MEASURE:  
A Program for Morphometric Measurements

by

David Lazarus

Woods Hole Oceanographic Institution  
Woods Hole, Massachusetts 02543

October, 1986

Technical Report

Funding was provided by the National Science Foundation under grant Nos. DEB81-18743 and DPP83-17087.

Reproduction in whole or in part is permitted for any purpose of the United States Government. This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept. WHOI-86-40.

Approved for publication; distribution unlimited.

Approved for Distribution:

[Signature]
David A. Ross, Chairman  
Department of Geology and Geophysics
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>3</td>
</tr>
<tr>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td>Program Overview</td>
<td>5</td>
</tr>
<tr>
<td>Measurement Routines</td>
<td>8</td>
</tr>
<tr>
<td>Protocol Files</td>
<td>10</td>
</tr>
<tr>
<td>Customization</td>
<td>13</td>
</tr>
<tr>
<td>References</td>
<td>15</td>
</tr>
<tr>
<td><strong>MEASURE</strong> Source Code Listing</td>
<td></td>
</tr>
<tr>
<td>INITMEAS</td>
<td>16</td>
</tr>
<tr>
<td>NEXUS</td>
<td>20</td>
</tr>
<tr>
<td>MENU</td>
<td>22</td>
</tr>
<tr>
<td>DELSPEC</td>
<td>24</td>
</tr>
<tr>
<td>EDITSPEC</td>
<td>25</td>
</tr>
<tr>
<td>NEXTSPEC</td>
<td>27</td>
</tr>
<tr>
<td>ENDMEAS</td>
<td>29</td>
</tr>
<tr>
<td>GETCMD</td>
<td>30</td>
</tr>
<tr>
<td>GETMSUB</td>
<td>31</td>
</tr>
<tr>
<td>LABEL</td>
<td>34</td>
</tr>
<tr>
<td>LABEL2</td>
<td>35</td>
</tr>
<tr>
<td>LENG1</td>
<td>36</td>
</tr>
<tr>
<td>LENG2</td>
<td>39</td>
</tr>
<tr>
<td>OUTL1</td>
<td>43</td>
</tr>
<tr>
<td>USER1</td>
<td>46</td>
</tr>
<tr>
<td>CRTPRINT</td>
<td>47</td>
</tr>
<tr>
<td>BEEP</td>
<td>48</td>
</tr>
<tr>
<td>CLEAN$</td>
<td>48</td>
</tr>
<tr>
<td>CLRSCRN</td>
<td>48</td>
</tr>
<tr>
<td>DISTANCE</td>
<td>49</td>
</tr>
<tr>
<td>INPUT</td>
<td>50</td>
</tr>
<tr>
<td>HIPAD</td>
<td>51</td>
</tr>
<tr>
<td>LASICO</td>
<td>53</td>
</tr>
<tr>
<td>3DHRDWRE (A test program for hardware interfaces)</td>
<td>55</td>
</tr>
</tbody>
</table>

## FIGURES

1 . MEASURE Program Outline-----------------------------------7
2 . Sample Protocol File-------------------------------------12
MEASURE: A Program for Morphometric Measurements

by

David Lazarus

ABSTRACT

'MEASURE', a FORTRAN77 program for collecting morphometric measurements, has been developed. Features of the program include the ability to accept distance and outline measurements in either 2 or 3 dimensions, to collect several different types of measurements on individual specimens, and to keep a running tally of the number of each type of specimen measured. The current version of the program is used for taxonomic and evolutionary studies of radiolarian microfossils and accepts data input from a digitizer tablet and a vertical measurement device attached to a microscope. An ordinary text file is used to configure the program for any desired set of distance or outline measurements. The program is highly modular, and is relatively easy to expand and modify. This report provides a detailed description of the program for both end-users and for programmers wishing to adapt MEASURE to their own uses. A full listing of the source code is included.
I: INTRODUCTION

This document explains the design, use, and customization of the morphometrics program MEASURE. MEASURE is a FORTRAN77 program which controls the acquisition of morphometrics data using a microcomputer and a variety of peripheral measurement devices. The version of the program described here has been used to collect measurements of radiolarian microfossils, using a NEC APC microcomputer, a transmitted light microscope, a camera-lucida, a digitizer tablet and a vertical coordinate digitizer. A description of this particular system is given in Lazarus (1986). MEASURE however has several general-purpose features which make it of potential interest to many geologists and biologists:

• The MEASURE program is written in a layered, modular fashion, designed specifically for easy modification to accommodate different types of measurement routines and different types of hardware. The program includes a set of general-purpose routines for handling measurement acquisition, including routines for assigning specimens to classes, labeling each specimen, selecting measurement routines from a menu, editing measured data, and saving data on the computer's disc-drives.

• The MEASURE program allows the user to measure a standard set of characteristics on a variety of different, though related, classes of objects. The program prompts the user to classify each specimen as part of the measurement procedure, and keeps a running tally of the different classes which have been measured. This feature of the program is designed for studies which employ morphometric measurements as a tool to help classify and discriminate between closely related groups of specimens. Examples of this type of study are very common in geology and biology, and include many problems in taxonomy, evolution, and petrography.

• MEASURE uses a protocol file which allows the user to change the characteristics being measured at any time without the need for modifying the program code. Several different protocol files can be created using an ordinary word processor, and used to configure the system for a variety of measurement tasks. For example, a taxonomist might create several protocol files, one for each of several different groups of organisms.

• Generalized routines are included for outline digitization and for measuring linear characteristics, with optional repeat measurements, variance computations, and extensive data-checking to prevent inaccurate measurements (whether due to operator or instrumental error) from being
incorporated into the data base. All measurement routines can handle data input in either 2 or 3 dimensions.

**About this document** - All users should read parts II and III below, which briefly describe how the different parts of the program operate. Non-programmers who wish to use the program should first make sure that it has been properly customized to run on their particular equipment. Assuming that this has been done, the system can be used simply by employing a word processor or text editor to create a 'protocol' file describing the measurements you wish to take. A protocol file is a standard ASCII format file which describes the names of the characteristics to be measured and the procedures to be followed when taking each measurement. Protocol files are described in part IV below. Part V of this document describes program features in more detail for the benefit of programmers wishing to adapt MEASURE to their own needs.

To save typing time, computer-readable copies of the program can be obtained for a nominal media and handling charge from the author. Please specify what media format you require.

**II: PROGRAM OVERVIEW**

The MEASURE program consists of 21 FORTRAN77 subprograms and two assembly language device drivers. A generalized flow-chart of the program is given in Figure 1. The program operates as follows. When the program is started, an initialization program reads the protocol file specified by the operator to obtain the number, types and names of the characteristics to be measured, the specific on-screen prompts to be displayed for each measurement step, the format to be used in the output data file, and several other items of information. This file allows most users of the system to custom design a standard set of measurements for their particular application without the need to modify the program itself. A description of the protocol file format and the options available to the user are given in Section IV of this technical report. Once the protocol file has been read, a data file is opened on the disc, and the main program begins. A central program flow module (NEXUS) first calls a subprogram which displays the menu of available commands, then calls a second subprogram to obtain a command from the user, and finally branches to the appropriate command module. Commands are entered by number from the keyboard of the computer, and may be entered in any order. The available commands are: save the the current set of measurements on the disc and begin a new specimen (NEXTSPEC), change one or more data items on the current specimen (EDITSPEC), delete all information about the current specimen (DELSPEC), measure a
characteristic of the specimen, or end the program (ENDMEAS). Each character specified in the protocol file is treated by the program as a separate command. The menu highlights the name of each character as it is measured. The menu also displays the total number of specimens measured and displays how many of each type of specimen have been measured. When the program is ended, the data file is closed and control returns to the computer's operating system.

The data file itself is a direct access file containing records in ASCII format. The program treats the specimen type, label, and full suite of character measurements as a single 'logical record' with a length specified by the user in the protocol file. The first 5 columns of this output record are reserved for the specimen type, but the remainder of the output data format is specified by the user in the protocol file. A logical record length can be of any value up to 4,000 spaces, and is determined by the user in the protocol file. The protocol file also determines whether the data for each specimen is to be written as one long line in the output file, or 'wrapped' and written as a number of shorter lines, with as many output lines per specimen as needed to contain the data. This feature is provided for users with FORTRAN compilers and data editors which limit the length of lines permitted in disc-files. Note also that direct access files may or may not be directly readable by your text editor or operating system. If this is a problem, the program can be easily modified to write a simple 'sequential' type file.
Figure 1

MEASURE PROGRAM OUTLINE

plus: DISTANCE BEEP CLEAN$ CLRSCRN CRTPRINT
III: MEASUREMENT ROUTINES

MEASURE has been designed so that it can be modified by the user to incorporate new types of measurement routines. The existing measurement routines however should be sufficient for many users. There are 5 routines included - three for digital measurements, and two for entry of data at the keyboard. All of these routines are invoked by typing the name of the routine, the prompt text lines, and the desired test criteria into the appropriate spots in the protocol file. The current version of the input routine for the HIPAD digitizer and LASICO uses the white button of a 4 button cursor to digitize data, the red button to end the measurement, and the green button to delete previous data points (this last feature is only used in the OUTL1 routine). INPUT returns an error flag if invalid data was read from the device drivers. (This is usually the result of the operator entering data too rapidly, or is caused by the wrong button being pressed on the tablet cursor). An error condition in LENG1 or LENG2 terminates the measurement. An error condition in OUTL1 causes the data point to be ignored, but the measurement is permitted to continue. Note that all measurements, including OUTL1, are entered with the tablet in 'point' mode.

- LENG1 -. The LENG1 measurement routine measures the distance between two digitized points and computes the distance in micrometers between them. Any number of repeat measurements can be taken. LENG1 will compute the average of the measurements. Three types of data checking are performed by this routine. As each pair of data points is digitized, the distance value obtained is checked to see if it is abnormally small or large. If it is, the program requests the user to either confirm the value or to ignore it. When the measurement loop is ended LENG1 checks to see if the minimum number of repetitions specified in the protocol file has been done. If not, the routine requests the operator to measure additional points, to confirm that the number measured is sufficient, or to abandon the measurement. If more than one measurement has been obtained, LENG1 checks the range of measurements against a value specified in the protocol file. If the range of measured values is too large, the operator is prompted to confirm the values, redo the measurement, or abandon the routine and return to the main menu. LENG1 returns a single value for the distance in FORTRAN G10.4 format.
• LENG2 -. LENG2 is a modification of the LENG1 routine. It differs only in that two or three data values are returned instead of one. The values can be the mean and standard deviation of the distances measured; the maximum and minimum values measured; the mean of the measured values and the range of the measurements taken; or the mean, standard deviation and the number of measurements taken. LENG2 returns a 20 character long string, with the data format determined by the option chosen. The option to be used is selected by the code entered into the protocol file, and cannot be altered once the program has been started.

• OUTL1 -. This routine records digitized data as a series of values in either 2 or 3 dimensions. Digitized points are scaled to micrometers and formatted as specified in the protocol file for output, but are otherwise unprocessed. Depending on the data format chosen, anywhere from 100 to 200 points may be digitized at a time. The program automatically computes the maximum number of points possible for a specified format, and will abandon the measurement if this number is exceeded. The protocol file can also be used to specify that a minimum number of points needs to be digitized. If the operator attempts to end the digitization prior to reaching this number, the program deletes the existing data and restarts the measurement. OUTL1 returns a 3,000 character long string, with the data beginning at position 11. The first character in the output string contains a 2 or a 3, indicating the number of dimensions in the data. Positions 3 to 5 contain N, the number of points digitized. Positions 6 through 9 contain the data format specified in the protocol file that was used to write the values (e.g., F6.2, etc.). Each digitized point consists of 2 or 3 floating point values separated from each other by commas, and separated from other points by a leading '/' character.

