NICOLET MED-80® PROGRAMS IN THE
NAVAL BIODYNAMICS LABORATORY EVOKED POTENTIAL SERIES

ROBERT E. TABLER, JR.

JULY 1984

NAVAL BIODYNAMICS LABORATORY
New Orleans, Louisiana

Approved for public release. Distribution unlimited.
This report gives complete documentation and listings for five computer programs developed at the Naval Biodynamics Laboratory to record, analyze, and plot evoked potential data using the Nicolet Med-80® computer. The programs are: EPPROG, PERDAT, EPDATA, PLOTEP, and LSTPER. Appendices include: wiring for the Med-80®, flowcharts for all programs, and printouts of all programs.
NICOLET MED-80® PROGRAMS
IN THE NAVAL BIODYNAMICS LABORATORY
EVOKED POTENTIAL SERIES

ROBERT E. TABLER, JR.
July 1984

Naval Medical Research and Development Command
Research Work Unit No. M0097PN001-5004

Approved by

J. C. GUIGNARD
Chairman, Editorial Review Board

Released by

Captain L. E. WILLIAMS, MC, USN
Commanding Officer

Naval Biodynamics Laboratory
P. O. Box 29407
New Orleans, LA 70189

Opinions or conclusions contained in this report are those of the author and
do not necessarily reflect the views or the endorsement of the Department of
the Navy. Approved for public release; distribution unlimited. Reproduction
in whole or in part is permitted for any purpose of the United States
Government.
SUMMARY

PROBLEM:

Average evoked potentials are used to monitor possible dysfunctions in the central nervous system of rhesus monkey (M. mulatta) or human subjects during impact acceleration studies at the Naval Biodynamics Laboratory. Automated procedures are required to collect evoked potential via the Nicolet Med-80® with minimum interaction by the experimenter.

FINDINGS:

This report gives complete documentation and listings for five computer programs developed at the Naval Biodynamics Laboratory to record, analyze, and plot evoked potential data. The programs are EPPROG, PERDAT, EPDATA, PLOTEP, and LSTPER.

The EPPROG program (Evoked Potential) is a central program used as a pointer to all other programs in the evoked potential test series. A short introduction and some general instructions for each program are provided, if requested.

The PERDAT program (Personal Data) is used to collect personal and montage information for the evoked potential series on both human and rhesus subjects.

The EPDATA program (Evoked Potential Data) is used to automatically collect and store parameter values and evoked potential data.

The PLOTEP program (Plot Evoked Potentials) is used to plot both evoked potential parameter and waveform data, with the corresponding personal data on the Nicolet Zeta Plotter®.

The LSTPER program (List Data) is used to provide a detailed listing of the data contained in the personal data file and the parameter data file.

It should be noted that these programs can also be used in a clinical setting, though some modifications of the personal data portions should be made.

RECOMMENDATIONS:

Any new programs or changes made to the current programs should be incorporated in this manual.

ACKNOWLEDGEMENTS:

Special thanks to Dr. Marc S. Weiss for his technical expertise.

Trade names of materials or products of commercial or non-government organizations are cited where essential for precision in describing research procedures or evaluation of results. Their use does not constitute official endorsement or approval of the use of such commercial hardware or software.

Current address: Robert E. Tabler, Jr., Department of Psychology, University of West Florida, Pensacola, Florida 32504.
# TABLE OF CONTENTS

**PROGRAMS IN THE NAVAL BIODYNAMICS LABORATORY EVOKED POTENTIAL SERIES**

<table>
<thead>
<tr>
<th>Program</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPPROG (pointer program)</td>
<td>1.1 - 1.2</td>
</tr>
<tr>
<td>PERDAT (personal data)</td>
<td>2.1 - 2.5</td>
</tr>
<tr>
<td>EPDATA (evoked potentials collected)</td>
<td>3.1 - 3.6</td>
</tr>
<tr>
<td>PLOTEP (plot evoked potentials)</td>
<td>4.1 - 4.8</td>
</tr>
<tr>
<td>LSTPER (list all parameter and run variables)</td>
<td>5.1 - 5.3</td>
</tr>
</tbody>
</table>

**APPENDIXES**

1. Wiring for the NIC-293 ................. 6.1
2. Flow Charts for All Programs ............ 7.1 - 7.11
3. Printouts of All Programs ............... 8.1 - 8.34
EPPROG

PURPOSE: The EPPROG program (Evoked Potential) is a central program used as a pointer to all other programs in the evoked potential test series. A short introduction and some general instructions for each program are provided, if requested.

10 The variables to be used in the EPPROG program are dimensioned.
10 - 40 The paper is advanced.

50 The user is asked whether or not a listing or a description of the programs being offered is needed. A 'Y' response to the question will cause some general instructions for the execution of all programs to be printed out. A listing of the programs in the evoked potential series and the file names used in calling up the programs are also provided. An 'N' response to the question will cause the program to advance to line 230.

230 The user inputs the name of the program to be executed, and validates the response. Via a conditional branch, the EPPROG program is advanced to the location which contains the information on the requested program. The program requested is then executed. The user can enter 'END' here to exit from the program.

320 These lines contain a brief introduction to each of the programs.
320 - 790 Some of the programs also contain the pre-run information needed. This is followed by a RUN statement for that program.

800 End of program.
VARIABLES USED IN EPPROG

PROGS$ - selected program to be run
X$  - response to a yes or no question
PERDAT

PURPOSE: The PERDAT program (Personal Data) is used to collect personal and montage information for the evoked potential series on both human and rhesus subjects.

10 All variables to be used in this program are dimensioned.

10 - 50

60 Values are assigned to some variables.

60 - 70

80 The following steps are repeated for each variable.

80 - 710

1. P is set equal to the size of the variable under consideration.
2. The variable under consideration is filled with a series of #'s (see SUBROUTINE 1930).
3. The user is now asked to input the value of the variable. If there is no input for this variable then the return key is pressed causing the #'s to be left in place.
4. XX$ is now made equal to the variable under consideration and the user is asked to validate the variable's value.* See possible responses below:
   a). A 'Y' will allow program to continue to the next step.
   b). A '#' will skip to part b of step 6.
   c). Any other response other than 'Y' or '#' will cause the program to return to step 3.
5. The area represented by #'s is cleared (see SUBROUTINE 2060).
6. a). The variable is now placed in its assigned area. The program continues to step 7.
    b). The #'s are left in the area assigned to the variable under consideration.
7. The size of Y$ is increased by the size, P, of the variable under consideration.
8. The program now moves to the next variable and returns to step 1, until all variables have been viewed.

* Unless otherwise noted a 'Y' response to a validation request will allow the continuation of the program's sequential flow; while any other response will cause the repetition of the question.
PERDAT

720 Conditional branch; the keyboard input of INPUT A and INPUT B is directly related to the number of channels to be recorded. The value representing the number of channels obtained here is used in the plotting of the evoked potential data. The user is asked to input: (a) the electrode placement of INPUT A (positive); and (b) the electrode placement of INPUT B (negative). The user starts with the input for the highest numbered channel and ends with the input to channel 1. If there are, for example, only six channels being recorded, the following would happen:

The markers (#) are left in the ranges of channels 7 and 8, then steps 1, 2, 3, 5, 6a, 7, and 8 of the steps followed in lines 80 - 710 are followed.

1470 After all data for the channels has been entered, a listing of the channels and their INPUT A's and INPUT B's is provided and the user is asked to validate all of the variable values. If all values are correct an 'N' response will cause the program to advance to line 1720.

1550 Input the number of the channel that is to be changed and then input the correct INPUT A and INPUT B values. The user is asked to validate the variable's value. The correct value then replaces the old value stored in Y$ (see SUBROUTINE 1970). The program is then returned to line 1470.

1720 The user is asked whether or not data are to be stored. A 'Y' response will be answered with a request for the file's subscript. An 'N' response will cause the program to skip to line 1890. Any other response will cause the question to be repeated. Disk drive #2 is selected, start of DMEM is set to 0, and the size of DMEM is set to 1K. The user is asked to input the file's subscript and validate the variable's value. The file header is made equal to the first digit of the I.D. number. The file base is made equal to the third and fourth digits of the I.D. number. The data contained in Y$ is changed from normal characters into ASCII code and stored on disk.*

* NOTE: IF FILE ALREADY EXISTS, THE ORIGINAL FILE IS DELETED WITH NO WARNING TO THE USER.
The user is given the option to go straight to the EPDATA program. A 'Y' response will cause the EPDATA program to run. An 'N' response will allow the program to return to sequential flow. Any other response will cause the question to be repeated.

The user has the option to return to the pointer program (EPPROG) or to exit the program. A 'Y' response will cause the EPPROG program to run. An 'N' response will cause the program to exit. Any other response will cause the question to be repeated.

Stop execution of the program. The following are subroutines.

End of the program.
SUBROUTINES FOR PERDAT

Subroutine 1930 (1930 - 1960)

PURPOSE: This subroutine places a series of markers (#) in Y$ in the space allocated to the input variable.

The range of the For/Next loop is the size of Y$, represented by the W, to the size of Y$ when increased by the size of the variable under consideration (W + P - 1). Each Data Memory (DMEM) point in the increased size of Y$ is filled with #'.s. The program returns to its sequential flow.

Subroutine 1970 (1970 - 2050)

PURPOSE: This subroutine enables the user to correct any errors made in the input to a channel's INPUT A or INPUT B.

