00004   71077777                 19    PAGE      XPEQ      **                  ZAP IT
  00005   54100000 P               20              LDI       ZAP,X1              FOR THE RETURN
  00006   01100001                 21              UJP       1,X1                RETURN
                                   22              END

   NO LINES WITH ERRORS
```

```
ASSEMBLER/OS3  V1.0 11/06/75 1417  PAGE   1    SBJP

LENGTH OF  PRG        00002
                                    1               IDENT     SBJP
                00000 P             2               ENTRY     SBJP
                                    3   ************************************************************************
                                    4   ***                                                                  *
                                    5   ***  THIS SUBROUTINE PUTS THE USER IN OS-3 CONTROL MODE.              *
                                    6   ***                                                                  *
                                    7   ***  THE CALLING SEQUENCE IS:                                        *
                                    8   ***                                                                  *
                                    9   ***       CALL SBJP                                                  *
                                   10   ***                                                                  *
                                   11   ***  THERE IS NO RETURN TO THE CALLING PROGRAM                        *
                                   12   ***                                                                  *
                                   13   ************************************************************************
                00000 P            14   SBJP      EQU       *                    ENTRY POINT
    00000  01077777                15               UJP       **
    00001  77620000                16               SBJP      CALL OS-3 CONTROL MODE
                                   17               END

     NO LINES WITH ERRORS
```

```
        OS3 FORTRAN  VERSION 3.13          11/05/75  2240

 1              FUNCTION SPEED (ISREV,IPREV,NDATE,NTIME)
 2        C  FINDS SPEED FROM EMLOG AND ENGINE REVS FOR ONE AND TWO
 3        C  SHAFT OPERATION DURING YALOC 71 LEG 6
 4              COMMON /DATA/ SP1(40),SP2(40)
 5        C
 6        C  ONE SHAFT OPERATION TABLE
 7        C
 8              DATA((SP1(I),I=1,40)=3.0, 3.2, 3.4, 3.5, 3.7, 3.8, 4.0,
 9             1 4.1, 4.3, 4.4, 4.6, 4.7, 4.9, 5.0, 5.2, 5.3, 5.4, 5.6,
10             2 5.7, 5.9, 6.0, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8,
11             3 6.9, 7.0, 7.1, 7.2, 7.2, 7.3, 7.4, 7.4, 7.5, 7.6, 7.7)
12        C
13        C  TWO SHAFT OPERATION TABLE
14        C
15              DATA((SP2(I),I=1,40)=4.5, 4.8, 5.0, 5.2, 5.4, 5.6, 5.8,
16             1 6.0, 6.2, 6.3, 6.5, 6.7, 6.9, 7.1, 7.2, 7.4, 7.5, 7.7,
17             2 7.8, 7.9, 8.1, 8.2, 8.3, 8.4, 8.6, 8.7, 8.8, 8.9, 9.0,
18             3 9.2, 9.3, 9.4, 9.6, 9.7, 9.8, 9.9, 10.0, 10.1, 10.2, 10.3)
19              IF(IABS(ISREV-IPREV).LE.50)GOTO200
20        C
21        C  CHECK FOR ONE SHAFT OPERATION
22        C
23              IF(ISREV.LT.100.OR.IPREV.LT.100)GOTO210
24        C
25        C  FOR TWO SHAFT OPERATION, BUT DELTA S > 50 RPM
26        C
27              WRITE(61,290)NDATE,NTIME,ISREV,IPREV
28          290 FORMAT(≠ ≠,4I7≠ 2 SHAFT DELTA REVS .GT. 50 RPM≠)
29        C
30        C  CALC SPEED FOR TWO SHAFT OPERATION
31        C
32          200 JREV=(ISREV+IPREV)/20 - 27
33              IF(JREV.LE.0)GOTO220
34              IF(JREV.GT.40)JREV=40
35              SPEED =SP2(JREV)
36              RETURN
37        C
38        C  CALC SPEED FOR ONE SHAFT OPERATION
39        C
40          210 JREV=(ISREV+IPREV)/10 - 27
41              IF(JREV.LE.0)GOTO220
42              IF(JREV.GT.40)JREV=40
43              SPEED =SP1(JREV)
44              RETURN
45        C
46        C  FOR RPM < 270 USE SPEED = 0.0
47        C
48          220 SPEED =0
49              RETURN
50              END
LENGTH OF SUBPROGRAM      00175
LENGTH OF COMMON DATA     00240
```