• LABEL and
LABL2 -. These two routines are used to enter data or comments from the keyboard of the computer. LABEL returns a 10 character long string, LABL2 a 30 character long string. (The reason why 2 different routines are used instead of one general purpose one has to do with the way MEASURE has been written, and with the way FORTRAN handles character data).
IV: PROTOCOL FILES

A protocol file is a standard ASCII sequential data file which contains information telling the MEASURE program the specific procedures to follow while measuring a suite of characteristics. A sample protocol file is listed in Figure 2. The first line of this file is a header line identifying the file. The second line gives the length of output records in the data file in columns 1 to 5, and the name of the measurement protocol in columns 21 to 40. This name is written at the beginning of all data files output by the program. The third line of the file gives the length of each logical record in columns 1 to 5.

The remainder of the protocol file is organized into groups of 4 lines, one group per character. The first line in the group contains, in order: the character number (in columns 1 to 4); the name of the character (columns 5 to 19); the character type (i.e. the measurement routine which the GETMSUB subprogram should call) in columns 20 to 24; the beginning column of the logical record where the character measurement should be written (in protocol columns 25 to 29); the end column of the output data-field (protocol file columns 30 to 34); the magnification to be used (columns 35 to 39); the minimum number of repeated measurements needed to complete the measurement routine (columns 40 to 44); and in columns 45 to 54, a floating point number which can be used as a test criterion by the measurement routine to screen out invalid data. LENG1 and LENG2 use this number to test the variability of multiple measurements, while OUTL1 does not use this number. The next two lines of each group contain text in columns 1 to 79 which can be used by the measurement routine in any fashion. The measurement routines included in this report use these two lines to store the text prompts that are displayed on the computer screen to guide the operator in the measurement. The last line in each group of 4 lines can also be used by the measurement routine in any way that the programmer desires. Each of the existing routines uses this line in a different way.

- LENG1 uses positions 60 to 79 to hold 2 F10.x numbers representing the minimum and maximum permissible values for each measurement, and expects to find a '2' or a '3' in column 80, indicating whether the measurement is to be made in 2 or 3 dimensions.

- LENG2 uses positions 1 and 2 to specify what type of data set is to be returned. The permitted values are 'SD' for mean and standard deviation, 'HL' for the maximum and minimum values, 'RN' for the mean and the range, and 'EV' for mean, standard deviation and number of measurements taken. Any other characters in these two columns are equivalent to option
'EV'. Column 5 contains a '2' or a '3' to indicate whether 2 or 3 dimensional data is to be taken. Positions 60 through 79 contain two F10.x numbers representing, as in LENG1, the minimum and maximum permissible values for each measurement.

- OUTL1 uses columns 75 to 78 to specify what FORTRAN format to use when writing floating point data values for output. Column 76 specifies how many spaces to allot for each point, and must be a number between 2 and 9. Thus F5.4 and F9.2 are permissible, but F12.6 is not. Column 80 should contain a '2' or a '3' to determine whether 2 or 3 dimensional data points are to be recorded.

LABEL and LABEL2 do not make any use of the fourth line.

Protocol files can be created with any ordinary text editor or word processor that will produce ASCII text as output. The column alignment defined above and shown in the sample file must be followed exactly, and all lines in the file must be at least 80 columns long. The MEASURE program will fail to run (i.e. crash) if the protocol file is not properly laid out! Note also that the beginning and ending positions in the logical record specified for the output from each subroutine must equal the length of the data returned by that routine. LENG1 for example, returns a 10 character long value, and thus the end position minus the beginning position specified in the protocol file must equal 10. Failure to observe this restriction will probably cause the program to fail in exciting and unpredictable ways!
A Sample protocol file with one each of Label, Length and OUTL1 formats
80  Column Records.Example1.prt:10/86 is name of this protocol file.
3100 is the length of each Logical Record.
1 Stage Position LABEL  6  15
Inter stage position of specimen on slide. Format is 00.0/000.0

2 Comments.......LABL2 16  45
Note any special features of the specimen or difficulties in the measurement here. Keep your comments to no more than 30 spaces please!

3 Shell Diameter LENGL  46  55  60X  3  5.0
Measure the diameter of the shell across the outer whorl. Digitize 1st Edge. Digitize the opposite edge. Take at least three measurements of diameter.

50.0   150.0  3

4 Pore dimensionsLENGL  56  75  100X  10  3  1.5
Measure at least 10 pores near edge of shell. Digitize on edge of a pore. Digitize the opposite edge.

IV  3

5 Shell outline OUTL1  76  3075  60X  50
Begin digitization at pylome edge and continue to opposite edge of pylome. WHITE button to digitize, GREEN to delete a point, RED to end measurement.

F6.1  3
V: CUSTOMIZATION

To use the MEASURE program, it must be first customized to the particular computer and measurement peripherals employed. It must also be configured to support the particular types of measurements desired. Once these customization steps have been accomplished, the particular set of characteristics to be measured can be specified via the use of a protocol file.

Hardware related customization -. MEASURE is currently configured to run on a NEC APC microcomputer under the CP/M-86 operating system, and to use assembly language device drivers to obtain x and y coordinates from a Houston Instruments HIPAD digitizer tablet, and vertical coordinates from a Los Angeles Scientific XE processor connected via an RS-232 serial interface. The operation of this system is explained in Lazarus (1986). To modify the program to work with other hardware, certain machine specific parts of the code will have to be rewritten. To the maximum extent possible, all machine-specific portions of the program have been isolated in a small number of subroutines. These routines are: INPUT, CRTPRINT, CLRSCRN, and (possibly) BEEP.

INPUT is the portion of the program which calls (via assembly code) the measurement peripherals, decodes the input, and converts the data into standard measurement units. The present version of input uses the assembly routines HIPAD and LASICO, written by Ken Prada of the Woods Hole Oceanographic Ocean Engineering Department, to access the digitizer tablet and vertical measurement device. LASICO returns an ASCII string representing the vertical motion of the microscope stage in an integer form. It is converted into a micrometer value by the scaling factor ZSCALE, which has been calibrated to the particular microscope employed in the measurement system. HIPAD also returns a character variable (XY$) which contains, in ASCII form, an x and y coordinate, and an ASCII code representing the cursor status. INPUT uses scaling factors to convert the peripheral inputs into micrometers, and returns a 3 element real array P containing the scaled X, Y, and Z coordinates. The effective magnification or scaling between the digitizer tablet and microscope will vary with different optical arrangements of the microscope, and particularly with the objective used for the measurement. The character variable MAG is used to notify the operator which optical arrangement to use for each measurement, and is also used by INPUT to select the proper scaling constant to use in converting XY$ into micrometers in X and Y. The value of MAG should be specified for each character measurement in the protocol file. The HIPAD tablet cursor contains four buttons which are used to specify the beginning
and termination of measurement data. INPUT decodes the status character and returns a single status letter to the main measurement routine via the character variable STATUS. 'G' (the white button) stands for normal data entry, 'S' (the red button) serves as an end of data flag, and 'D' (green button) is used by OUTL1 to delete data points. INPUT will probably require extensive modification for use on other computers and input devices, although modifications should be minimal so long as a Houston Instruments HIPAD digitizer tablet is used as the x-y measurement device.

CRTPRINT is employed by the main program to write text in various fonts at specified row and column positions of the computer screen. The main (calling) program sends the text string, the length of the string, the row and column position, and an integer specifying which font is to be used for printing. It is assumed that whatever computer is used will possess these minimal video attributes. The current version of CRTPRINT sends escape and control code sequences to produce reverse video and blinking video characters in the form used by the NEC APC. These are very similar to ANSI standard (DEC VT100) code. For most other computers, this routine can be customized simply by changing the escape codes.

CLRSCRN is a subprogram with no parameters, which, when called, clears the computer screen and places the cursor at the upper left corner. Again, the present version is designed to work with the NEC APC (virtually identical with DEC's VT100 standard), and can usually be customized by a simple modification of the control code sequence.

BEEP sounds the speaker of the computer N times by use of the ASCII bell character, control-G. This routine should work without the need for modification on most computers, but may require special code on some machines.

Other Software modification -. MEASURE has been written as a number of small modules to make it easier to modify. Existing functions can be expanded without the necessity to redo major portions of the program. The GETCMD subprogram for example at present expects the user to select a command by typing the number of the command at the computer keyboard. The GETCMD subprogram can be modified to accept input from the digitizer tablet, or from some other input device, without affecting other portions of the program.

The most important modification which most users are likely to make is the addition of special measurement routines. Many types of morphometric analysis exist which require specialized measurement algorithms. Examples
include area and angle measurement, and measurement of textural characteristics from digital video images. One 'dummy' routine, USER1, has already been provided in the program, and can be made functional with the addition of the appropriate code. Users who wish to add other routines to the MEASURE program will need to modify the GETCMD procedure by the addition of a CALL statement to the new routine, with the passing of the appropriate parameters. Note that many of the parameters passed are defined in the protocol file. These protocol file parameters are meant to allow a general purpose measurement routine to be used to measure a variety of different specific characteristics. Because the variables containing these parameters are passed through a chain of subprograms within the main program, adding a new variable to contain a new type of parameter is not an easy task. Provision has been made for this problem in the variables used to pass parameters. In addition to providing a variety of variables for standard aspects of a character measurement (name, test criteria, prompts, etc.), the user can pass many parameters from the protocol file to the target measurement routine as substrings within the character variable MS$. The routine can decode the substrings via internal reads, or by string comparisons. The existing measurement routines all make use of this feature. Note also that MS$(241:300)$ is also available for passing parameters between subroutines, although it cannot be used to load parameters from a protocol file.

VI: ACKNOWLEDGEMENTS

I wish to thank Ken Prada for help in interfacing computer peripherals, and Peter Shaw and Pat Lohmann for reviewing the text. This work supported under NSF grants DEB81-18743 and DPP83-17087.

VII: REFERENCES

PROGRAM INITMEAS

INITIALIZATION ROUTINE FOR MEASURE PROGRAM. READS A PROTOCOL
FILE TO OBTAIN CHARACTER AND OUTPUT FORMAT INFO, THEN CALLS NEXUS.
ALL VARIABLES PASSED IN CALL STATEMENT.

REQUIRES SUBROUTINES CLEAN$, BEEP, NEXUS

BY DAVE LAZARUS. VERSION 3.1 10-6-86.
MODIFIED TO READ AND PASS LREC TO NEXTSPEC VIA NEXUS.
THIS VERSION COMPILED USING DR FORTRAN V.4.1.