The user inputs the correct INPUT A for the selected channel and is asked to validate the variable's value. The channel's new INPUT A replaces the old value in Y$, in the range specified in lines 1810 - 1880.

The user inputs the correct INPUT B for the selected channel and is asked to validate the variable's value. The channel's new INPUT B replaces the old value in Y$, in the range specified in lines 1810 - 1880. The program is returned to its sequential flow.

Subroutine 2060 (2060 - 2080)

PURPOSE: This subroutine enables the user to check the value of the variable just requested.

The user is asked to validate the value of the variable which has just been entered. A 'Y' is the response for yes. An 'N' is the response for no and will cause the question to be repeated. A return is the response for a variable with no value. The program is returned to its sequential flow.
VARIABLES USED IN PERDAT

A$ - represents individual data points during storage
AIRTP$ - air temperature in room at start of experiment
BASE - file base
BLANKS$ - empty variable used to cleanout '#' which mean no response
CH$ - number of channels to be recorded
COMNT$ - any comments on experiment or subject
DATES$ - date of experiment
INA$ - new value for INPUT A
INA1$ - INPUT A for channel 1
INA2$ - INPUT A for channel 2
INA3$ - INPUT A for channel 3
INA4$ - INPUT A for channel 4
INA5$ - INPUT A for channel 5
INA6$ - INPUT A for channel 6
INA7$ - INPUT A for channel 7
INA8$ - INPUT A for channel 8
INB$ - new value for INPUT B
INB1$ - INPUT B for channel 1
INB2$ - INPUT B for channel 2
INB3$ - INPUT B for channel 3
INB4$ - INPUT B for channel 4
INB5$ - INPUT B for channel 5
INB6$ - INPUT B for channel 6
INB7$ - INPUT B for channel 7
INB8$ - INPUT B for channel 8
GFRCS$ - G-Force used in experiment
HEAD - file header
HEARTS - subject's heart rate at the start of experiment
N1 - start of range for INPUT A
N2 - end of range for INPUT A
N3 - start of range for INPUT B
N4 - end of range for INPUT B
NAMES$ - name of subject
P - size of the variable to be added to Y$
RCH$ - the channel which has a wrong variable
RUNNOS$ - run number for experiment
S - loop control variable
SS - value of range's for listing INPUT A & INPUT B
SUB - file subscript
SUBNO$ - subject number
SUBTP$ - subjects temperature at start of experiment
TIMDA$ - time-of-day for start of experiment
W - place marker for Y$
XS$ - reponse to 'Is variable correct' question is a 'Y' for yes, an 'N' for no, or a return for null (empty) variable
XXS$ - temporary retainer for a variable while doing a visual check on it's value
Y$ - contains all data for storage

2.5
EPDATA

PURPOSE: The EPPDATA program (Evoked Potential Data) is used to automatically collect and store parameter values and evoked potential data.

10 This line zeros 4K of memory (DMEM).

20 Variables to be used in this program are dimensioned and some variables have values assigned.

70 The variable for the subject's identification (I.D.) number is zeroed, and the user inputs the correct value. The user is asked to validate the variable's value. The file base used for the storage of evoked potentials (E.P.) is made equal to the last two numbers of the I.D. number. The file header is made equal to the first digit (letter) of the I.D. number. This letter is either an 'H' for human, or an 'R' for rhesus monkey. The variable for the file subscript is zeroed. The user is asked to input the file subscript's correct value and to validate that value. This is followed by a check to insure that the value is in the range of 0 to 999. If the value is not in this range, the question is repeated. S is made equal to the file subscript and will later be used as a counter for the file subscript. A value of zero is placed in the post stimulus delay time.

210 The following steps are taken:

210 - 460
1. Input the value for a variable.
2. If the value has to be in some range there is a check to insure that the value is in this range.
3. If the variable is non-string, its value is placed in a string variable for storage.
4. The value of the variable under consideration is placed in its proper storage location in Y$ by the counter variable W.
5. P is set equal to the size of the variable under consideration.
6. W is increased by P.

* Unless otherwise noted, a 'Y' response to a validation request will allow the continuation of the program's sequential flow, while any other response will cause the repetition of the question.

** NOTE: IF A FILE ALREADY EXISTS, THE ORIGINAL FILE IS DELETED WITH NO WARNING TO THE USER.

3.1
Each of the following variables are put through the steps (1 - 6) found on page 3.1.

1. Wrist stimulated: WRISTS.
2. Post stimulus delay time: DELAY.
3. Number of stimuli to be presented per second: STSEC.
4. Stimulus period: STPER is calculated by dividing 1 by the number of stimuli to be presented per second. This value is then multiplied by 1000 to be in the same data range as the other values (starts at step 4).
5. Minimum sweep length: SWPSZ (skips all steps). If the sweep length is an improper value an error message is printed, and the user is asked to input a new value for the sweep length (see SUBROUTINE 1510).
6. Size of memory: SZMEM is calculated by dividing the sweep length by .04.
7. Dwell time: DWELL is calculated by dividing the sweep length by the size of memory to be used. The dwell time is now divided by 1000 to be in the same data range as the other values.
8. The dwell time is multiplied by the size of memory to obtain the correct sweep length.
9. Start of memory: STMEM is set to 0 and along with size of memory, dwell time and length of sweep are stored in memory (starts at step 4).

The user is given a chance to find the stimulus threshold level for the subject. When the subject's threshold has been found a 'Q' is typed to turn off the stimulus (see SUBROUTINE 1620).

It is now requested that the user input values for the following variables (see steps 1 - 6 found on page 3.1):

1. Subject's threshold: THRES$.
2. Stimulus intensity: STINT$.
3. Amplifier/filter's sensitivity: SENTY.
4. Amplifier/filter's low bandpass: LBAND.
5. Amplifier/filter's high bandpass: HBAND.
6. Number of sweeps to be averaged: SWPS.
7. Number of channels to be collected: CHS.
The user is offered a listing of all the variables to be used in the experiment. A 'Y' response will provide a list of all variables with an identification number located on the left hand side. An 'N' response will cause the program to skip to line 850. Any other response will cause the question to be repeated.

After looking at the variable list, the user is asked if the values are correct. A 'Y' response will cause the program to skip to line 1200 of the program. An 'N' response will cause the user to be asked which variable is to be changed. Any other response will cause the question to be repeated. The variable to be changed is identified by the numerical value found on the left hand side. The following variables can not be changed as they are not under user control:

1. dwell time
2. stimulus period
3. start of memory
4. size of memory

It should be noted that changing the following variables' values affects other parameter values:

1. stimulus per second - changes stimulus period.
2. sweep length - changes size of memory and dwell time.

Any changes made on the list of variables will also be made on the file of these parameters (see SUBROUTINE 1750). This section is repeated until there is a 'Y' response to the "are variables correct" question.

Disk drive #2 is selected for storage of files. The letter P (for parameter), represented by the value 208, is placed in the file header. The file's base and subscript are entered to form the file name. The data file is then translated into ASCII code and stored on disk.
1250 The size of memory to be used is zeroed. The stimulus is turned on (see SUBROUTINE 1620). To begin averaging waveforms, the user hits any key. Once the averaging is completed the baseline is zeroed and the stimulus is turned off. The user is asked whether or not the waveforms collected are to be stored on disk. A 'Y' response will cause a D (for data), represented by the value 196, to appear in the file header. The file base and file subscript remain the same as for the parameter file. The data file is then stored on disk #2. The subscript number is now incremented by one, in case the user wants to rerun the program. An 'N' response will cause the program to skip to line 1330, any other response will cause the question to be repeated.

1330 The user is asked whether or not a plot of the evoked potential data is wanted. A 'Y' will cause a plot of all channels to appear on the Hewlett Packard X-Y plotter. An 'N' response will cause the program to skip to line 1440. Any other response will cause the question to be repeated. The user is now asked whether or not there is to be a second plot of the data enabling a change in the vertical display scale (VDS). A 'Y' response will cause a plot to be produced (make any VDS changes before typing 'Y'). An 'N' response will cause the program to go to line 1410, any other response will cause the question to be repeated.

1410 The user is asked if the program is to be rerun without any changes in the parameter values. An 'N' response will cause the program to skip to line 1470. A 'Y' response will allow the continuation of the program's sequential flow. Any other response will cause the question to be repeated. The user is asked if there are any parameter values to be changed. A 'Y' response here will make the program go to line 680; an 'N' response will make the program go to line 1250, any other response will cause the question to be repeated.

1470 The user is asked whether or not the pointer program EPPROG is to be run. A 'Y' response will execute the EPPROG; an 'N' response will make the program go to line 1830, any other response will cause the question to be repeated.

1500 Stop execution of the program. The following are subroutines.

1790 End of program.

3.4
SUBROUTINES FOR EPDATA

SUBROUTINE 1510 (1510 - 1610)

PURPOSE: This subroutine is used to calculate the amount of memory to be used, the dwell time, and the length of the sweep to be made in averaging evoked potentials.

To determine the size of memory, the length of the sweep is divided by .04. The size of memory is then set equal to the next largest power of 2. To determine the dwell time, the length of sweep is divided by the size of memory and the remainder is then multiplied by 1000. The dwell time cannot be less than 40 μsec., therefore if the value of the dwell time is less than 40 the time is set equal to 40 μsec. The dwell time is now divided by 1000 to place it in the same millisecond range as the other time parameter value. This new dwell time is multiplied by the size of memory to find the correct sweep length.