```
      OS3 FORTRAN  VERSION 3.13          11/05/75  2240

    1            SUBROUTINE GETMATAB (MZ, ITAB, IEND)
    2       C  THIS SUBROUTINE GETS THE MATHEWS ZONE TABLE FOR ZONE ≠MZ≠ AND
    3       C  PUTS IT IN THE INTEGER ARRAY ≠ITAB≠.  IF THE DATA FOR THE ZONE
    4       C  ≠MZ≠ IS NOT CONTAINED IN THE RAF ≠*MATAB≠ THEN AN ERROR IS WRITTEN
    5       C  AND THE SUBROUTINE ABORTS THE USER.  LUN 51 IS USED TO INPUT DATA
    6       C  FROM THE RAF ≠*MATAB≠.  PRESENTLY ZONES 23-25 AND 41-45 ARE INCLUDED.
    7       C  ≠IEND≠ CONTAINS THE LENGTH OF THE TABLE FOR ZONE ≠MZ≠.
    8            DIMENSION ITAB (60), INDEX (2,52)
    9            IF(MZ .LT. 1 .OR. MZ .GT. 52) GO TO 15
   10       C  HAS THE INDEX ALREADY BEEN READ IN
   11            IF (IEND) GO TO 10
   12       C  NO. READ IN THE INDEX
   13            CALL UNEQUIP (51)
   14            CALL EQUIP (51, 8H*MATAB  )
   15            BUFFER IN (51, 1) (INDEX, INDEX(2, 52))
   16       C  IS THE MATHEWS ZONE ≠MZ≠ IN THE INDEX
   17         10 IF (INDEX (1, MZ)) GO TO 20
   18       C  NO. WRITE ERROR AND ABORT
   19         15 WRITE (61,100) MZ
   20        100 FORMAT (≠-NO DATA FOR MATHEWS ZONE≠,I3,≠ SEE KEN KEELING≠)
   21            CALL SBJP
   22       C  POSITION THE RAF TO THE RIGHT PLACE
   23         20 CALL SEEK (51, INDEX (2, MZ))
   24       C  GET THE LENGTH OF THE TABLE
   25            IEND = INDEX (1, MZ)
   26            BUFFER IN (51, 1) (ITAB, ITAB(IEND))
   27            RETURN
   28            END
LENGTH OF SUBPROGRAM      00327
```

APPENDIX B

```
      OS3 FORTRAN  VERSION 3.13          11/11/75  1148

 1            PROGRAM ZORK
 2            DIMENSION ITAB (60)
 3            INTEGER  CMETERS
 4         10 READ (60, 100) MZ, IFMS
 5        100 FORMAT (I2,I4)
 6      C  START ITAB AT (2) SUCH THAT ITAB (1) = 0
 7            ITAB (1) = 0
 8            CALL GETMATAB (MZ, ITAB (2), IEND)
 9      C  USE 1.87452 IF IFMS ASSUMED 800 FATHOMS PER SEC.
10      C  USE 1.98288 IF IFMS ASSUMED 820 FATHOMS PER SEC.
11            METERS = IFMS * 1.87452 + 0.5
12      C I1 IS INDEX FOR ITAB
13            I1 = METERS / 200 + 1
14            IF (I1 .LE. IEND) GO TO 20
15            WRITE (61, 101) MZ, I1
16        101 FORMAT (≠0MATHEWS ZONE≠,I3,≠ ELEMENT ≠,I2,≠ TOO DEEP FOR TABLE≠)
17            CALL SBJP
18      C  INTERPOLATE THE CORRECTED METERS USING THE MATHEWS TABLE
19         20 CMETERS = METERS + (METERS - (I1-1)*200)/ 200. * (ITAB (I1+1)
20           1           - ITAB (I1)) + ITAB (I1)
21            WRITE (61, 102) CMETERS
22        102 FORMAT (≠ CORRECTED METERS = ≠,I5)
23            GO TO 10
24            END

 NO ERRORS FOR ZORK
LENGTH OF SUBPROGRAM      00250
```

MATHEWS ZONE

DEPTH	23	24	25	41	42	43	44	45
200	-2	-4	-5	4	1	-1	0	-5
400	-6	-7	-9	4	0	-3	0	-8
600	-9	-11	-14	2	-2	-6	0	-13
800	-11	-14	-18	-1	-4	-7	0	-17
1000	-13	-18	-21	-3	-5	-9	-13	-20
1200	-16	-21	-25	-5	-7	-12	-16	-24
1400	-19	-23	-27	-7	-9	-14	-19	-27
1600	-20	-26	-30	-9	-11	-16	-21	-30
1800	-23	-28	-31	-11	-12	-18	-23	-31
2000	-24	-28	-34	-12	-14	-20	-24	-34
2200	-25	-29	-34	-12	-15	-21	-26	-34
2400	-26	-30	-35	-13	-16	-21	-26	-35
2600	-26	-31	-35	-14	-16	-21	-26	-36
2800	-26	-30	-35	-13	-15	-21	-26	-35
3000	-26	-30	-37	-12	-14	-20	-26	-34
3200	-23	-29	-35	-11	-13	-17	-23	-35
3400	-23	-28	-32	-9	-11	-16	-23	-32
3600	-20	-24	-32	-7	-10	-15	-22	-29
3800	-18	-23	-31	-5	-8	-13	-18	-29
4000	-17	-20	-27	-3	-5	-10	-6	-25
4200	-12	-17	-26	3	0	-6	-11	-20
4400	-8	-14	-22	6	3	-3	-9	-18
4600	-4	-9	-18	9	9	1	-3	-13
4800	0		-14	13	13	5	0	-7
5000	4		-10	20	17	9	7	-3
5200				25	25	16	14	4
5400				33	30	21	19	11
5600				39	38	29	27	19
5800				48	44	36	34	24
6000				56	52	45	43	33
6200				63	60	52	50	42
6400				71	71	62	60	51
6600				83	81	71	69	61
6800				93	90	81	79	70
7000				103	103	91	89	81
7200				113	116	102	100	
7400				124	127	114	112	
7600				138	139	126		
7800				149	152	139		
8000				163	165	152		
8200				179	178			
8400				190	190			
8600				206				
8800				219				
9000				235				
9200				251				
9400				267				
9600				285				
9800				302				
10000				319				