INTEGER MAXVEC, MAXC, MAXTAX, MAXSPEC, MAXMS$
PARAMETER (MAXC=40, MAXVEC=4000, MAXTAX=40, MAXSPEC=500,
          MAXMS$=300)

INTEGER
   CB(MAXC),
   CE(MAXC),
   CNUM(MAXC),
   ERROR,
   I,
   LDATE,
   LDILNAM,
   LPFITNAM,
   LPROTNAM,
   LREC,
   LSNAME,
   LVEC,
   MINN(MAXC),
   NCHARS,
   NSPEC

REAL
   TEST(MAXC)

CHARACTER
   A1$*1,
   A79$*79,
   CNAME(MAXC)*15,
   CTYPE(MAXC)*5,
   DATE*50,
   DFILNAM*50,
   EOS$*1,
   MAG(MAXC)*5,
   MS$(MAXC)*300,
   PFITNAM*50,
   PROTNAM*50,
   SNAM*160,
   SVEC*(MAXVEC),
   TAXCODE(MAXSPEC)*5

LOGICAL
   BEGIN,
   CMEASURD(MAXC),
   LABELED

C
C INITIALIZE VARIABLES
C
Labeled=FALSE.
BEGIN=.TRUE.
LPfILNAM=0
NSPECS=1
1000 FORMAT(A)
EOS=CHAR(10)
DO 7 I=1,MAXSpec
TAXCODE(I)='
7 CONTINUE
DO 6 I=1,MAXC
CMeasureD(I)=.FALSE.
CNAME(I)='
CTYPE(I)='
6 CONTINUE
DO 10 I=1,MAXVEC
Svec(I,I)='
10 CONTINUE
C
DISPLAY START-UP MESSAGE.
C
WRITE(6,*) ' MICROMEASURE:
WRITE(6,*) ' A Measurement System for Microfossils.'
WRITE(6,*) ' By Dave Lazarus. Version: 10-10-86.'
WRITE(6,*) ' MICROMEASURE is a morphometric data acquisition'
WRITE(6,*) ' program which collects data on specimens for'
WRITE(6,*) ' subsequent analysis by the user. The program is'
WRITE(6,*) ' documented in:
WRITE(6,*) ' Lazarus, D.B. 1986. MICROMEASURE: A program'
WRITE(6,*) ' for microfossil morphometrics. WHOI Technical'
WRITE(6,*) ' Reports.'
WRITE(6,*) ' This program reads a user created ASCII formatted'
WRITE(6,*) ' disc-file to define the measurement names and'
WRITE(6,*) ' measurement types, as well as the structure of'
WRITE(6,*) ' the data in the output file.'
WRITE(6,*) ' Enter the name of the protocol file you wish to'
WRITE(6,*) ' use ->:
READ(5,1000) PFILNAM
CALL CLEAN$(EOS,PFILNAM,LPfILNAM)
C
OPEN FILE
C
OPEN(UNIT=3,FILE=PFILNAM(1:LPfILNAM),STATUS='OLD',ERR=11,
! IOSTAT=ERROR)
11 IF(ERROR.GT.0) THEN
CALL BEEP(5)
WRITE(6,*) 'ERROR 'ERROR,' OPENING ',PFILNAM(1:LPfILNAM),''
WRITE(6,*) 'TRY AGAIN...'
GOTO 3
END IF
C
PROTOCOL FILE OPENED OK. READ DATA.
C
2 CONTINUE
READ(3,1000) A79$
WRITE(6,*) A79$
READ(3,101) LREC,PROTNAM
101 FORMAT(15,15X,(A))
CALL CLEAN$(EOS,PROTNAM,LPROTNAM)
READ(3,102) LVEC
102 FORMAT(15)
DO 4 I=1,MAXC
READ(3,103,END=5) CNUM(I),CNAME(I),CTYPE(I),CB(I),CE(I),
! MAG(I),MINN(I),TEST(I)
READ(3,1000) MS$(I)(1:80)
READ(3,1000) MS$(I)(81:160)
READ(3,1000) MS$(I)(161:240)
103 FORMAT(14,A15,A5,I5,I5,A5,I5,F10.3)
WRITE(6,*) 'CHARACTER ',I,'.',',CNAME(I)', READ.'
4 CONTINUE
5 NCHARS=I-1
CLOSE(3)
WRITE(6,*) 'Protocol file ',PFILNAM(1:LPFILNAM),
! ' Successfully read.'
WRITE(6,*) 'Protocol name is ',PROTNAM
WRITE(6,*) 'There are ',NCHARS, ' characters.'
C
C OPEN DATA FILE FOR OUTPUT

WRITE(6,*)
WRITE(6,*) 'DATA FILE NAME SHOULD NOT HAVE BLANKS OR SYMBOLS!' 9
WRITE(6,*) 'ENTER NAME OF NEW DATA FILE ->: '
READ(5,1000) DFILNAM
CALL CLEAN$(EOS,DFILNAM,LDSPFILNAM)
OPEN(UNIT=1,FILE=DFILNAM(1:LDSPFILNAM),STATUS='NEW',
! ACCESS='DIRECT',RECL=LREC,FORM='FORMATTED',ERR=9)
WRITE(6,*) 'FILE OPENED...'
8 WRITE(6,*) 'ENTER DATE AND TIME, 20 CHAR MAX. EX: ',
! 'TUES 10/11/85 23:06'.
WRITE(6,*) 'DATE? ->: '
READ(5,1000) DATE
CALL CLEAN$(EOS,DATE,LDSPFILNAM)
WRITE(6,*)
WRITE(6,*) 'ENTER THE NAME OF THE SAMPLE (80 CHAR MAX):'
WRITE(6,*)
READ(5,1000) SNAM
CALL CLEAN$(EOS,SNAM,LSNAM)
WRITE(UNIT=1,REC=1,FMT=104) SNAM(1:80)
WRITE(UNIT=1,REC=2,FMT=105) PROTNAM(1:20),DATE(1:20)
104 FORMAT(A80)
105 FORMAT(A20,40X,A20)
CLOSE(1)
WRITE(6,*) 'FILE ',DFILNAM(1:LDSPFILNAM), ' READIED FOR OUTPUT:'
PAUSE 'PRESS ANY KEY TO BEGIN MAIN PROGRAM....'
C
C GET LABEL OF 1RST SPECIMEN.
C
WRITE(6,'*') 'ENTER LABEL OF FIRST SPECIMEN (MAX OF 5 CHAR$)*>&: ' 
READ(5,1000) A79$
CALL CLEAN$(EOS,A79$,I)
SVEC(1:5)=A79$(1:5)
TAXCODE(1)=A79$(1:5)
C
CALL NEXUS(EOS,SVEC,LVEC,P$FILNAM,MAXC,MAXVEC,MAXSPEC,MAXTAX, 
! L$FILNAM,P$ROTNAM,L$ROTNAM,N$CHARS,C$NUM,C$NAME,C$TYPE,C$B,C$E, 
! BE$GIN,D$FILNAM,L$DFILNAM,NS$PECS,D$ATE,L$DATE,C$MEASURD, 
! MAG,MINN,T$EST,MS$,TAXCODE,L$REC)
END
SUBROUTINE NEXUS(EOS,SVEC,LVEC,PFLNAM,MAXC,MAXVEC,MAXSPEC, 
   ! MAXTAX,LFILNAM,PROTNAM,LPFILNAM,NCHARS,CNUM,CNAME,CTYPE,  
   ! CB,CE,BEGIN,DFILNAM,LDFILNAM,NSPECS,DATE,LDATE,CMEASURD, 
   ! MAG,MINN,TEST,MS$,$TAXCODE,LREC)

COMMAND PROCESSOR FOR MEASURE PROGRAM. SHOWS MENU AND GETS 
COMMAND, THEN BRANCHES TO CORRECT SUBROUTINE. COMMANDS -1 TO -4 
ARE RESERVED FOR HOUSEKEEPING, COMMANDS >0 FOR MEASUREMENTS.

USES SUBROUTINES MENU,GETCMD,NEXTSPEC,EDITSPEC,DELSPEC,ENDMEAS, 
GETMSUB

BY DAVE LAZARUS. VERSION 2: 7-24-86. (PASS LREC TO NEXTSPEC).

INTEGER MAXC,MAXSPEC,MAXTAX,MAXVEC
INTEGER
   ! CB(MAXC),
   ! CE(MAXC),
   ! CMD,
   ! CNUM(MAXC),
   ! LDATE,
   ! LDFILNAM,
   ! LPFILNAM,
   ! LPROTNAM,
   ! LREC,
   ! LVEC,
   ! MINN(MAXC),
   ! NCHARS,
   ! NSPECS
REAL
   ! TEST(MAXC)
CHARACTER
   ! CNAME(MAXC)*15,
   ! CTYPE(MAXC)*15,
   ! DATE*50,
   ! DFILNAM*50,
   ! EOS*1,
   ! MAG(MAXC)*5,
   ! MS$(MAXC)*300,
   ! PFILNAM*50,
   ! PROTNAM*50,
   ! SVEC*(MAXVEC),
   ! TAXCODE(MAXSPEC)*5
LOGICAL
   ! BEGIN,
   ! CMEASURD(MAXC),
   ! LABELED

CONTINUE
CALL MENU(NSPECS,DFILNAM,PFLNAM,DATE,CNUM,CMEASURD,CNAME, 
   ! NCHARS,MAXC,MAXSPEC,MAXTAX,TAXCODE)
CALL GETCMD(CMD)

IF(CMD.EQ.-1)
! CALL NEXTSPEC(E0$,SVEC,LVEC,CMD,NCHARS,NSPECS,CMEASURD,
! LABELED,MAXC,MAXVEC,TAXCODE,MAXSPEC,DFILNAM,LDFILNAM,LREC)
! IF(CMD.EQ.-2)
! CALL EDITSPEC(SVEC,LVEC,CNAME,NCHARS,NSPECS,CB,CE,
! CMEASURD,MAXC,MAXVEC,E0$,CMD,TAXCODE,MAXSPEC)
! IF(CMD.EQ.-3)
! CALL DELSPEC(SVEC,LVEC,NCHARS,CMEASURD,LABELED,CMD,MAXVEC,
! MAXC,TAXCODE,MAXSPEC,NSPECS)
! IF(CMD.EQ.-4)
! CALL ENDEMEAS(CMEASURD,NCHARS,NSPECS,MAXSPEC,DFILNAM,
! LDFILNAM,CMD,E0$,MAXC,MAXVEC,TAXCODE)
! IF(CMD.GT.0)
! CALL GETMSUB(CMD,CNUM,CNAME,CTYPE,CB,CE,SVEC,LVEC,E0$,
! MAXC,MAXVEC,BEGIN,NCHARS,PFLNAM,LPFLNAM,
! MAG,MINN,TEST,MS$,CMEASURD)

C GOTO 1
RETURN
END
SUBROUTINE MENU(NSPECS,DFILNAM,PFILNAM,DATE,
CNUM,CMEASURD,CNAME,NCHARS,MAXC,MAXSPEC,MAXTAX,TAXCODE)

MENU DISPLAY FOR MEASURE.F77. SHOWS ACTIVE FILE AND PROTOCOL,
LIST OF TAXA MEASURED SO FAR, CHARACTERS MEASURED ON
CURRENT SPECIMEN, AND AVAILABLE COMMANDS.

USES SUBROUTINES CRTPRINT,CLRSCRN

CREATED 1985 BY DAVE LAZARUS. VERSION: 4-24-85.