SUBROUTINE 1620 (1620 - 1740)

PURPOSE: This subroutine is used to turn on the stimulus, with parameters pre-set in the NIC-293 I/O Controller Unit.

The stimulus is turned on using the following parameters: stimulus period, stimulus period x 1000, and the delay time. A negative trigger is selected and the user is able to view the ongoing evoked potential signal being received. This subroutine is also used to start the stimulus for finding the subject's threshold and for the collection of averaged evoked potentials (see Appendage 1).

SUBROUTINE 1750 (1750 - 1790)

PURPOSE: This subroutine enables the user to make changes in the values of any variables to be used in the experiment.

The user is asked to input the value of the parameter variable to be changed and then validates the response. The new parameter value is placed in its correct position in the parameter file.
VARIABLES USED IN EPDATA

A$ - represents individual data points from Y$ during storage
CHS - number of channels being recorded
CHS$ - number of channels being recorded
DELAY - delay time before recording
DELAY$ - delay time before recording
DWELL - dwell time
DWELL$ - dwell time
HBAND - high bandpass
HBAND$ - high bandpass
LBAND - low bandpass
LBAND$ - low bandpass
LLEWD - dwell time divided by 1000
N1 - start of range for the variable being changed
N2 - end of range for the variable being changed
NO - the number of the variable being changed
REPTS - stimulus period multiplied by 1000
REPTS$ - stimulus period multiplied by 1000
S - loop control variable
SUB - file subscript
SSUB - file subscript
SENTY - amplifier sensitivity
SENTY$ - amplifier sensitivity
STINTS - stimulus intensity
STMEM$ - start of memory
STPER - stimulus period
STPER$ - stimulus period
STSEC - number of stimuli presented per second
STSEC$ - number of stimuli presented per second
SUBNO$ - subject's I.D. number
SWPS - number of sweeps to be averaged
SWPS$ - number of sweeps to be averaged
SWPSZ - length of sweep
SWPSZ$ - length of sweep
SZMEM - size of memory
SZMEM$ - size of memory
THRESS - subject's stimulus threshold
VAL$ - new value for the variable being changed
W - place marker for Y$
WRISTS - wrist being stimulated
X$ - response to a YES or NO question
Y$ - contains all variable data for this program
PLOTEP

PURPOSE: The PLOTEP program (Plot Evoked Potential) is used to plot both evoked potential parameter and waveform data, with the corresponding personal data on the Nicolet Zeta Plotter®.

10 All variables to be used in this program are dimensioned, and values are assigned to some of the variables.

10 - 110 This line sets the starting point of Data Memory (DMEM) to 0 and memory size to 1K. Disk drive #2 which contains the data files is selected.

130 The user is asked to input the subject's identification (I.D.) number, and then validate the response.* The file base is made equal to the last two digits of the I.D. number, while the file header is made equal to the first digit (a letter) of the I.D. number. The user inputs the subscript number, which is to be used in retrieving data files, and validates the response. This is followed by a check to insure that the value is in the correct data range. A conditional branch is used on the value of the file header to see whether the file header represents a human file (read value of 40 or H), or a rhesus file (read value of 50 or R). The correct file header (write values of 200 = H or 210 = R), file base, and file subscripts are stored in DMEM. The selected personal file is read from disk and translated from ASCII code into normal characters. These characters are then stored in VS. A P represented by 208 is put into the file header in order to call up the parameter file. The file is read from disk using the file base and subscript used in calling the personal file. The new file is translated from ASCII code into normal characters, and placed in Y$.

130 - 310 The sweep length, HS, and the number of channels recorded, OS$, are obtained from VS. CH is made equal to the string variable for the number of channels recorded, CH$. The increments to be used on the horizontal time scale for the sweep length are determined by dividing the sweep length by 10.

320 * Unless otherwise noted a 'Y' response to a validation request will allow the continuation of the program's sequential flow, while any other response will cause the repetition of the question.
The user inputs the minimum voltage level, VA, obtained from the experiment and validates the variable's value. VR is made equal to the minimum voltage level times 2 to get the full (+/-) range. This number is divided by ten, giving the size of the increments to be used on the vertical voltage scale. The user inputs the gain which is to be used in the plotting of the evoked potential (E.P.) and validates the variable's value.

The Zeta Plotter's® pen should be placed in the upper right hand corner when facing the plotter. The current pen position of the plotter is selected to be the point of reference (origin). The rotation of the pen is selected as 1.0.*

SUB$ is made equal to the SUB. A variable which represents the filename of the E.P. file to be plotted 'NOFS', is built by having its first digit as the first digit of the I.D. number, its second and third digits as the third and forth digits of the I.D. number, and its fourth, fifth, and sixth digits as the file subscript number.

Via a conditional branch, the program advances to the location which contains the correct format for plotting the number of channels recorded.

* See Zeta Plotter® - SBASIC Commands handout MO290601 for SUP800.
The following are steps to be followed in the plotting of experimental data:

1. Plotting paper is advanced and the point of origin for the Zeta Plotter® is selected.
2. Value is given to the variable which determines the distance that channel information is to be printed from the bottom of the page.
3. The number of channels to be plotted is obtained from the personal file.
4. The montage information for the selected channel is obtained from the parameter file.
5. The channel number and the montage information is plotted in its proper location.
6. The top vertical line for the box containing the bottom E.P. waveform to be plotted is drawn with incremented tick marks.
7. The header information for the vertical scale is drawn at a right angle (2.0 rotation) to the other header information.
8. The rest of the header information is printed at the top of the page. This is repeated until the container and header information for all E.P. channels has been plotted.
9. The value for the size of DMEM, SZ, is obtained from the parameter file. The size of memory for each channel, PLTSZ, is determined (PLTSZ = SZ / CH). The size of DMEM for the E.P. file is input into BASEXC. Disk drive #2 is selected. The value 196, representing 'D' for E.P. data file, is put into the file header. The file base number and the subscript number used to select the personal and parameter files are used to select the E.P. data file. The E.P. data file is then read from disk. The size of memory used in storing each individual channel is placed in DMEM. The present pen location is made the point of origin, and a rotation of 1.0 is selected.
10. A conditional branch for the plotting of E.P. waveforms is dependent on the number of channels which were recorded during the experiment.
11. The graph paper is advanced to the page where the highest numbered channel is to be plotted and the point of origin is found. The E.P. waveform for that channel is read from disk and plotted, followed by the plotting of all other waveforms waveforms in their prospective boxes.
12. After the plotting of channel one's waveform the pen is returned to the point of origin.
The user is asked whether or not another file is to be plotted. If the response is 'Y' then the program continues with its sequential flow. An 'N' response will advance the program to line 1310. Any other response will cause the question to be repeated. The user is asked whether or not the next file to be plotted is in the same series as the previous file. A 'Y' response will advance the program to line 1230. If the response is 'N' the program advances graph paper to a clean page (see SUBROUTINE 2100) and the program is returned to line 120. Any other response will cause the question to be repeated. The user is asked whether or not the next file to be plotted is the next number in the series. A 'Y' response will increase the file subscript by one, advance the graph paper to a clean page (see SUBROUTINE 2100), and return the program to line 420. An 'N' response will advance the program to line 1260, any other response will cause the question to be repeated.

Input the subscript of the new file and validate the value. A 'Y' response will advance the graph paper (see SUBROUTINE 2100) and return the program to line 420, while any other response will cause the question to be repeated.

The user is asked whether or not another program in the E.P. series is to be run. A 'Y' response will initiate the Pointer Program, while an 'N' response will cause the program to exit. Any other response will cause the program to repeat the original question.

Stop execution of the program. The following are subroutines.

End of program.
SUBROUTINES FOR PLOTEP

SUBROUTINE 1340 (1340 - 1380)

PURPOSE: The purpose of this subroutine is to plot a channel's montage information in the correct location on the page.

The number of the channel being plotted is printed. The channel's correct INPUT A montage is pulled off of the parameter file and printed. A dash is then printed to separate the montages. The channel's correct INPUT B montage is pulled off of the parameter file and printed. The program then returns to its sequential flow.

SUBROUTINE 1390 (1390 - 1420)

PURPOSE: The purpose of this subroutine is to draw the header information for the vertical voltage scale at a right angle to all other header information.

The Zeta Plotter® pen is returned to its point of origin. The rotation of the pen direction is changed to 2.0, and the pen is moved into position for plotting. The header information for the vertical voltage scale is printed. The pen is moved to the bottom of the preceding page. This location is made the new point of origin with a rotation of 1.0. The program is returned to its sequential flow.

SUBROUTINE 1430 (1430 - 1460)

PURPOSE: The purpose of this subroutine is to draw the top of the enclosure which contains the E.P. waveform for a file with an odd number of recorded channels. Only the lower box is drawn for the individual channel.

The value of XX is set to -10. The Zeta Plotter® pen is moved into position and the top horizontal line is drawn for the E.P. waveform box. The eleven incremented tick marks on the top horizontal line. All header information and the rest of the box is now printed (see SUBROUTINE 1470).
SUBROUTINES FOR PLOTEP

SUBROUTINE 1470 (1470 - 1920)

PURPOSE: The purpose of this subroutine is to plot the header information and the rest of the bottom enclosure for plotting an E.P. waveform.

All header information is plotted. The bottom horizontal line with incremented tick marks and the time scale is plotted, along with its header information. The left vertical line is plotted with its incremented tick marks on the voltage scale. The tick marks for the center line of the lower box are plotted followed by the plotting of the lower right vertical line with incremented tick marks.