INTEGER MAXC,MAXSPEC,MAXTAX
INTEGER
!   CNUM(MAXC),
!   COL,
!   I,
!   IFONT,
!   J,
!   NCHARS,
!   NSPECS,
!   NUMTAX,
!   NTAXON(MAXTAX),
!   ROW
CHARACTER
!   CALLS$15,
!   CNAME(MAXC)$15,
!   DATE$50,
!   DFILNAM$50,
!   NULLS$1,
!   PFILNAM$50,
!   TAXCODE(MAXSPEC)$5,
!   TAXLIST(MAXTAX)$5
LOGICAL
!   CMEASURD(MAXC)

WRITE STATUS LINE

CALL CLRSCRN
WRITE(6,101) NSPECS,DFILNAM(1:22),PFILNAM(1:20),DATE(1:20)
101 FORMAT(1X,'#',J3.3,2X,'DF:',A22,'PROT:',A20,A20)

SHOW TOTALS FOR TAXA MEASURED SO FAR

TALLY CODES

DO 11 I=1,MAXTAX
   NTAXON(I)=0
   TAXLIST(I)=
11 CONTINUE

NUMTAX=1
TAXLIST(1)=TAXCODE(1)
NTAXON(1)=1
DO 5 I=2,NSPECS
   DO 6 J=1,NUMTAX
IF(TAXCODE(I).EQ.TAXLIST(J)) THEN
   NTAXON(J)=NTAXON(J)+1
   GOTO 5
END IF
6
   CONTINUE
   NUMTAX=NUMTAX+1
   TAXLIST(NUMTAX)=TAXCODE(I)
   NTAXON(NUMTAX)=1
5
   CONTINUE

DISPLAY CODES

WRITE(6,*)
   DO 7 I=1,40,8
      WRITE(6,103) (TAXLIST(I+J),NTAXON(I+J),J = 0,7)
   103   FORMAT(8(1X,A5,':',I3))
   CONTINUE

DISPLAY COMMAND OPTIONS

WRITE(6,*)
   WRITE(6,*) 'NEW SPECIMEN : EDIT SPECIMEN :',
   'DELETE SPECIMEN : END PROGRAM'

DISPLAY CHARACTER MEASUREMENT OPTIONS. FIRST WRITE NUMBERS,
THEN FILL IN CHARACTER NAMES USING CRTPRINT. USE HILITE
VIDEO FOR CHARACTERS ALREADY MEASURED.

DO 1 I=1,40,4
   WRITE(6,102) (I+J, J=0,3)
102   FORMAT(4(1X,I2,':',15X))
   CONTINUE

I=1
DO 2 ROW=11,20
   DO 3 COL=5,62,19
      IF(CMEASURD(I)) THEN
         IFONT=2
      ELSE
         IFONT=1
      END IF
      CALL$=CNAME(I)
      CALL CRTPRINT(ROW,COL,CALL$,15,IFONT)
   3   I=I+1
   IF(I.GT.NCHARS) GOTO 4
2   CONTINUE
3
   CONTINUE
   END OF SUBROUTINE
4
   CONTINUE
   CALL CRTPRINT(21,10,NULL$,1,1)
   RETURN
END
SUBROUTINE DELSPEC(SVEC,LVEC,NCHARS,CMEASURD,LABELED,CMD,
MAXVEC,MAXC,TAXCODE,MAXSPEC,NSPECS)

SUBROUTINE TO DELETE ALL DATA FOR CURRENT SPECIMEN.

CREATED 1985 BY DAVE LAZARUS. VERSION: 4-24-85.

INTEGER MAXC,MAXVEC,MAXSPEC
INTEGER LVEC,NCHARS,I,CMD,NSPECS
CHARACTER SVEC*(MAXVEC),ANSWER*3,TAXCODE(MAXSPEC)*5
LOGICAL CMEASURD(MAXC),LABELED

1000 FORMAT(A)

CMD=0

WRITE(6,*) 'CONFIRM: DELETE CURRENT SPECIMEN? (YES/NO) ->: '
READ(5,1000) ANSWER
IF (ANSWER.EQ.'YES'.OR.ANSWER.EQ.'yes') THEN
   CONTINUE
ELSE
   RETURN
END IF

DO 1 I=1,LVEC
   SVEC(I:I)=''
   CONTINUE
1 DO 2 I=1,NCHARS
   CMEASURD(I)=.FALSE.
   CONTINUE
2 CONTINUE
   TAXCODE(NSPECS)=''
   LABELED=.FALSE.

RETURN
END
SUBROUTINE EDITSPEC(SVEC,LVEC,CNAME,NCHARS,NSPECS,CB,CE,
CMEASURED,MAXC,MAXVEC,EOS,CMD,TAXCODE,MAXSPEC)

SUBROUTINE TO SELECTIVELY DELETE VALUES, OR CHANGE LABEL.

USES CLEAN$,CLRSCRN,BEEP

4-18-85 BY DAVE LAZARUS.

INTEGER MAXC,MAXVEC,MAXSPEC
INTEGER
CB(MAXC),
CE(MAXC),
CHAR,
CMD,
I,
J,
K,
LVEC,
NCHARS,
NSPECS,
T1,
T2

CHARACTER
CNAME(MAXC)*15,
EOS*1,
OUT$(2)*20,
SVEC*(MAXVEC),
TAXCODE(MAXSPEC)*5

LOGICAL
CMEASURED(MAXC)

1000 FORMAT(A)

CMD=0

SHOW CURRENT VALUES. TRUNCATE CHARACTER FIELDS TO 20 SPACES
LONG IF NEEDED. UNMEASURED CHARACTERS DISPLAY AS BLANK.

CALL CLRSCRN
WRITE(6,*) 'EDIT CURRENT SPECIMEN ROUTINE.'
WRITE(6,*)
WRITE(6,*) 'EDIT SPECIMEN #',NSPECS,' ,LABEL= ',SVEC(1:5)

DO 1 I=1,NCHARS-1,2
  DO 2 J=0,1
    OUT$(I+J)=
    IF((I+J).LE.NCHARS) THEN
      K=MIN(19,CE(I+J)-CB(I+J))
      T1=CB(I+J)
      T2=T1+K
      WRITE(OUT$(I+J),1000) SVEC(T1:T2)
    END IF
  2 CONTINUE
WRITE(6,101) I,CNAME(I),OUT$(I),I+1,CNAME(I+1),OUT$(2)
101      FORMAT(2(1X,I2,':',A15,':',A20))
          CONTINUE
C
C      GET NUMBER FROM USER AND EXECUTE
C
3      WRITE(6,102) NCHARS
102     FORMAT(1X, '(0)':CHANGE LABEL. (1-',I2, ')':DELETE CHAR N:',
           '(NEG.#):EXIT TO MAIN. #? -->: ')
          READ(5,*,ERR=3) CHAR
C
      IF(CHAR.LT.0) GOTO 6
      IF(CHAR.EQ.0) THEN
          WRITE(6,*,'(ENTER NEW LABEL (5 CHAR.MAX.) ->: ')
          READ(5,1000) OUTS(1)
          CALL CLEANS(EOS,OUTS(1),K)
          SVEC(1:5)=OUTS(1)(1:5)
          TAXCODE(NSPECS)=OUTS(1)(1:5)
      END IF
      IF(CHAR.GT.0.AND.CHAR.LT.NCHARS) THEN
          DO 4 I=0,CE(CHAR)-CB(CHAR)
              T1=CB(CHAR)+I
              SVEC(T1:T1)=''
        4      CONTINUE
      CMEASURD(CHAR)=.FALSE.
      END IF
6      CONTINUE
      RETURN
END
SUBROUTINE NEXTSPEC(E0$,SVEC,LVEC,CMD,NCHARS,NSPEC5,CMEASURED,LABELED,MAXC,MAXVEC,TAXCODE,MAXSPEC,DFILNAM,LDFILNAM,LREC)
!
WRITES CURRENT SPECIMEN TO DISC FILE AND UPDATES COUNTERS.
USES SUBROUTINES BEEP,CLEAN$'

BY DAVE LAZARUS. VERSION 2.1:9-29-86
...LREC SENT FROM MAIN PROG, SKIPS TXCODE ENTRY IF CMD IS FROM
ENDMEAS (-4). THIS VERSION COMPILED USING DR FORTRAN 4.1.

INTEGER MAXC,MAXVEC,MAXSPEC
INTEGER

CMD,
I,
INDEX,
J,
K,
LDFILNAM,
LREC,
LVEC,
N,
NCHARS,
NSPEC5,
R,
REMAINEAV.

SET

CHARACTER SVEC*(MAXVEC),IN$*20,TAXCODE(MAXSPEC)*5,DFILNAM*50,

EO$*1

LOGICAL CMEASURED(MAXC),LABELED

1000 FORMAT(A)

N=INT(LVEC/LREC)
REMAINEV=LVEC-N*LREC
SET=N
IF(REMAINEV.GT.0) SET=SET+1
OPEN(UNIT=1,FILE=DFILNAM(1:LDFILNAM),ACCESS='DIRECT',

! FORM='FORMATTED',RECL=LREC)
R=2+(NSPEC5-1)*SET
DO 4 I=1,SET
K=I*LREC
J=K-LREC+1
WRITE(UNIT=1,FMT=1000,REC=R+I) SVEC(J:K)
4 CONTINUE
CLOSE(1)
NSPEC5=NSPEC5+1
IF(NSPEC5.GT.MAXSPEC) THEN
CALL BEEP(5)
WRITE(6,*) 'MAXIMUM SPECIMEN COUNT EXCEEDED!'
WRITE(6,*) 'CLOSING FILE AND ENDING RUN!'
CALL ENDMEAS(CMEASURED,NCHARS,NSPEC5,MAXSPEC,DFILNAM,

! LDFILNAM,CMD,E0$,MAXC,MAXVEC,TAXCODE)
END IF
C
DO 3 I=1,LVEC
   SVEC(I:I)=''
3 CONTINUE
C
DO 1 I=1,NCHARS
   CMEASURED(I)=.FALSE.
1 CONTINUE
C
   WRITE(6,*) 'ENTER LABEL FOR NEW SPECIMEN (MAX OF 5 CHARs) ->: ' 
   READ(5,1000) IN$ 
   CALL CLEAN$(EO$,IN$,I) 
   SVEC(1:5)=IN$(1:5) 
   TAXCODE(NSPECS)=IN$(1:5) 
   LABELED=.TRUE. 
   RETURN 
C
2 CALL BEEP(5) 
   WRITE(6,*) 'ERROR ATTEMPTING TO SAVE SPECIMEN TO FILE!' 
   PAUSE 'HIT ANY KEY TO CONTINUE.' 
   RETURN 
END
SUBROUTINE ENDMEMS(CMEASURD,NCHARS,NSPECs,MAXSPEC,DFILNAM,
LDILNAM,CM,D,EO$,MAXC,MAXVEC,TAXCODE)

C CHECKS TO SEE IF CURRENT SPECIMEN SAVED, PROMPTS USER AND ABORTS IF IT HASN'T BEEN ALREADY. IF OK TO DO SO, CLOSES DATA FILE AND EXITS TO THE OPERATING SYSTEM.

C BY DAVE LAZARUS. VERSION 2.0. 10-6-86.
C DOES NOT ATTEMPT TO SAVE LAST SPECIMEN AUTOMATICALLY, BUT PROMPTS USER INSTEAD.
C THIS VERSION COMPILED USING DR FORTRAN V.4.1.

C USES SUBROUTINE BEEP.

INTEGER MAXC,MAXSPEC,MAXVEc
INTEGER NCHARS,LDFILNAM,L,CM,MAXVEc,NSPECs
CHARACTER DFLNAM*50, ANSWER*3, EO$*1, SVEc*(MAXVEc),
TAXCODE(MAXSPEC)*5
LOGICAL CMEASURD(MAXC),LABELED

1000 FORMAT(A)

C IF(NSPECs.GT.MAXSPEC) GOTO 2

C DO 1 I=1,NCHARS
  IF(CMEASURD(I)) THEN
    CALL BEEP(5)
    WRITE(6,*) 'SAVE CURRENT SPECIMEN FIRST!'
    RETURN
  END IF
1 CONTINUE
WRITE(6,*) 'CONFIRM: END MEASUREMENTS PROGRAM? (YES/NO) ->:
READ(5,1000) ANSWER
IF(ANSWER.EQ.'YES'.OR.ANSWER.EQ.'yes') THEN
  CONTINUE
ELSE
  RETURN
END IF

C 2 CONTINUE
CLOSE(1)
WRITE(6,*) 'MEASUREMENT RUN COMPLETED.'
WRITE(6,*) 'FILE ',DFILNAM(1:LDFILNAM), ' CONTAINS YOUR DATA.'
WRITE(6,*) '...BYE...' STOP
END
SUBROUTINE GETCMD(CMD)
INTEGER CMD
1   WRITE(6,*) 'ENTER A NUMBER TO SELECT A COMMAND ->: '
    READ(5,*,ERR=2) CMD
    RETURN
2   CALL BEEP(3)
    GOTO 1
END
SUBROUTINE GETMSUB(CHAR,CNUM,CNAME,CTYPE,CB,CE,SVEC,LVEC,EO$,
   MAXC,MAXVEC,BEGIN,NCHARS,PFILNAM,LPFILNAM,
   MAG,MINN,TEST,MS$,CMEASURD)
!
BRANCHES TO MEASUREMENT SUBROUTINE CTYPE OF CHAR. EACH CHAR-
ACTER HAS A CTYPE ASSOCIATED WITH IT VIA THE PROTOCOL FILE.
PERMITTED CTYPES ARE:

LABEL: GETS A 10 CHARACTER LONG STRING FROM THE KEYBOARD.
LABL2: GETS A STRING OF LENGTH 30 FROM KEYBOARD.
LENGL: MEASURES LENGTH BETWEEN 2 POINTS IN 3-SPACE, WITH
MAGNIFICATION MAG(CHAR),MAXN(CHAR) REPEATS OF THE
MEASUREMENT, TEST(CHAR) PERMITTED VARIANCE IN THE
REPITITIONS, AND PROMPTS CONTAINED IN MSS(CMD). RETURNS
A SINGLE FLOATING POINT NUMBER IN A 10 CHARACTER LONG OUT$.
LENGL2: MEASURES LENGTH AS IN LENGL, BUT RETURNS 2 VARIABLES.
THE OUTPUT STRING IS THUS 2 TIMES AS LONG (IE LENGTH=20).
SEVERAL OPTIONS FOR THE CONTENT OF THE VARIABLES EXIST,
AND ARE SELECTED BY A 2 CHARACTER CODE IN MSS(239:240).
OUTL1: DIGITIZES OUTLINES IN 2 OR 3 DIMENSIONS, RETURNING DATA IN
A 3K LONG OUTPUT STRING. MSS(235:239) CONTAINS FORMAT INFO.
MSS(240) SETS 2-D OR 3-D MODE FOR DATA ENTRY.
USER1: DUMMY SUBROUTINE WHICH CAN BE EXPANDED BY THE USER INTO
A CUSTOM MEASUREMENT ROUTINE.

ALL SUBROUTINES ARE EXPECTED TO RETURN A STRING VALUE IN SVEC
WHICH REPRESENTS THE FINAL MEASUREMENT. THE LENGTH OF THE
STRING IN THE SUBROUTINE MUST BE THE SAME AS THAT SPECIFIED BY
THE LENGTH CE(CHAR)-CB(CHAR).

REQUIRES SUBROUTINE BEEP, MEASUREMENT SUBS SPECIFIED ABOVE.

WRITTEN 1985,1986 BY DAVE LAZARUS.
VERSION 4.0 TO INCLUDE OUT1 CREATED 10-1-86.

INTEGER MAXC,MAXVEC
INTEGER
  !
  CB(MAXC),
  CE(MAXC),
  CHAR,
  CNUM(MAXC),
  I,
  J,
  LPFILNAM,
  LVEC,
  MINN(MAXC),
  NCHARS
REAL
  !
  TEST(MAXC)
CHARACTER
   CNAME(MAXC)*15,
   CTYPE(MAXC)*5,
   EOS*1,
   MAG(MAXC)*5,
   MSS(MAXC)*300,
   PFLNAM*50,
   SVEC*(MAXVEC)
LOGICAL
   BEGIN,
   CMEASURD(MAXC)
C
   IF(CHAR.GT.NCHARS) THEN
      CALL BEEP(3)
      RETURN
   END IF

   IF(CTYPE(CHAR).EQ.'LENG1') THEN
      I=CB(CHAR)
      J=CE(CHAR)
      CALL LENG1(SVEC(I:J),CHAR,CNAME(CHAR),MAG(CHAR),MINN(CHAR),
               TEST(CHAR),MSS(CHAR),CMEASURD(CHAR))
   END IF

   IF(CTYPE(CHAR).EQ.'LENG2') THEN
      I=CB(CHAR)
      J=CE(CHAR)
      CALL LENG2(SVEC(I:J),CHAR,CNAME(CHAR),MAG(CHAR),MINN(CHAR),
               TEST(CHAR),MSS(CHAR),CMEASURD(CHAR))
   END IF

   IF(CTYPE(CHAR).EQ.'LABEL') THEN
      I=CB(CHAR)
      J=CE(CHAR)
      CALL LABEL(SVEC(I:J),CNAME(CHAR),MSS(CHAR),CMEASURD(CHAR))
   END IF

   IF(CTYPE(CHAR).EQ.'LABL2') THEN
      I=CB(CHAR)
      J=CE(CHAR)
      CALL LABL2(SVEC(I:J),CNAME(CHAR),MSS(CHAR),CMEASURD(CHAR))
   END IF

   IF(CTYPE(CHAR).EQ.'OUTL1') THEN
      I=CB(CHAR)
      J=CE(CHAR)
      CALL OUTL1(SVEC(I:J),CHAR,CNAME(CHAR),MAG(CHAR),MINN(CHAR),
                  TEST(CHAR),MSS(CHAR),CMEASURD(CHAR))
   END IF

   IF(CTYPE(CHAR).EQ.'USER1') THEN
I=CB(CHAR)
J=CE(CHAR)
CALL USER1(SVEC(I:J),CHAR,CNAME(CHAR),MAG(CHAR),MINN(CHAR),
  TEST(CHAR),MSS$(CHAR),CMEASURD(CHAR))
END IF
C
RETURN
END
SUBROUTINE LABEL(OUT$, CNAME, MSS, CMEASURD)

C
C KEYBOARD ENTRY OF 10 SPACE STRING. USES SUBROUTINES
C CLRSCRN, CLEAN$, BEEP
C
C INTEGER L
C CHARACTER EOS*$1, CNAME*15, INS*$30, MSS*$300, OUT*$10
C LOGICAL CMEASURD
C
1000 FORMAT(A)
C EOS$=CHAR(10)
C CALL CLRSCRN
C WRITE(6,*)
C WRITE(6,*) 'ENTER ', CNAME,' USING KEYBOARD.'
C WRITE(6,*) MSS(1:80)
C WRITE(6,*)
C READ(5,1000,ERR=1) INS$  
C CALL CLEAN$(EOS$, INS$, L)
C IF(L.LT.10) THEN
C WRITE(6,*) 'ENTRY TOO LONG. PLEASE TRUNCATE TO 10 SPACES.'
C WRITE(6,*) 'PAUSE 'hit RETURN TO CONTINUE....'
C GOTO 2
C END IF
C OUT$(1:10)=INS$(1:10)
C CMEASURD=.TRUE.
C
C RETURN
C
1 CALL BEEP(5)
C WRITE(6,*) 'ERROR!'
C GOTO 2
C END
SUBROUTINE LABL2(OUT$, CNAME, MS$, CMEASURD)

KEYBOARD ENTRY OF 30 SPACE STRING. USES SUBROUTINES
CLRSCRN, CLEAN$, BEEP

INTEGER L
CHARACTER EOS$*1, CNAME*15, IN$*60, MS$*300, OUT$*30
LOGICAL CMEASURD

1000 FORMAT(A)
EO$=CHAR(10)
CALL CLRSCRN
WRITE(6,*) 'LABL2 V3.2. 7-30-86.'
2 WRITE(6,*) 'ENTER ', CNAME, ' USING KEYBOARD.'
WRITE(6,*) MS$(1:80)
WRITE(6,*) MSS$(81:160)
WRITE(6,*)
READ(5,1000, ERR=1) IN$
CALL CLEAN$(EOS$, IN$, L)
IF(L.GT.30) THEN
   WRITE(6,*) 'ENTRY TOO LONG. PLEASE TRUNCATE TO 30 SPACES.'
   PAUSE 'HIT RETURN TO CONTINUE....'
   GOTO 2
END IF
OUT$(1:30)=IN$(1:30)
CMEASURD=.TRUE.

RETURN

1 CALL BEEP(5)
WRITE(6,*) 'ERROR!'
GOTO 2
END
SUBROUTINE LENG1(OUT$,CHAR,CNAME,MAG,MINN,TESTRNGE,MSS$,CMEASURD)

LENGTH MEASUREMENT IN 2 OR 3 DIMENSIONS. THIS ROUTINE INPUTS 2 DATA
POINTS, COMPUTES THE DISTANCE BETWEEN THEM, CHECKS THE VALUE TO MAk
SURE THAT IT IS NOT ABnormally SMALL OR LARGE, AND THEN LOOPS TO
TAKE ANOTHER MEASUREMENT. WHEN THE MEASUREMENT LOOP IS TERMINATE
THE VALUES ARE CHECKED FOR EXCESSIVE VARIABILITY AND FOR THE NUMBER
OF MEASUREMENTS OBTAINED. THE VALUES ARE THEN AVERAGED, THE MEAN
WRITTEN TO OUT$ (IN G10.X FORMAT), AND THE ROUTINE ENDED.
THE PROTOCOL FILE SHOULD CONTAIN THE FOLLOWING (IN ADDITION TO MINN,
TESTRNGE, MAG, ETC): MSS$(240) DETERMINES
WHETHER 2-D OR 3-D DATA IS TO BE COLLECTED. MSS$(220:239)
CONTAINS 2 F10.X VALUES FOR THE MINIMUM AND MAXIMUM ACCEPTABLE
RANGE THAT A MEASUREMENT MAY TAKE WITHOUT BEING FLAGGED.

USES SUBROUTINES BEEP, CLRSCRN, INPUT; FUNCTION DISTANCE.

INTEGER
  !
  CHAR,
  !
  MINN,
  !
N
REAL
  !
  D,
  !
  DSUM,
  !
  MAX,
  !
  MAXTEST,
  !
  MIN,
  !
  MINTEST,
  !
  P1(3),
  !
  P2(3),
  !
TESTRNGE
CHARACTER
  !
  ANSWER*1,
  !
  CNAME*15,
  !
  DIM*1,
  !
  MAG*5,
  !
  MSS*300,
  !
  OUTS*10,
  !
  STATUS*1
LOGICAL CMEASURD

INITIALIZE

1000 FORMAT(A)
5  MAX=-1000000.0
MIN=1000000.0
READ(MSS$(220:229),1001) MINTEST
READ(MSS$(230:239),1001) MAXTEST
1001 FORMAT(F10.3)
DSUM=0.0
N=0
DIM=MSS$(240:240)
CALL CLRSCRN
WRITE(6,*) 'LENGTH MEASUREMENT OF CHARACTER ',CHAR,' ',CNAME,','
WRITE(6,*), 'MAGNIFICATION SHOULD BE SET TO ',MAG,'.'
WRITE(6,*), 'LENG1 VERSION 1.1 9-26-86 BY DBL.'
WRITE(6,*)
WRITE(6,*)

INPUT DATA, CALC D's, MIN,MX

1 WRITE(6,*), MSS$(1:79)
   WRITE(6,*), 'NUMBER MEASURED SO FAR:',N
   CALL INPUT(P1,MAG,STATUS,DIM)
   CALL BEEP(1)
   IF(STATUS.EQ.'D') GOTO 1
   IF(STATUS.EQ.'S') GOTO 2
   IF(STATUS.EQ.'E') GOTO 3
   WRITE(6,*)

7 WRITE(6,*), MSS$(81:159)
   CALL INPUT(P2,MAG,STATUS,DIM)
   CALL BEEP(2)
   IF(STATUS.EQ.'D') GOTO 7
   IF(STATUS.EQ.'E') GOTO 3

D=DISTANCE(P1,P2,DIM)

IS THIS VALUE ABNORMAL?