SUBROUTINE 1930 (1930 - 2090)

PURPOSE: The purpose of this subroutine is to plot the enclosures for the upper E.P. enclosure.

The upper left vertical line with the voltage scale is drawn with incremented tick marks. The top horizontal line is plotted with incremented tick marks, as is the right vertical line. The center line for the upper enclosure is now plotted. The program is returned to its sequential flow.

SUBROUTINE 2100 (2100 - 2180)

PURPOSE: The purpose of this subroutine is to advance the graph paper to a clean page for the plotting of another E.P. file.

The paper is advanced via a conditional branch according to the number of channels plotted in the last execution of the program. The program is returned to its sequential flow.
VARIABLES USED IN PLOTEP

A$ - G-FORCE
B$ - DATE:
BASE - file base number
C - distance incrementation for tick marks
CH - number of channels recorded
CH$ - number of channels being plotted
CS - NAME:
C1$ - CH.#1
C2$ - CH.#2
C3$ - CH.#3
C4$ - CH.#4
C5$ - CH.#5
C6$ - CH.#6
C7$ - CH.#7
C8$ - CH.#8
D - distance from bottom of page for plotting channel information
D$ - SUBJECT NUMBER:
E$ - SLED RUN NUMBER:
F$ - SOMATOSENSORY EVOKED POTENTIAL
FNS - FILENAME:
G$ - (IN MSEC.)
GA$ - PLOT GAIN:
GAINS - gain to be used in plotting of E.P. waveforms
HS - (IN UV)
HEAD - file header
HR - incrementation for length of sweep scale
HS - length of sweep
I - loop control variable
IS$ - DELAY TIME:
JS$ - STIMULI/SECOND:
KS$ - DWELL TIME (IN US):
VARIABLES USED IN PLOTEP

LS - SWEEP LENGTH:
M - distance incrementation for tick marks
MS - 
N1 - start of location for 'INPUT A' montage
N2 - end of location for 'INPUT A' montage
N3 - start of location for 'INPUT B' montage
N4 - end of location for 'INPUT B' montage
NOS - NUMBER SWEEPS:
NOFS - name of data file to be plotted
OS - number of channels recorded
P - distance for incrementation of dashes
PP - individual data points from DMEM to Y$
PLTSZ - size of memory used in storage of individual waveforms
SUB - file subscript number
SUBS - file subscript number
SUBNO - subject's I.D. number
SZ - size of memory used in the storage of waveforms
U - ASC data from data memory for Y$
UU - individual data points from DMEM to V$
UHA - distance for plotting length of sweep scales
UVA - distance for plotting voltage level scale
V - ASC data from data memory for Y$
VA - voltage level in E.P. recording
VR - incrementation for the range of the voltage level
VS - parameter file
W - loop control variable
WS - WRIST STIMULATED:
XX - distance from edge of paper for making tick marks
XS - response to a 'Y' or an 'N' question
YS - personal data file
Z - distance from edge of paper for making dashes
P trend: The LSTPER program (List Data) is used to provide a detailed listing of the data contained in the personal data file and the parameter data file.

10 All the variables to be used in this program are dimensioned.

20 The range of data memory (DMEM) is provided with 0 as the start and 1K as the size. Disk drive #2 is selected, as it is here that the data files are stored.

30 The user inputs the subject's identification number (I.D.), and then validates the value. The file header is the first digit of the I.D. number (a letter), and the file base is the third and fourth digit of the I.D. number. The user now inputs a three digit file subscript to be used in calling the correct data files and validates the subscript's value. The value of the subscript is checked to ensure that it is in the correct data range.

150 A letter H (200) for human, or an R (210) for rhesus is placed in the file header 'PFH', the file base is placed in 'PFB', and the file subscript is placed in 'PFS'. After the filename has been selected the data file is read from disk into DMEM. Individual data points are read from DMEM, translated from ASCII code into normal characters, and placed in Y$.

210 Input the letter P (208) for parameter in the file header. The file is read from disk into DMEM. The individual data points are read from DMEM, translated from ASCII code to normal characters, and placed in V$.

260 To continue the program, the user is asked to advance the paper to the top of a new page and type 'Y'. This allows a separation of unwanted instructions from the desired listing of the personal and parameter data values.

290 The variable in question is printed out followed by its assigned value. The program advances to the next variable.
The user is asked whether or not another personal data file is to be listed. A 'Y' response will return the program to line 30. An 'N' response will continue the program in its sequential flow. Any other response will cause the question to be repeated.

The user is asked whether or not another program is to be executed. A 'Y' response will cause the execution of the pointer program (EPPROG). An 'N' response will stop the program. Any other response will cause the question to be repeated.

End of program.
VARIABLES USED IN LSTPER

BASE - file base
HEAD - file header
S - loop control variable
SUB - file subscript
SUBNOS - subject's identification number
X - ASC data from data memory
X$ - response to a yes or no question
Y$ - contains variable values from the personal data file
V$ - contains variable values from the parameter data file
APPENDIX 1

PATCH PANEL CONNECTIONS MADE ON THE NIC-293®
PATCH PANEL CONNECTIONS MADE ON THE NIC-293® *

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>DESTINATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-C2</td>
<td>B-D7</td>
</tr>
<tr>
<td>B-C1</td>
<td>A-A1</td>
</tr>
<tr>
<td>B-C6</td>
<td>B-D5</td>
</tr>
<tr>
<td>B-C4</td>
<td>A-C8</td>
</tr>
<tr>
<td>A-C8</td>
<td>A-J1</td>
</tr>
<tr>
<td>B-C2</td>
<td>A-A3</td>
</tr>
<tr>
<td>A-J2</td>
<td>B-D3</td>
</tr>
<tr>
<td>B-E5</td>
<td>B-D4</td>
</tr>
<tr>
<td>A-G3</td>
<td>B-D6</td>
</tr>
<tr>
<td>B-H3</td>
<td>B-J1</td>
</tr>
<tr>
<td>B-J1</td>
<td>B-J2</td>
</tr>
<tr>
<td>B-J2</td>
<td>B-J3</td>
</tr>
</tbody>
</table>

FUNCTIONS OF PINS **

A-A1 - BNC connector A
A-A3 - BNC connector C
A-C8 - Amphenol terminal V
A-G3 - I/O level output (if the I/O level is low then the timer is enabled; if I/O level is high then the timer is disabled)
A-J1 - One Shot input (0 to 4 volt transition)
A-J2 - One Shot output (100 nanosecond pulse)
B-C1 - (-)1 Buffered timing pulse output
B-C2 - (+)1 Buffered timing pulse output
B-C4 - (-)2 Buffered timing pulse output
B-C6 - (-)3 Buffered timing pulse output
B-D3 - ORed and buffered trigger inputs for trigger A1
B-D4 - ORed and buffered trigger inputs for trigger B1
B-D5 - ORed and buffered trigger inputs for trigger A2
B-D6 - ORed and buffered trigger inputs for trigger B2
B-D7 - ORed and buffered trigger inputs for trigger A3
B-E5 - unassigned decoded I/O command used to start timers
B-H3 - Internal clock frequency (5 MHz)
B-J1 - Timer clock # 1
B-J2 - Timer clock # 2
B-J3 - Timer clock # 3

* See NIC-293® Timer Configuration Handout for Prestimulus Averaging.
** See NIC 1080® Series Description and Instructions Manual for NIC-293 I/O Controller®.
START

DIMENSIONING OF VARIABLES

LIST PROGRAMS OFFERED

Y

OUTPUT DESCRIPTIVE LIST OF PROGRAMS & REFERENCE NAMES

N

INPUT OF DESIRED PROGRAM

WAS END TYPED?

Y

STOP

N

OUTPUT COMPLETE DESCRIPTION OF DESIRED PROGRAM

RUN DESIRED PROGRAM
START

DIMENSION VARIABLES & ASSIGN VALUES

INPUT VARIABLES REAL VALUE
1. SUBJECT I.D. NUMBER
2. DATE OF EXPERIMENT
3. G-FORCE
4. RUN NUMBER
5. TIME-OF-DAY
6. AIR TEMPERATURE
7. SUBJECT'S TEMPERATURE
8. SUBJECT'S HEART RATE
9. COMMENTS ON EXPERIMENT
10. NUMBER OF CHANNELS

ENTRY CORRECT?

Y

STORE VARIABLE INTO Y$

INPUT MONTAGES

N

A

B
ENTRY CORRECT?

Y

STORE VALUES INTO Y$

N

STORE ON DISK?

Y

STORE DATA

N

RUN EPDATA?

Y

RUN EPDATA

N

RUN EPPROG?

Y

RUN EPPROG

STOP
START

DIMENSION VARIABLES & ASSIGN VALUES

INPUT SUBJECT'S I.D. NUMBER AND THE FILE SUBSCRIPT

INPUT VARIABLE VALUES FOR
1. WRIST STIMULATED
2. DELAY TIME
3. STIMULI PER SECOND

STIMULUS PERIOD CALCULATED

INPUT SWEEPLENGTH

SIZE OF DMEM, SWEEP LENGTH, & DWELL TIME CALCULATED
FIND THRESHOLD

INPUT VARIABLE VALUES FOR
1. SUBJECT'S THRESHOLD
2. STIMULUS INTENSITY
3. AMPLIFIER/FILTER SENSITIVITY
4. AMPLIFIER/FILTER LOW BANDPASS
5. AMPLIFIER/FILTER HIGH BANDPASS
6. NUMBER OF SWEEPS COLLECTED
7. NUMBER OF CHANNELS RECORDED

LIST VARIABLES?

LIST VAR. & VALUES

ENTRIES CORRECT?

CHANGE VARIABLES VALUE
NY CHANGE PARAMETER VALUES?  

Y  NO  

CHANGE PARAMETER VALUES?  

Y  NO  

RUN ANOTHER PROGRAM?  

Y  NO  

STOP

RUN EPPROG

RERUN PROGRAM?
START

DIMENSION & ASSIGN VALUES TO VARIABLES

INPUT SUBJECT I.D. NUMBER & SUBSCRIPT NUMBER

READ PERSONAL & PARAMETER DATA FILES

RETRIEVE NUMBER OF CHANNELS & SWEEP LENGTH FROM DATA FILES. DETERMINE INCREMENTED TIME SCALE

INPUT MIN. VOLTAGE & GAIN FOR PLOTTING WAVEFORMS & DETERMINE INCREMENTED VOLTAGE SCALE

SELECT POINT OF ORIGIN ON ZETA PLOTTER FOR PLOTTING
PRINT HEADER INFO. & WAVEFORM ENCLOSURES FOR EACH CHANNEL

PLOT WAVEFORMS IN THEIR ENCLOSURES

PLOT ANOTHER FILE?

IS FILE IN THIS SERIES?

INCREASE FILE SUBSCRIPT BY 1

FILE NEXT IN SEQUENCE?

INPUT NEW FILE SUBSCRIPT

RUN ANOTHER PROGRAM?

RUN EPPROG

STOP

Y

N
START

DIMENSION VARIABLES

INPUT SUBJECT'S I.D. NUMBER & FILE SUBSCRIPT

READ PERSONAL & PARAMETER DATA FILES FROM DISK

WRITE VAR. DESCRIPTION AND VAR. DATA

ANOTHER VARIABLE?

A

N

Y

B
A

LIST ANOTHER FILE?

B

RUN ANOTHER PROGRAM?

STOP

RUN EPPROG

Y

N

N

Y
APPENDIX 3

PRINTOUTS OF EACH PROGRAM
LIST EPPROG

10 DIM PROG$(6), X$(1)
20 PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT
30 PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT
40 PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT
50 PRINT "DO YOU NEED A LISTING OF THE PROGRAMS BEING OFFERED OR"
60 INPUT "DO YOU NEED SOME INFORMATION ON THE PURPOSE OF THE PROGRAMS? - ", X$
70 IF X$ = "Y" THEN 80; IF X$ = "N" THEN 230; GOTO 50
80 PRINT; PRINT "THE PROGRAM DISK SHOULD BE IN DISK DRIVE # 1."
90 PRINT; PRINT "THE DATA DISK SHOULD BE IN DISK DRIVE # 2."
100 PRINT; PRINT "NOTE: IF YOU MAKE A ERROR NO ERROR MESSAGE IS PRINTED."
110 PRINT "THE QUESTION WILL JUST BE REASKED."
120 PRINT; PRINT
130 PRINT "THE FOLLOWING IS A LISTING OF ALL PROGRAMS IN THE EVOKED POTENTIAL"
140 PRINT "SERIES AND THE SYMBOLS TO BE USED WHEN CALLING A PROGRAM."
150 PRINT; PRINT "INPUT OF PERSONAL DATA - PERDAT"
160 PRINT "COLLECTION OF PARAMETER AND EVOKED POTENTIAL DATA - EPDATA"
170 PRINT "PLOTTING OF THE EVOKED POTENTIAL DATA ON THE ZETA PLOTTER - PLOTEP"
180 PRINT "OBTAINING A DETAILED LISTING OF ALL EXPERIMENTAL DATA - LSTPER"
190 PRINT "TRANSFER OF DATA FILES TO THE ECLIPSE - MEDECL"
210 PRINT; PRINT "IF YOU DESIRE TO EXIT THE PROGRAM TYPE 'END' AS A "
220 PRINT "RESPONSE TO THE FOLLOWING QUESTION."
230 PRINT; INPUT "INPUT THE SYMBOL OF THE DESIRED PROGRAM - ", PROG$(1)
240 PRINT "IS ", PROG$(1), " CORRECT?"
250 INPUT X$(1)
260 IF X$(1) = "Y" THEN 270; GOTO 230
270 PRINT
280 IF PROG$(1) = "VARUSE" THEN 320; IF PROG$(1) = "PERDAT" THEN 350
290 IF PROG$(1) = "LSTPER" THEN 560; IF PROG$(1) = "EPDATA" THEN 600
300 IF PROG$(1) = "MEDECL" THEN 700; IF PROG$(1) = "PLOTEP" THEN 730
310 IF PROG$(1) = "END" THEN 800; IF PROG$(1) = PROG$(1) THEN 140
350 PRINT "THIS PROGRAM IS USED TO OBTAIN AND SAVE FILES ON PERSONAL DATA FOR"
360 PRINT "HUMAN AND RHESUS MONKEY SUBJECTS USED IN THE NDDL EVOKED"
370 PRINT "POTENTIAL PROGRAM."
380 PRINT; PRINT "BEFORE RUNNING A SUBJECT THE FOLLOWING INFORMATION"
390 PRINT "SHOULD BE OBTAINED:"
400 PRINT; PRINT "SUBJECT'S NAME"
410 PRINT "SUBJECT'S I.D. NUMBER"
420 PRINT "G-FORCE TO BE USED IN EXPERIMENT"
430 PRINT "RUN NUMBER FOR EXPERIMENT"
440 PRINT "TIME-OF-DAY"
450 PRINT "DATE OF EXPERIMENT"
460 PRINT "ROOM AIR TEMPERATURE (IN F)"
470 PRINT "SUBJECT'S BODY TEMPERATURE (IN F)"
480 PRINT "SUBJECT'S HEART RATE (BEATS PER MIN.)"
490 PRINT "NUMBER OF CHANNELS TO BE RECORDED"
500 PRINT "MONTAGES TO BE USED IN RECORDING"
510 PRINT; PRINT "BE SURE THAT YOU KNOW THE CORRECT FILE SUBSCRIPT ">
520 PRINT; PRINT "ANSWER ALL QUESTIONS WITH Y (YES), N (NO), OR A # (N/A),"
PRINT "UNLESS OTHERWISE REQUESTED. (NOTE) THE N IS ALSO USED IN"  
PRINT "THE 'IS VARIABLE CORRECT' QUESTION."

RUN PERDAT
PRINT "THIS PROGRAM IS USED TO OBTAIN A DETAILED LISTING OF ALL OF THE"  
PRINT "PERSONAL DATA FROM THE PERDAT PROGRAM AND THE PARAMETER"  
PRINT "INFORMATION FROM THE EPDATA PROGRAM."

RUN LSTPER
PRINT "THIS PROGRAM IS USED TO AUTOMATICALLY COLLECT EVOKED POTENTIAL"  
PRINT "DATA."

PRINT ; PRINT "THE FOLLOWING INFORMATION SHOULD BE KNOWN BEFORE RUNNING"  
PRINT ; PRINT "THE STIMULUS DELAY TIME"
PRINT "NUMBER OF STIMULI PER SECOND"
PRINT "NUMBER OF SWEEPS TO BE COLLECTED"
PRINT "NUMBER OF CHANNELS TO BE RECORDED"
PRINT "SUBJECT'S I.D. NUMBER"
PRINT "LENGTH OF SWEEP"

RUN EPDATA
PRINT ; PRINT "THIS PROGRAM IS USED TO TRANSFER PERSONAL, EVOKED"  
PRINT "POTENTIAL, OR PARAMETER DATA FROM THE MED-80 TO THE ECLIPSE."

RUN MEDECL
PRINT "THIS PROGRAM IS FOR THE PLOTTING OF EVOKED POTENTIAL DATA ONTO THE"  
PRINT "ZETA PLOTTER."
PRINT ; PRINT "WHEN STANDING IN FRONT OF THE ZETA PLOTTER THE PEN"  
PRINT "SHOULD BE IN THE UPPER RIGHT HAND CORNER."
PRINT ; PRINT "ANSWER ALL QUESTIONS WITH A Y (YES) OR N (NO),"
PRINT "UNLESS OTHERWISE REQUESTED."