IF(D,LT.MINTEST.OR.D,GT.MAXTEST) THEN
   CALL BEEP(3)
   WRITE(6,*), 'ABNORMAL DATA VALUE:,D,'.
   WRITE(6,*), 'EXPECTED RANGE IS:,MINTEST,' TO',MAXTEST,'.
   WRITE(6,*), 'OK TO USE? (Y/N)?'
   READ(5,1000) ANSWER
   IF(ANSWER.EQ.'N'.OR. ANSWER.EQ.'n') GOTO 1
END IF

N=N+1
   IF(D,GT.MAX) MAX=D
   IF(D,LT.MIN) MIN=D
   DSUM=DSUM+D
   IF(STATUS.EQ.'S') GOTO 2
   GOTO 1

END OF MEASUREMENT LOOP

2 CONTINUE

CHECK DATA FOR VALIDITY

1: IS N BIG ENOUGH?

IF(N,LT.MINN) THEN
   CALL BEEP(3)
   WRITE(6,*), 'ONLY ',N,' MEASUREMENTS TAKEN. THIS IS LESS THAN'
   WRITE(6,*), 'THE MINIMUM PERMITTED NUMBER OF MEASUREMENTS ','.
MINN,'"
WRITE(6,*) 'CONTINUE MEASUREMENTS? YES/NO/ABORT (Y/N/A)?'  
READ(5,1000) ANSWER  
IF(ANSWER.EQ.'Y'.OR.ANSWER.EQ.'y') GOTO 1  
IF(ANSWER.EQ.'N'.OR.ANSWER.EQ.'n') GOTO 4  
C  ANSWER WAS ABORT OR ANSWER WAS INVALID. BAIL-OUT...
RETURN  
END IF

4 CONTINUE

2: IF MORE THAN ONE VALUE MEASURED, IS DATA TOO VARIABLE?

IF(N.GT.1.AND.(MAX-MIN).GT.TESTRNGE) THEN
   CALL BEEP(3)
   WRITE(6,*) 'VARIABLE DATA! MIN=',MIN,' MAX=',MAX,' N=',N,'.'
   WRITE(6,*) 'ACCEPTABLE? (Y/N/A[abort]) ->:'
   READ(5,1000) ANSWER  
   IF(ANSWER.EQ.'N'.OR.ANSWER.EQ.'n') GOTO 5  
   IF(ANSWER.NE.'Y'.AND.ANSWER.NE.'y') RETURN  
END IF

VALID DATA. ASSIGN TO OUTPUT VARIABLE AND RETURN.

D=DSUM/N  
WRITE(OUT$,101) D  
101 FORMAT(G10.4)  
CMEASURD=.TRUE.  
RETURN

ERROR HANDLING

3 CALL BEEP(5)
   WRITE(6,*) 'ERROR!! LENGTH MEASUREMENT TERMINATED!'  
   PAUSE 'HIT ANY KEY TO RETURN TO MENU.'
   RETURN

END
SUBROUTINE LENG2(OUT$, CHAR, CNAME, MAG, MINN, TESTRNGE, MS$, CMEASURD)
C
LENGTH MEASUREMENT IN 2 OR 3 DIMENSIONS. DIMENSION SET BY INPUT
C ROUTINE VIA CHARACTER VARIABLE 'DIM'. DISTANCE FUNCTION WORKS
C WITH EITHER 2D OR 3D DATA.
C LENG2 RETURNS 2 OR 3 NUMBERS IN OUTPUT STRING. SEVERAL OPTIONS
C ARE AVAILABLE. CHOICE IS CONTROLLED BY 2-CHARACTER SUBSTRING
C IN POSITIONS 161 AND 162 OF MS$:
C 'SD' - MEAN IN OUT$(1:10) AND STD DEVIATION IN OUT$(11:20)
C 'RN' - MEAN IN OUT$(1:10) AND RANGE (MIN-MAX) IN OUT$(11:20)
C 'HL' - MIN IN OUT$(1:10) AND MAX IN OUT$(11:20).
C 'EV' - MEAN IN OUT$(1:8), STD DEV IN 10:16, AND N IN 17:20.
C OTHER OR BLANK - SAME AS 'EV'.
C POSITIONS 220:229 AND 230:239 OF MS$ CONTAIN MINTEST AND MAXTEST.
C THESE TWO VALUES ARE USED TO CHECK EACH MEASUREMENT AS IT IS
C MADE. IF MEASUREMENT IS BEYOND RANGE, IT IS FLAGGED, AND THE
C VALUE IGNORED UNLESS THE USER SPECIFIES THAT IT IS OK.
C POSITION 165 IN MS$ IS EITHER A '2' OR A '3', AND DETERMINES
C WHETHER 2D OR 3D DATA IS TO BE INPUT.
C
C USES SUBROUTINES BEEP, CLRSCRN, INPUT; FUNCTION DISTANCE.
C
INTEGER
! CHAR,
! MINN,
! N
REAL
! D,
! DSQUARES,
! DSUM,
! MAX,
! MAXTEST,
! MIN,
! MINTEST,
! P1(3),
! P2(3),
! SD,
! T,
! TESTRNGE
CHARACTER
! ANSWER*1,
! CNAME*15,
! DIM*1,
! MAG*5,
! MS$*300,
! OUT$*20,
! STATUS*1
LOGICAL CMEASURD

C 1000 FORMAT(A)
  S MAX=-1000000.0
  MIN=1000000.0
  DSUM=0.0
  DSQUARES=0.0
N=0
READ(MSS(220:229),1001) MINTEST
READ(MSS(230:239),1001) MAXTEST
1001 FORMAT(F10.3)
DIM=MSS(165:165)
CALL CLRSCRN
WRITE(6,*) 'LENGTH MEASUREMENT OF CHARACTER ',CNAME,'
WRITE(6,*) 'MAGNIFICATION SHOULD BE SET TO ',MAG,'
WRITE(6,*) 'LENG2 VERSION 1.1; 9-29-86 BY DBL.'
WRITE(6,*)

INPUT DATA, CALC D's, MIN,MAX

1 WRITE(6,*) MSS(1:79)
  WRITE(6,*) 'NUMBER ALREADY MEASURED: ',N
  CALL INPUT(P1,MAG,STATUS,DIM)
  CALL BEEP(1)
  IF(STATUS.EQ.'D') GOTO 1
  IF(STATUS.EQ.'S') GOTO 2
  IF(STATUS.EQ.'E') GOTO 3
  WRITE(6,*)
7 WRITE(6,*) MSS(81:159)
  CALL INPUT(P2,MAG,STATUS,DIM)
  CALL BEEP(2)
  IF(STATUS.EQ.'D') GOTO 7
  IF(STATUS.EQ.'E') GOTO 3

D=DISTANCE(P1,P2,DIM)

IS THIS VALUE ABNORMAL?

IF(D.LT.MINTEST.OR.D.GT.MAXTEST) THEN
  CALL BEEP(3)
  WRITE(6,*) 'ABNORMAL DATA VALUE: ',D,'
  WRITE(6,*) 'EXPECTED RANGE IS: ',MINTEST,' TO ',MAXTEST,'
  WRITE(6,*) 'OK TO USE? (Y/N)?
  READ(5,1000) ANSWER
  IF(ANSWER.EQ.'N'.OR. ANSWER.EQ.'n') GOTO 1
END IF