RUN PLOTEP
STOP ; END
LIST PROGBU

10 REM DIMENSION VARIABLES
20 DIM PROGS(10), X$(1)
30 REM ADVANCE PAPER
40 PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
50 PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
60 PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
70 REM ENABLES USER TO OBTAIN INFORMATION ON A PROGRAM IS THE E.P SERIES
80 PRINT "DO YOU NEED A LISTING OF THE PROGRAMS BEING OFFERED OR"
90 INPUT "DO YOU NEED SOME INFORMATION ON THE PURPOSE OF THE PROGRAMS? - " , X$  
100 IF X$(1) = "Y" THEN 120 ; IF X$(1) = "N" THEN 290 ; GOTO 80
110 REM GENERAL INFORMATION ON A PROGRAM
120 PRINT "THE PROGRAM DISK SHOULD BE IN DISK DRIVE # 1."
130 PRINT "THE DATA DISK SHOULD BE IN DISK DRIVE # 2."
140 PRINT ; PRINT " NOTE: IF YOU MAKE A ERROR NO ERROR MESSAGE IS PRINTED."
150 PRINT " THE QUESTION WILL JUST BE REASKED."
160 PRINT ; PRINT
170 REM LISTING OF PROGRAMS AND NAMES USED TO CALL THEM
180 PRINT " THE FOLLOWING IS A LISTING OF ALL PROGRAMS IN THE EVOKED POTENTIAL"
190 PRINT " SERIES AND THE SYMBOLS TO BE USED WHEN CALLING A PROGRAM."
200 PRINT ; PRINT " INPUT OF PERSONAL DATA - PERDAT"
210 PRINT " COLLECTION OF PARAMETER AND EVOKED POTENTIAL DATA - EPDATA"
220 PRINT " PLOTTING OF THE EVOKED POTENTIAL DATA ON THE ZETA PLOTTER - PLOTEP"
230 PRINT " OBTAINING A DETAILED LISTING OF ALL EXPERIMENTAL DATA - LSTPER"
240 PRINT " TRANSFER OF DATA FILES TO THE ECLIPSE - MEDECL"
250 PRINT ; PRINT " IF YOU DESIRE TO EXIT THE PROGRAM TYPE 'END' AS A "
260 PRINT ; PRINT " RESPONSE TO THE FOLLOWING QUESTION."
270 REM CALL DESIRED PROGRAM
280 PRINT ; INPUT " INPUT THE SYMBOL OF THE DESIRED PROGRAM - " , PROGS$(1)
290 PRINT " IS " , PROGS$(1), " CORRECT?"
300 INPUT X$(1)
310 IF X$(1) = "Y" THEN 330 ; GOTO 290
320 PRINT
330 REM CONDITIONAL BRANCH TO PROGRAM SELECTED
340 IF PROGS$(1) = "PERDAT" THEN 440
350 IF PROGS$(1) = "LSTPER" THEN 660 ; IF PROGS$(1) = "EPDATA" THEN 710
360 IF PROGS$(1) = "MEDECL" THEN 820 ; IF PROGS$(1) = "PLOTEP" THEN 860
370 IF PROGS$(1) = "END" THEN 940 ; IF PROGS$(1) = PROGS$(1) THEN 190
380 PRINT " INFORMATION ON THE PERSONAL DATA PROGRAM"
390 PRINT " THIS PROGRAM IS USED TO OBTAIN AND SAVE FILES ON PERSONAL DATA FOR"
400 PRINT " HUMAN AND RHESUS MONKEY SUBJECTS USED IN THE NDBL EVOKED"
410 PRINT " POTENTIAL PROGRAM."
420 PRINT ; PRINT " BEFORE RUNNING A SUBJECT THE FOLLOWING INFORMATION"
430 PRINT " SHOULD BE OBTAINED:"
440 PRINT ; PRINT " SUBJEC'T'S NAME"
450 PRINT " SUBJECT'S I.D. NUMBER"
460 PRINT " G-FORCE TO BE USED IN EXPERIMENT"
470 PRINT " RUN NUMBER FOR EXPERIMENT"
480 PRINT " TIME-OF-DAY"

8.3
PRINT "DATE OF EXPERIMENT"
PRINT "ROOM AIR TEMPERATURE (IN F)"
PRINT "SUBJECT'S BODY TEMPERATURE (IN F)"
PRINT "SUBJECT'S HEART RATE (BEATS PER MIN.)"
PRINT "NUMBER OF CHANNELS TO BE RECORDED"
PRINT "MONTAGES TO BE USED IN RECORDING"
PRINT; PRINT "BE SURE THAT YOU KNOW THE CORRECT FILE SUBSCRIPT "
PRINT "ANSWER ALL QUESTIONS WITH Y (YES), N (NO), OR A N (N/A),
UNLESS OTHERWISE REQUESTED. (NOTE) THE N IS ALSO USED IN"
PRINT "THE 'IS VARIABLE CORRECT' QUESTION."
RUN PERDAT
REM INFORMATION ON THE LISTING OF PERSONAL AND PARAMETER FILE VARIABLES
PRINT "THIS PROGRAM IS USED TO OBTAIN A DETAILED LISTING OF ALL OF THE"
PRINT "PERSONAL DATA FROM THE PERDAT PROGRAM AND THE PARAMETER"
PRINT "INFORMATION FROM THE EPDATA PROGRAM."
RUN LSTPER
REM INFORMATION ON THE EVOKED POTENTIAL DATA PROGRAM
PRINT "THIS PROGRAM IS USED TO AUTOMATICALLY COLLECT EVOKED POTENTIAL"
PRINT "THE FOLLOWING INFORMATION SHOULD BE KNOWN BEFORE RUNNING"
PRINT; PRINT "THE STIMULUS DELAY TIME"
PRINT "NUMBER OF STIMULI PER SECOND"
PRINT "NUMBER OF SWEEPS TO BE COLLECTED"
PRINT "NUMBER OF CHANNELS TO BE RECORDED"
PRINT "SUBJECT'S I.D. NUMBER"
PRINT "LENGTH OF SWEEP"
RUN EPDATA
REM INFORMATION ON THE TRANSFER OF DATA PROGRAM
PRINT "THIS PROGRAM IS USED TO TRANSFER PERSONAL, EVOKED"
PRINT "POTENTIAL, OR PARAMETER DATA FROM THE MED-80 TO THE ECLIPSE."
RUN MEDECL
REM INFORMATION ON THE PLOTTING OF EVOKED POTENTIALS PROGRAM
PRINT "THIS PROGRAM IS FOR THE PLOTTING OF EVOKED POTENTIAL DATA ONTO THE"
PRINT "ZETA PLOTTER."
PRINT; PRINT "WHEN STANDING IN FRONT OF THE ZETA PLOTTER THE PEN"
PRINT "SHOULD BE IN THE UPPER RIGHT HAND CORNER."
PRINT "ANSWER ALL QUESTIONS WITH A Y (YES) OR N (NO),
UNLESS OTHERWISE REQUESTED."
RUN PLOTEP
REM EXIT PROGRAM
STOP; END
LIST PERDAT

10 DIM RCH$(1), BLANK$(50), XX$(60), X$(1), CH$(2), NAME$(35), COMMT$(60)
20 DIM SUBNO$(5), DATE$(9), GFORC$(6), RUNNO$(7), TIMDA$(6), AIRTP$(6), SUBTP$(5)
30 DIM HEART$(4), IMAS$(5), INAS$(5), IN2AS$(5), IN3AS$(5), IN4AS$(5), IN5AS$(5)
40 DIM INA7$(5), INAS$(5), INE$(2), INB$(5), IN2$(5), INB$(5), INB$(5), IN5$(5)
50 DIM INEB$(5), INB$(5), INEB$(5), Y$(230)
60 N1(1) = 0; N2(1) = 0; N3(1) = 0; N4(1) = 0; W = 1; Y$ = ""
70 BLANK$ = ""
80 PRINT ; PRINT "INPUT NAME AS FOLLOWS."
90 P = 35; GOSUB 1930
100 INPUT "LAST, FIRST, THEN M.I. - ", NAME$(1)
110 XX$(1) = NAME$(1); GOSUB 2060
120 IF X$(1) = "Y" THEN 130; IF X$(1) = "M" THEN 140; GOTO 100
130 Y$(W) = BLANK$(1); Y$(W) = NAME$(1)
140 W = W + P; F = 5; GOSUB 1930
150 PRINT ; PRINT "M = HUMAN SUBJECT    F = RHESUS SUBJECT"
160 INPUT "SUBJECT'S I.D. NUMBER - ", SUENO$(1)
170 XX$(1) = SUENO$(1); GOSUB 2060
180 IF X$(1) = "Y" THEN 190; IF X$(1) = "M" THEN 200; GOTO 150
190 Y$(W) = BLANK$(1); Y$(W) = SUENO$(1)
200 W = W + P; F = 9; GOSUB 1930
210 INPUT "MON.-DAY/YEAR - ", DATE$(1)
220 XX$(1) = DATE$(1); GOSUB 2060
230 IF X$(1) = "Y" THEN 240; IF X$(1) = "M" THEN 250; GOTO 210
240 Y$(W) = BLANK$(1); Y$(W) = DATE$(1)
250 W = W + P; F = 6; GOSUB 1930
260 PRINT ; PRINT " IF G-FORCE IS ELLICIT : ENTER THE A WITH A & FIRST (E.G. 024:1)"
270 INPUT "G-FORCE - ", GFORC$(1)
280 XX$(1) = GFORC$(1); GOSUB 2060
290 IF X$(1) = "Y" THEN 300; IF X$(1) = "M" THEN 340; GOTO 260
300 IF GFORC$(4, 5) = "Y" OR GFORC$(4, 5) = "Z" THEN 310; GOTO 260
310 IF GFORC$(5, 5) = "Y" OR GFORC$(5, 5) = "Z" THEN 330
320 IF GFORC$(5, 5) = GFORC$(1) THEN 260
330 Y$(W) = BLANK$(1); Y$(W) = GFORC$(1)
340 W = W + P; F = 7; GOSUB 1930
350 INPUT "A $ NUMBER - ", RUNNO$(1)
360 XX$(1) = RUNNO$(1); GOSUB 2060
370 IF X$(1) = "Y" THEN 380; IF X$(1) = "M" THEN 390; GOTO 350
380 Y$(W) = BLANK$(1); Y$(W) = RUNNO$(1)
390 W = W + P; F = 6; GOSUB 1930
400 INPUT "TIME-OF-DA\O\A 24 HOUR CLOCK (I.E. 14:33) - ", TIMDA$(1)
410 XX$(1) = TIMDA$(1); GOSUB 2060
420 IF X$(1) = "Y" THEN 430; IF X$(1) = "M" THEN 440; GOTO 400
430 Y$(W) = BLANK$(1); Y$(W) = TIMDA$(1)
440 W = W + P; F = 5; GOSUB 1930
450 INPUT "AIR TEMPERATURE (IN F) - ", AIRTP$(1)
460 XX$(1) = AIRTP$(1); GOSUB 2060
470 IF X$(1) = "Y" THEN 480; IF X$(1) = "M" THEN 500; GOTO 450
480 IF AIRTP$(1) < 69 OR AIRTP$(1) > 81 THEN 450
490 Y$(U) = BLANK$(1); Y$(W) = AIRTP$(1)
500 W = W + P; P = 5; GOSUB 1930
510 INPUT "SUBJECT'S TEMPERATURE (IN F) - ", SUBTP$(1)
520 XX$(1) = SUBTP$(1); GOSUB 2060
530 IF XX$(1) = "Y" THEN 540; IF XX$(1) = "M" THEN 550; GOTO 510
540 Y$(U) = BLANK$(1); Y$(W) = SUBTP$(1)
550 W = W + P; P = 4; GOSUB 1930
560 INPUT "SUBJECT'S HEART RATE (BEATS/Min.) - ", HEART$(1)
570 XX$(1) = HEART$(1); GOSUB 2060
580 IF XX$(1) = "Y" THEN 590; IF XX$(1) = "M" THEN 600; GOTO 560
590 Y$(U) = BLANK$(1); Y$(W) = HEART$(1)
600 U = W + P; P = 0; GOSUB 1930
610 INPUT "PRINT COMMENTS (USE LESS THAN 60 CHARACTERS) - ", COMNT$(1)
620 XX$(1) = COMNT$(1); GOSUB 2060
630 IF XX$(1) = "Y" THEN 640; IF XX$(1) = "M" THEN 650; GOTO 610
640 Y$(U) = BLANK$(1); Y$(W) = COMNT$(1)
650 W = W + P; P = 2; GOSUB 1930
660 INPUT "NUMBER OF CHANNELS BEING RECORDED (1 - 8) - ", CH$(1)
670 XX$(1) = CH$(1); GOSUB 2060
680 IF XX$(1) = "Y" THEN 690; IF XX$(1) = "M" THEN 700; GOTO 660
690 IF CH$(1) < "1" OR CH$(1) > "8" THEN 680
700 Y$(U) = BLANK$(1); Y$(W) = CH$(1)
710 U = W + P
720 IF CH$(1) = "1" THEN 1380; IF CH$(1) = "2" THEN 1290
730 IF CH$(1) = "3" THEN 1200; IF CH$(1) = "4" THEN 1110
740 IF CH$(1) = "5" THEN 1020; IF CH$(1) = "6" THEN 930
750 IF CH$(1) = "7" THEN 840
760 F = 0; GOSUB 1930
770 PRINT "INPUT MONTAGE FOR CHANNEL B"
780 INPUT " INPUT A - ", IN$A$(1)
790 Y$(U) = BLANK$(1); Y$(W) = IN$A$(1)
800 W = U + P; P = 3; GOSUB 1930
810 PRINT " INPUT B - ", IN$B$(1)
820 Y$(U) = BLANK$(1); Y$(W) = IN$B$(1)
830 U = W + P; GOTO 850
840 P = 10; GOSUB 1930; W = U + P
850 PRINT "INPUT MONTAGE FOR CHANNEL C"
860 P = 5; GOSUB 1930
870 PRINT " INPUT A - ", IN$A$(1)
880 Y$(U) = BLANK$(1); Y$(W) = IN$A$(1)
890 W = U + P; P = 5; GOSUB 1930
900 PRINT " INPUT B - ", IN$B$(1)
910 Y$(U) = BLANK$(1); Y$(W) = IN$B$(1)
920 U = W + P; GOTO 940
930 P = 20; GOSUB 1930; W = U + P
940 PRINT "INPUT MONTAGE FOR CHANNEL D"
950 P = 5; GOSUB 1930
960 PRINT " INPUT A - ", IN$A$(1)
970 Y$(U) = BLANK$(1); Y$(W) = IN$A$(1)
980 W = U + P; P = 5; GOSUB 1930
990 PRINT " INPUT B - ", IN$B$(1)
1000 Y$(W) = BLANK$(1); Y$(W) = INB$(1)
1010 U = U + P; GOTO 1030
1020 P = 30; GOSUB 1930; W = W + P
1030 PRINT "INPUT MONTAGE FOR CHANNEL 5"
1040 F = 5; GOSUB 1930
1050 INPUT " INPUT A - ", INA$(1)
1060 Y$(W) = BLANK$(1); Y$(W) = INA$(1)
1070 U = U + P; P = 5; GOSUB 1930
1080 INPUT " INPUT B - ", INB$(1)
1090 Y$(W) = BLANK$(1); Y$(W) = INB$(1)
1100 W = W + P; GOTO 1120
1110 P = 40; GOSUB 1930; W = W + P
1120 PRINT "INPUT MONTAGE FOR CHANNEL 4"
1130 P = 5; GOSUB 1930
1140 INPUT " INPUT A - ", INA$(1)
1150 Y$(W) = BLANK$(1); Y$(W) = INA$(1)
1160 U = U + P; P = 5; GOSUB 1930
1170 INPUT " INPUT B - ", INB$(1)
1180 Y$(W) = BLANK$(1); Y$(W) = INB$(1)
1190 W = W + P; GOTO 1210
1200 P = 50; GOSUB 1930; W = W + P
1210 PRINT "INPUT MONTAGE FOR CHANNEL 3"
1220 P = 5; GOSUB 1930
1230 INPUT " INPUT A - ", INA$(1)
1240 Y$(W) = BLANK$(1); Y$(W) = INA$(1)
1250 U = U + P; P = 5; GOSUB 1930
1260 INPUT " INPUT B - ", INB$(1)
1270 Y$(W) = BLANK$(1); Y$(W) = INB$(1)
1280 W = W + P; GOTO 1300
1290 P = 50; GOSUB 1930
1300 PRINT "INPUT MONTAGE FOR CHANNEL 2"
1310 P = 5; GOSUB 1930
1320 INPUT " INPUT A - ", INA$(1)
1330 Y$(W) = BLANK$(1); Y$(W) = INA$(1)
1340 U = U + P; P = 5; GOSUB 1930
1350 INPUT " INPUT B - ", INB$(1)
1360 Y$(W) = BLANK$(1); Y$(W) = INB$(1)
1370 W = W + P; GOTO 1390
1380 P = 70; GOSUB 1930; W = W + P
1390 PRINT "INPUT MONTAGE FOR CHANNEL 1"
1400 P = 5; GOSUB 1930
1410 INPUT " INPUT A - ", INA$(1)
1420 Y$(W) = BLANK$(1); Y$(W) = INA$(1)
1430 U = U + P; P = 5; GOSUB 1930
1440 INPUT " INPUT B - ", INB$(1)
1450 Y$(W) = BLANK$(1); Y$(W) = INB$(1)
1460 W = W + P; P = 5; GOSUB 1930
1470 PRINT; PRINT " CH. M", "INPUT A", "INPUT B"
1480 SS(1) = 225
1490 FOR S = 1 TO 8
1500 SS(1) = SS(1) - 10
1510 PRINT $ IN 4, 4, Y$(SS(1)+15, SS(1)+9), Y$(SS(1)+5, SS(1)+9)