N=N+1
IF(D.GT.MAX) MAX=D.
IF(D.LT.MIN) MIN=D
DSUM=DSUM+D
DSQUARES=DSQUARES+D*D
IF(STATUS.EQ.'S') GOTO 2
GOTO 1

END OF MEASUREMENT LOOP

2 CONTINUE

CHECK DATA FOR VALIDITY
IF(N.LT.MINN) THEN
CALL BEEP(3)
WRITE(6,*)ONLY 'N,' MEASUREMENTS TAKEN. THIS IS LESS THAN'
WRITE(6,*) 'THE MINIMUM PERMITTED NUMBER OF MEASUREMENTS ','
MINN,''
WRITE(6,*) 'CONTINUE MEASUREMENTS (YES/NO/ABORT) (Y/N/A)? '
READ(5,1000) ANSWER
IF(ANSWER.EQ.'Y'.OR.ANSWER.EQ.'y') GOTO 1
IF(ANSWER.EQ.'N'.OR.ANSWER.EQ.'n') GOTO 4
C
*ANSWER WAS ABORT OR ANSWER WAS INVALID. BAIL-OUT!...
RETURN
END IF
C
CONTINUE
C
IF(N.GT.1.AND.(MAX-MIN).GT.TESTRNGE) THEN
CALL BEEP(3)
WRITE(6,*) 'VARIABLE DATA! MIN=',MIN,' MAX=',MAX,' N=',N,''
WRITE(6,*) 'ACCEPTABLE? (Y/N/A[abort]) ->:
READ(5,1000) ANSWER
IF(ANSWER.EQ.'N'.OR.ANSWER.EQ.'n') GOTO 5
IF(ANSWER.NE.'Y'.AND.ANSWER.NE.'y') RETURN
END IF
C
VALID DATA. ASSIGN VARIABLES TO OUTPUT STRING AND RETURN.
C
D=DSUM/N
IF(N.GT.1) THEN
T=DSQUARES-(DSUM*DSUM)/N
SD=SQR(T/(N-1))
ELSE
SD=0,0
END IF
IF(MSS(161:162).EQ.'SD'.AND.N.EQ.1) THEN
WRITE(OUT$,101) D
C
(THE STD DEV FIELD IS LEFT BLANK IF N=1)...
ELSE IF(MSS(161:162).EQ.'SD'.AND.N.GT.1) THEN
WRITE(OUT$,103) D,SD
ELSE IF(MSS(161:162).EQ.'RN'.AND.N.GT.1) THEN
WRITE(OUT$,103) D,MAX-MIN
ELSE IF(MSS(161:162).EQ.'RN'.AND.N.EQ.1) THEN
WRITE(OUT$,101) D
C
(RANGE FIELD IS LEFT BLANK IF N=1)...
ELSE IF(MSS(161:162).EQ.'HL'.AND.N.GT.1) THEN
WRITE(OUT$,103) MAX-MIN
ELSE IF(MSS(161:162).EQ.'HL'.AND.N.EQ.1) THEN
WRITE(OUT$,101) D
C
(ONLY FIRST FIELD USED FOR VALUE IF N=1)...
ELSE
WRITE(OUT$(1:20),102) D,SD,N
END IF
102 FORMAT(F8.3,F7.3,I3)
101 FORMAT(F10.4)
103 FORMAT(2F10.4)
    CMEASURD=.TRUE.
    RETURN
C
C    ERROR HANDLING
C
3 CALL BEEP(5)
    WRITE(6,*) 'ERROR!! -LENGTH MEASUREMENT TERMINATED!'
    PAUSE 'HIT ANY KEY TO RETURN TO MENU.'
    RETURN
C
    END
C OUTLINE DIGITIZATION ROUTINE FOR MEASURE PROGRAM, BY DAVE LAZARUS.
C
C DATA IS WRITTEN TO SVEC SUBSTRING IN EITHER 2D OR 3D FORMAT.
C THE FIRST COLUMN OF THE SUBSTRING CONTAINS A 2 OR A 3 TO INDICATE
C WHICH DATA FORMAT IS USED. MS$(240) CONTAINS A 2 OR A 3 TO SPECIFY
C WHICH TO USE. MS$(235:238) CONTAINS THE FORMAT TO USE FOR DATA OUTPUT,
C SUCH AS F7.2, ETC IN STANDARD FORTRAN FORMAT STATEMENT FORM. MS$(236) M
C CONTAIN A NUMBER BETWEEN 1 AND 9, THE PERMISSIBLE RANGE OF OUTPUT FORM
C THE CONTENTS OF MS$(235:238) ARE WRITTEN INTO COLUMNS 2 TO 6 OF THE
C OUTPUT SUBSTRING. COLUMNS 7 TO 10 OF THE OUTPUT STRING CONTAIN N, THE
C NUMBER OF POINTS DIGITIZED. COLUMNS 11 AND UP CONTAIN DATA IN GROUPS OF
C 2 OR 3 POINTS SEPARATED INTERNALLY BY COMMAS, AND SEPARATED BETWEEN GI
C BY SLASHES.
C THE LENGTH OF THE SUBSTRING IS DIMENSIONED AT 3,000, ENOUGH FOR BETWEEN
C 100 AND 200 POINTS. NOTE THAT THE SVEC SUBSTRING PASSED TO THE ROUTINE
C MUST BE 3K LONG (IE, CE-CB=3000). FOR EASE OF EDITING, IT IS BEST TO
C PLACE THE OUTPUT FROM OUTL1 AT THE END OF EACH SPECIMEN'S SET OF MEASU
C MENTS (IE, AT THE END OF EACH LOGICAL RECORD).
C IF YOUR EDITOR CAN'T HANDLE VERY LONG LINES OF TEXT, IT MAY ALSO
C BE NECESSARY TO WRAP EACH LOGICAL RECORD INTO SMALLER SECTIONS (SEE FIL
C WRITING SUBROUTINE AND INITIALIZATION ROUTINE FOR HOW TO SPECIFY THIS).
C MINN CAN BE USED TO FORCE A MINIMUM NUMBER OF POINTS TO BE DIGITIZED. TH
C PROGRAM COMPUTES AND DISPLAYS THE MAXIMUM NUMBER OF POINTS WHICH C
C DIGITIZED BASED ON THE VALUES OF MS$(236) AND MS$(240). IF MAXN IS EXCEED
C THE ROUTINE ABORTS THE DIGITIZATION AND RETURNS TO THE CALLING PROGRAM
C
C REQUIRES SUBROUTINES INPUT, BEEP, CLRSCRN.
C
SUBROUTINE OUTL1(OUTS,CHAR,CNAME,MAG,MINN,TEST,MS$,CMEASURD)
PARAMETER (LVEC=3000)
C
INTEGER
1 CHAR,
1 DIM,
1 I,
1 MINN,
1 MAXN,
1 N,
1 ONE,
1 OUTLENG,
1 PB,
1 PE,
1 PLENG
REAL
1 P(3),
1 TEST
CHARACTER
1 CNAME*15,
1 DIMCHAR*1,
1 FMT1$*4,
1 FMT2$*30,
1 FMT3$*40,
1 MAG*5,
1 MS$*300,
1  OUTS*(LVEC),
1  STATUS*1
LOGICAL
1  CMEASURD

C
CALL BEEP(1)
N=0
ONE=1
FMT1$=MSS(235:238)
   FMT2$=('""//FMT1$//""""//FMT1$//")
   FMT3$=('""//FMT1$//""""//FMT1$//")
   FMT4$=('""//FMT1$//")
DIMCHAR=MSS(240:240)
READ(DIMCHAR,102) DIM

102  FORMAT(11)
READ(MSS(236:236),102) OUTLENG
IF(DIM.EQ.0.OR.OUTLENG.LT.2) THEN
   CALL BEEP(5)
   WRITE(6,*) 'ERROR! DIMENSION OR OUTPUT FORMAT NOT CORRECT!'
   WRITE(6,*) 'DIM=',DIM,' OUTLENG=',OUTLENG
   WRITE(6,*) 'CHECK PROTOCOL FILE AND FIX!'
   WRITE(6,*) 'HIT ANY KEY TO RETURN TO MAIN PROGRAM....'
   PAUSE
   RETURN
END IF
PLENG=DIM*(OUTLENG+1)
MAXN=INT((LVEC-10)/PLENG)

C
CALL CLRSCRN
WRITE(6,*) 'Outline Digitization of character ','CNAME,','
WRITE(6,*) 'Magnification should be set to ','MAG,''
WRITE(6,*) '--- OUTL1 Version 1.1. 10-10-86 by DBL. ---'
WRITE(6,*)
WRITE(6,*) MSS(1:79)
WRITE(6,*) MSS(81:159)
WRITE(6,*)

C
C MAIN LOOP
C
10  WRITE(6,*)'N Points Measured:','N,' MINN:','MINN,
1   MAXN:','MAXN,''
   CALL BEEP(1)

C GET INPUT FROM MEASUREMENT DEVICES

CALL INPUT(P,MAG,STATUS,DIMCHAR)
IF(STATUS.EQ.'E') THEN
   CALL BEEP(5)
   WRITE(6,*) 'INPUT ERROR! POINT IGNORED!'
   GOTO 10
END IF
IF(STATUS.EQ.'D') THEN
CALL BEEP(3)
N=N-ONE
IF(N.LT.0) N=0
WRITE(6,*) 'PREVIOUS POINT DELETED...'
GOTO 10
END IF

INCREMENT POINT COUNT
N=N+1
IF(N.GT.MAXN) THEN
CALL BEEP(5)
WRITE(6,*) 'MAXIMUM # OF POINTS EXCEEDED!'
WRITE(6,*) 'MEASUREMENT TERMINATED!'
WRITE(6,*) 'HIT ANY KEY TO RETURN TO MAIN PROGRAM...'
PAUSE
RETURN
END IF

WRITE POINT TO OUTPUT STRING
PB=11+(N-ONE)*PLENG
PE=PB+PLENG-ONE
IF(DIM.EQ.2) WRITE(OUT$(PB:PE),FMT2$) P(1),P(2)
IF(DIM.EQ.3) WRITE(OUT$(PB:PE),FMT3$) P(1),P(2),P(3)

IF(STATUS.EQ.'S') THEN
CHECK TO SEE IF MINIMUM NUMBER OF POINTS DIGITIZED
IF(N.LT.MINN) THEN
CALL BEEP(5)
WRITE(6,*) 'ONLY',N,' POINTS. NEED AT LEAST,MINN,'!
WRITE(6,*) 'REDO MEASUREMENT. HIT ANY KEY TO CONTINUE...'
PAUSE
DO 11 I=1,LVEC
   OUTS(I:1)=''
11 CONTINUE
RETURN
END IF

WRITE NUMBER OF POINTS DIGITIZED ETC AT BEGINNING OF STRING
OUTS(1:1)=DIMCHAR
WRITE(OUTS(3:5),103) N
FORMAT(13)
OUTS(6:9)=FMT1$

EXIT FROM ROUTINE
CMESURD=.TRUE.
RETURN
END IF
CONTINUE MAIN LOOP

GOTO 10

END OF ROUTINE

END

SUBROUTINE USER1(OUT$,CHAR,CNAME,MAG,MINN,TEST,MSS$,CMEASURD)

DUMMY MEASUREMENT SUBROUTINE. TO BE FILLED IN BY USER AS DESIRED.

INTEGER CHAR,MINN
REAL TEST
CHARACTER OUT$(*),CNAME*15,MAG*5,MS$*300
LOGICAL CMEASURD
CMEASURD=.TRUE.
WRITE(6,*) 'THIS IS USER1 -- HI THERE!'  
PAUSE 'HIT ANY KEY TO CONTINUE...'  
RETURN
END
SUBROUTINE CRTPRINT(ROW,COL,TEXT,LTEXT,IFONT)
INTEGER ROW,COL,IFONT
CHARACTER TEXT*(LTEXT), ESC*1, FONT1*3, FONT2*3, FONT3*5
ESC=CHAR(27)
FONT1=[0m'
FONT2=[7m'
FONT3=[5;7m'

IF(IFONT.GT.3) IFONT=1

IF(IFONT.EQ.1) THEN
   WRITE(6,101) ESC,FONT1,ESC,'\',ROW,':',COL,'H',TEXT(1:LTEXT)
ELSE IF(IFONT.EQ.2) THEN
   WRITE(6,101) ESC,FONT2,ESC,'\',ROW,':',COL,'H',TEXT(1:LTEXT)
END IF


102 FORMAT(1X,A,A5,A,A,I2.2,A,I2.2,A,(A))

103 FORMAT(1X,A,A3,A,A3)
RETURN
END
SUBROUTINE BEEP(N)

WRITE 'N' CONTROL-G'S TO CREATE SOUND.
ALSO MOVES CURSOR BACK TO ORIGINAL LINE.

DO 1 I=1,N
   WRITE(6,101) CHAR(7),CHAR(11)
1 CONTINUE
RETURN
END

SUBROUTINE CLEAN$(XS,INSTRINGS,L)
INTEGER LENGTH,I,L
CHARACTER INSTRINGS$(*),XS$1
LENGTH=LEN(INSTRINGS$)

DO 1 I=LENGTH,1,-1
   IF(INSTRINGS$(I:I).NE. ' ') THEN
      L=I
      RETURN
   END IF
1 CONTINUE
L=LENGTH
RETURN
END

SUBROUTINE CLRSCRN

CLEAR SCREEN AND HOMES CURSOR ON NEC APC

WRITE(6,101) CHAR(26),CHAR(30)
101 FORMAT(A2)
RETURN
END
FUNCTION DISTANCE(P1,P2,DIM)

CALCULATES EUCLIDEAN DISTANCE IN 2-SPACE OR 3-SPACE.

INTEGER X,Y,Z
REAL P1(3),P2(3)
CHARACTER DIM*1
X=1
Y=2
Z=3

IF(DIM.NE.'2'.AND.DIM.NE.'3') THEN
    CALL BEEP(5)
    WRITE(6,*) 'INVALID DIMENSION IN DISTANCE FUNCTION: ',DIM
    PAUSE 'HIT ANY KEY TO RESUME....'
    DIM='E'
    RETURN
END IF

IF(DIM.EQ.'2') THEN
    DISTANCE=SQRT((P1(X)-P2(X)**2+(P1(Y)-P2(Y)**2)
ELSE
    DISTANCE=SQRT((P1(X)-P2(X)**2+(P1(Y)-P2(Y)**2+
                    (P1(Z)-P2(Z))**2)
END IF

RETURN
END
SUBROUTINE INPUT(P,MAG,STATUS,DIM)
C
C THIS VERSION DOES BOTH 2D AND 3D INPUT, DEPENDING ON VALUE
C OF DIM. WHEN GREEN BUTTON ON CURSOR PRESSED, A 'D' IS
C SENT BACK TO CALLING PROGRAM IN STATUS. THIS ALLOWS TABLET
C CURSOR TO BE USED TO DELETE POINTS WITHOUT HAVING TO REMOVE
C HAND FROM CURSOR AND USE COMPUTER KEYBOARD. IN GENERAL, THIS
C PROGRAM INPUTS DATA FROM HIPAD AND LASICO ON NEC APC AND SCALES
C INPUT TO MICROMETERS USING SCALE FACTORS. THIS VERSION
C CREATED: 9-26-86. ALL CALIBRATIONS ARE FOR ZEISS SCOPE.
C CALIBRATED FOR X AND Y WITH ~30 MEASUREMENTS EACH ON STAGE
C MICROMETER AT 40X AND 100X. STD DEV OF DATA ~1 %.
C CALIBRATED FOR Z USING 3-5 MEASUREMENTS ON EACH OF 5 COLLO-
C SPHAERIDS. STD DEV OF Z:XY RATIOS IS 1.3 %.
C EXTERNAL HIPAD,LASICO
C REAL P(3),XYSCALE,ZSCALE
C CHARACTER DIM*1,STATUS*1,MAG*5,XY$*15,Z$*8
C
C IF(MAG.EQ.'40X ') XYSCALE=0.03254
C IF(MAG.EQ.'60X ') XYSCALE=1.0
C IF(MAG.EQ.'100X') XYSCALE=0.01332
C ZSCALE=0.17465
C STATUS='E'
C 101 FORMAT(F6.0)
C 102 FORMAT(F7.0)
C
C CALL HIPAD(XY$)
C IF(DIM.EQ.'3') THEN
C CALL LASICO(Z$)
C READ(Z$(1:7),102) P(3)
C P(3)=P(3)*ZSCALE
C END IF
C READ(XYS(2:7),101) P(1)
C READ(XYS(8:13),101) P(2)
C P(1)=P(1)*XYSCALE
C P(2)=P(2)*XYSCALE
C IF(XYS(1:1).EQ.'2') STATUS='G'
C IF(XYS(1:1).EQ.'B') STATUS='S'
C IF(XYS(1:1).EQ.'R') STATUS='D'
C
C RETURN
C END
; NAME: HIPAD.A86
; DATE: 03 DECEMBER 1984
; AUTH: K. PRADA

; REVISION HISTORY:
; THIS SUBROUTINE INPUTS A STRING FROM THE HIPAD DIGITIZER.
; THE BAUD RATE IS ASSUMED TO BE 4800 AND IS SET BY THIS ROUTINE
; ON FIRST ENTRY.
; THE ROUTINE IS CALLABLE FROM DIGITAL RESEARCH FORTRAN-77 AS:
; CALL HIPAD(ARRAY)
; WHERE ARRAY DEFINES A BYTE ARRAY OF MINIMUM LENGTH 15.
; THE ROUTINE RETURNS THE HIPAD STRING SHOWN ON PAGE 3-6
; OF THE HIPAD OPERATION MANUAL.
; THIS VERSION SUPPORTS THE LARGE MEMORY MODEL ONLY

NAME 'HIPAD'

CSEG
PUBLIC HIPAD

HIPAD:    PUSH BP   ;SAVE BP
MOV BP,SP   ;STACK POINTER TO BP
LES BX,6[BP]   ;GET ARRAY ADDRESS

PUSH DI
PUSH SI

CMP CS:FIRST,0   ;CHECK FIRST FLAG
JNE INPUT      ;DO INPUT
MOV DH,6       ;SET 4800 BAUD
MOV DL,0CEH    ;SET MODE
MOV CL,10
PUSH ES
INT 220        ;NEC INTERRUPT
POP ES
MOV CS:FIRST,0FFH   ;RESET FIRST FLAG

INPUT:      MOV CL,3
PUSH ES
PUSH BX
INT 224      ;GET A BYTE FROM AXI:
POP BX
POP ES

AND AL,7FH   ;STRIP MSB
MOV ES:[BX],AL ;SAVE THE BYTE
INC BX        ;NEXT ADDRESS
CMP AL, 10 ; LOOK FOR LINE FEED
JNE INPUT ; LOOP TILL GET ONE
POP SI ; RESTORE REGISTERS
POP DI
POP BP
RETF

FIRST DB 0

END
; NAME: LASICO.A86
; DATE: 13 DECEMBER 1984
; AUTH: K. PRADA

; REVISION HISTORY:
; 14 FEB 85 - REVISE FOR RTS/DTR ENABLE DISABLE TO TRIGGER SAMPLE
; THIS SUBROUTINE INPUTS A STRING FROM THE LASICO POSITION
; ENCODER
; THE BAUD RATE IS ASSUMED TO BE 4800 AND IS SET BY THIS ROUTINE
; ON FIRST ENTRY.
; THE ROUTINE IS CALLABLE FROM DIGITAL RESEARCH FORTRAN-77 AS:
; CALL LASICO(ARRAY)
; WHERE ARRAY DEFINES A BYTE ARRAY OF LENGTH 8.
; THE ROUTINE RETURNS THE LASICO 6 DIGIT PLUS SIGN AND <CR>
; STRING
; THIS VERSION SUPPORTS THE LARGE MEMORY MODEL ONLY

NAME 'LASICO'
CSEG
PUBLIC LASICO
LASICO:   PUSH BP   ;SAVE BP
           MOV BP,SP   ;STACK POINTER TO BP
           LES BX,6[BP]  ;GET ARRAY ADDRESS
           PUSH DI
           PUSH SI
           CMP CS:FIRST,0 ;CHECK FIRST FLAG
           JNE START    ;DO INPUT
           MOV DH,6    ;SET 4800 BAUD
           MOV DL,0CEH ;SET MODE
           MOV CL,15
           MOV CH,0
           PUSH ES
           PUSH BX
           INT 220    ;NEC INTERRUPT
           POP BX
           POP ES
           START:    MOV CL,15 ;SET COMMAND BYTE
                      MOV CH,1
                      MOV DL,37H
                      PUSH BX
                      PUSH ES
                      INT 220 ;NEC INTERRUPT
POP ES
POP BX
MOV CS:FIRST,0FFH ;RESET FIRST FLAG

INPUT: PUSH ES
       PUSH BX

STAT: MOV CL,15 ;GET UART STATUS
       MOV CH,2
       INT 220
       AND AL,2
       JZ STAT ;LOOP TIL CHAR THERE
       MOV CL,15 ;GET THE CHARACTER
       MOV CH,3
       INT 220
       POP BX
       POP ES

       AND AL,7FH ;STRIP MSB
       MOV ES:[BX],AL ;SAVE THE BYTE
       INC BX ;NEXT ADDRESS
       CMP AL,13 ;LOOK FOR CARRAIGE RETURN
       JNE INPUT ;LOOP TILL GET ONE
       MOV CL,15 ;TURN OFF RTS AND DTR
       MOV CH,1
       MOV DL,15H
       INT 220
       POP SI ;RESTORE REGISTERS
       POP DI
       POP BP
       RETF

FIRST DB 0

END
3-D HARDWARE TEST PROGRAM. VERSION 3, DAVE LAZARUS. 1-15-85.
EXTERNAL HIPAD
EXTERNAL LASICO
   CHARACTER XYDATA(1000)*15, ZDATA(1000)*8, ANSWER*1

101 FORMAT(A)
 6 WRITE(6,*) '3-D DIGITIZER HARDWARE TEST PROGRAM.'
   WRITE(6,*) 'HOW MANY ITERATIONS?'
   READ(5,*) ITS
   WRITE(6,*) 'L-ASICO, H-IPAD, OR B-OTH?'
   READ(5,101) ANSWER
   IF(ANSWER.EQ.'B') THEN
     DO 1 I=1,ITS
       CALL HIPAD(XYDATA(I))
       CALL LASICO(ZDATA(I))
   1 CONTINUE
   ELSE IF(ANSWER.EQ.'L') THEN
     DO 3 I=1,ITS
       CALL LASICO(ZDATA(I))
   3 CONTINUE
   ELSE
     DO 4 I=1,ITS
       CALL HIPAD(XYDATA(I))
   4 CONTINUE
   END IF
   DO 2 I=1,ITS
     WRITE(6,*) I, ' HIPAD',XYDATA(I), ' LASICO',ZDATA(I)
   2 CONTINUE
   WRITE(6,*) 'AGAIN (Y/N)'
   READ(5,101) ANSWER
   IF(ANSWER.EQ.'Y' .OR. ANSWER.EQ.'Y') GOTO 6
STOP
END
Institute of Marine Sciences Library
University of Alaska
O'Neill Building
905 Koyukuk Ave., North
Fairbanks, AK

Attn: Stella Sanchez-Wade
Documents Section
Scripps Institution of Oceanography
Library, Mail Code C-075C
La Jolla, CA  92038

Hancock Library of Biology & Oceanography
Alan Hancock Laboratory
University of Southern California
University Park
Los Angeles, CA  90089-0371

Gifts & Exchanges
Library
Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, NS, B2Y 4A2, CANADA

Office of the International
Ice Patrol
c/o Coast Guard R & D Center
Avery Point
Groton, CT  06340

Library
Physical Oceanographic Laboratory
Nova University
8000 N. Ocean Drive
Dania, FL  33304

NOAA/EDIS Miami Library Center
4301 Rickenbacker Causeway
Miami, FL  33149

Library
Skidaway Institute of Oceanography
P.O. Box 13687
Savannah, GA  31416

Institute of Geophysics
University of Hawaii
Library Room 252
2525 Correa Road
Honolulu, HI  96822

Library
Chesapeake Bay Institute
4800 Atwell Road
Shady Side, MD  20876

MIT Libraries
Serial Journal Room 14E-210
Cambridge, MA  02139

Director, Ralph M. Parsons Laboratory
Room 48-311
MIT
Cambridge, MA  02139

Marine Resources Information Center
Bldg. E38-320
MIT
Cambridge, MA  02139

Library
Lamont-Doherty Geological Observatory
Colombia University
Palisades, NY  10964

Library
Serials Department
Oregon State University
Corvallis, OR  97331

Pell Marine Science Library
University of Rhode Island
Narragansett Bay Campus
Narragansett, RI  02882

Working Collection
Texas A&M University
Dept. of Oceanography
College Station, TX  77843

Library
Virginia Institute of Marine Science
Gloucester Point, VA  23062

Fisheries-Oceanography Library
151 Oceanography Teaching Bldg.
University of Washington
Seattle, WA  98195

Library
R.S.M.A.S.
University of Miami
4600 Rickenbacker Causeway
Miami, FL  33149

Library
Naval Oceanographic Office
NSTL Station
Bay St. Louis, MA  39522
ATTN: Code 4601
MEASURE: A Program for Morphometric Measurements

David Lazarus

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

National Science Foundation

This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept. WHOI-86-40.

'MEASURE', a FORTRAN77 program for collecting morphometric measurements, has been developed. Features of the program include the ability to accept distance and outline measurements in either 2 or 3 dimensions, to collect several different types of measurements on individual specimens, and to keep a running tally of the number of each type of specimen measured. The current version of the program is used for taxonomic and evolutionary studies of radiolarian microfossils and accepts data input from a digitizer tablet and a vertical measurement device attached to a microscope. An ordinary text file is used to configure the program for any desired set of distance or outline measurements. The program is highly modular, and is relatively easy to expand and modify. This report provides a detailed description of the program for both end-users and for programmers wishing to adapt MEASURE to their own uses. A full listing of the source code is included.

1. morphometrics
2. microcomputers
3. microscopy
4. microfossils
5. radiolaria

Approved for publication; distribution unlimited.