1520 NEXT S
1530 PRINT ; INPUT "ARE THERE ANY ERRORS IN THE ABOVE - ", X$
1540 IF X$ = "Y" THEN 1550 ; IF X$ = "N" THEN 1720 ; GOTO 1530
1550 INPUT "WHICH CHANNEL IS INCORRECT - ", RCH$(1)
1560 PRINT "IS ", RCH$(1), " CORRECT?"
1570 INPUT X$(1)
1580 IF X$(1) = "Y" THEN 1590 ; GOTO 1550
1590 IF RCH$(1) = "1" THEN 1640 ; IF RCH$(1) = "2" THEN 1650
1600 IF RCH$(1) = "3" THEN 1660 ; IF RCH$(1) = "4" THEN 1670
1610 IF RCH$(1) = "5" THEN 1680 ; IF RCH$(1) = "6" THEN 1690
1620 IF RCH$(1) = "7" THEN 1700 ; IF RCH$(1) = "8" THEN 1710
1630 IF RCH$(1) = "9" THEN 1720 ; GOTO 1530
1640 N1(1) = 215 ; N2(1) = 219 ; N3(1) = 220 ; N4(1) = 224 ; GOSUB 1970 ; GOTO 1470
1650 N1(1) = 205 ; N2(1) = 209 ; N3(1) = 210 ; N4(1) = 214 ; GOSUB 1970 ; GOTO 1470
1660 N1(1) = 155 ; N2(1) = 159 ; N3(1) = 200 ; N4(1) = 204 ; GOSUB 1970 ; GOTO 1470
1670 N1(1) = 185 ; N2(1) = 189 ; N3(1) = 190 ; N4(1) = 194 ; GOSUB 1970 ; GOTO 1470
1680 N1(1) = 175 ; N2(1) = 179 ; N3(1) = 180 ; N4(1) = 184 ; GOSUB 1970 ; GOTO 1470
1690 N1(1) = 165 ; N2(1) = 169 ; N3(1) = 170 ; N4(1) = 174 ; GOSUB 1970 ; GOTO 1470
1700 N1(1) = 155 ; N2(1) = 159 ; N3(1) = 160 ; N4(1) = 164 ; GOSUB 1970 ; GOTO 1470
1710 N1(1) = 145 ; N2(1) = 149 ; N3(1) = 150 ; N4(1) = 54 ; GOSUB 1970 ; GOTO 1470
1720 PRINT ; INPUT "STORE THIS PERSONAL DATA - ", X$
1730 IF X$ = "Y" THEN 1740 ; IF X$ = "N" THEN 1890 ; GOTO 1720
1740 PUT 2 IN PDV ; PUT 0 IN PST ; PUT 1024 IN PSZ
1750 PRINT ; INPUT "INPUT FILE SUBSCRIPT (3 & 5 0 - 999) - ", SUB
1760 PRINT "IS ", SUB, " CORRECT?"
1770 INPUT X$
1780 IF X$ = "Y" THEN 1790 ; GOTO 1750
1790 IF SUB < 0 OR SUB > 999 THEN 1750 ; PUT SUB IN FFS
1800 BASE = VAL(SUBR$(3, 4)) ; PUT BASE IN FFS
1810 HEAD = ASC(SUBR$(1)) ; IF HEAD = 4, THEN 1820 ; PUT 2, 0 IN PFH ; GOTO 1830
1820 PUT 02, 0 IN PFH
1830 FDA S = 1 TO 4
1840 A$ = Y$(S, S) ; DMEM(S) = ASC(A$) ; NEXt S
1850 XEQ DWR
1860 PRINT ; INPUT "COLLECT EVOKED POTENTIAL DATA NOW - ", X$
1870 IF X$ = "Y" THEN 1880 ; IF X$ = "N" THEN 1890 ; GOTO 1860
1880 RUN EFDATA
1890 PRINT ; INPUT "RUN ANOTHER PROGRAM - ", X$
1900 IF X$ = "Y" THEN 1910 ; IF X$ = "N" THEN 2090 ; GOTO 1890
1910 RUN EPFRG
1920 STOP
1930 FOR S = 4 TO W + (P - 1)
1940 Y$(S, S) = "#"
1950 NEAt S
1960 RETURN ; STp
1970 INPUT "CORRECT 'INPUT A' VALUE - ", IM$$(1)
1980 XX$(1) = IM$$(1) ; GOSUB 2000
1990 IF XX$(1) = "Y" THEN 2000 ; GOTO 1970
2000 Y$(W$(1), N2$(1)) = IM$$(1)
2010 INPUT "CORRECT 'INPUT B' VALUE - ", IM$$(1)
2020 XX$(1) = INB$(1); GOSUB 2060
2030 IF X$(1) = "Y" THEN 2040; GOTO 2010
2040 Y$(M3(1),M4(1)) = INB$(1)
2050 RETURN; STOP
2060 PRINT "IS ", XX$(1), " CORRECT?"
2070 INPUT X$(1)
2080 RETURN; STOP
2090 END
REM DIMENSION VARIABLES.
10 DIM RCHS(1), Y$(25), BLANK$(60), X$(1), CH$(2), NAME$(35), COMMT$(60)
20 DIM SUBNO$(5), DATE$(9), GFORC$(6), RUNNO$(7), TIMDA$(6), AOITP$(6), SUBTP$(5)
30 DIM HEARTS(4), FROM$(5), FROM1$(5), FROM2$(5), FROM3$(5), FROM4$(5), FROM5$(5)
40 DIM FROM6$(5), FROM7$(5), FROM8$(5), TO$(5), 'R0$(5), T02$(5), T03$(5), T04$(5)
50 DIM T05$(5), T06$(5), T07$(5), T08$(5)
55 REM ASSIGN VALUES TO SOME VARIABLES.
60 N1(1) = 0; N2(1) = 0; N3(1) = 0; N4(1) = 0; W = 1; Y$ = ""
70 BLANK$ = ""
75 REM FOR THE FOLLOWING VARIABLES USER INPUTS THEIR VALUES FOR THIS EXPERIMENT
76 REM A VISUAL CHECK ON THE VALUE IS PROVIDED. AFTER THE CORRECT VALUE HAS
77 REM BEEN ENTERED THE VARIABLE IS STORED IN THE PERSONAL DATA FILE IN ITS
78 REM CORRECT LOCATION, REPRESENTED BY 'U'. THIS LOCATION IS INCREASED BY THE
79 REM SIZE OF THE VAR UNDER CONSIDERATION 'P', AND PROG ADVANCES TO NEXT VAR.
80 PRINT; PRINT "INPUT NAME AS FOLLOWS."
90 P = 35; GOSUB 1930
100 INPUT "LAST, FIRST, THEN M.I. - ", NAME$(1)
110 XX$(1) = NAME$(1); GOSUB 2060
120 IF X$(1) = "Y" THEN 130; IF X$(1) = "N" THEN 140; GOTO 100
130 Y$(U) = BLANK$(1); Y$(U) = NAME$(1)
140 W = W + P; P = 5; GOSUB 1930
150 PRINT; PRINT "H = HUMAN SUBJECT      R = RHESUS SUBJECT"
160 INPUT "SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
170 XX$(1) = SUBNO$(1); GOSUB 2060
180 IF X$(1) = "Y" THEN 190; IF X$(1) = "" THEN 200; GOTO 100
190 Y$(U) = BLANK$(1); Y$(U) = SUBNO$(1)
200 W = W + P; P = 9; GOSUB 1930
210 INPUT "MON/DAY/YEAR - ", DATE$(1)
220 XX$(1) = DATE$(1); GOSUB 2060
230 IF X$(1) = "Y" THEN 240; IF X$(1) = "N" THEN 250; GOTO 210
240 Y$(U) = BLANK$(1); Y$(U) = DATE$(1)
250 W = W + P; P = 6; GOSUB 1930
260 PRINT; PRINT "IF S-FORCE IS BELOW 10 ENTER THE # WITH A 0 FIRST (I.E. 036-Y)"
270 INPUT "G-FORCE - ", GFORC$(1)
280 XX$(1) = GFORC$(1); GOSUB 2060
290 IF X$(1) = "Y" THEN 300; IF X$(1) = "N" THEN 340; GOTO 260
300 IF GFORC$(4,4) = "+" OR GFORC$(4,4) = "-" THEN 310; GOTO 260
310 IF GFORC$(5,5) = "X" OR GFORC$(5,5) = "Y" OR GFORC$(5,5) = "Z" THEN 330
320 IF GFORC$(1) = GFORC$(1) THEN 260
330 Y$(U) = BLANK$(1); Y$(U) = GFORC$(1)
340 W = W + P; P = 7; GOSUB 1930
350 INPUT "RUN NUMBER - ", RUNNO$(1)
360 XX$(1) = RUNNO$(1); GOSUB 2060
370 IF X$(1) = "Y" THEN 380; IF X$(1) = "N" THEN 390; GOTO 350
380 Y$(U) = BLANK$(1); Y$(U) = RUNNO$(1)
390 W = W + P; P = 6; GOSUB 1930
400 INPUT "TIME-OF-DAY ON A 24 HOUR CLOCK (I.E. 14:33) - ", TIMDA$(1)
410 XX$(1) = TIMDA$(1); GOSUB 2060
420 IF X$(1) = "Y" THEN 430 ; IF X$(1) = "#" THEN 440 ; GOTO 400
430 Y$(U) = BLANK$(1) ; Y$(U) = TIMDA$(1)
440 W = W + P ; P = 5 ; GOSUB 1930
450 INPUT "AIR TEMPERATURE (IN F) - ", AIRTP$(1)
460 XX$(1) = AIRTP$(1) ; GOSUB 2060
470 IF X$(1) = "Y" THEN 480 ; IF X$(1) = "#" THEN 500 ; GOTO 450
475 REM CHECK VALUE OF AIR TEMPERATURE.
480 IF AIRTP$(1) < "69" OR AIRTP$(1) > "81" THEN 450
490 Y$(U) = BLANK$(1) ; Y$(U) = AIRTP$(1)
500 W = W + P ; P = 5 ; GOSUB 1930
510 INPUT "SUBJECT'S TEMPERATURE (IN F) - ", SUBTP$(1)
520 XX$(1) = SUBTP$(1) ; GOSUB 2060
530 IF X$(1) = "Y" THEN 540 ; IF X$(1) = "#" THEN 550 ; GOTO 510
540 Y$(U) = BLANK$(1) ; Y$(U) = SUBTP$(1)
550 W = W + P ; P = 4 ; GOSUB 1930
560 INPUT "SUBJECT'S HEART RATE (BEATS/MIN.) - ", HEART$(1)
570 XX$(1) = HEART$(1) ; GOSUB 2060
580 IF X$(1) = "Y" THEN 590 ; IF X$(1) = "#" THEN 600 ; GOTO 560
590 Y$(U) = BLANK$(1) ; Y$(U) = HEART$(1)
600 W = W + P ; P = 2 ; GOSUB 1930
610 INPUT "PRINT COMMENTS (USE LESS THAN 60 CHARACTERS) - ", COMNT$(1)
620 XX$(1) = COMNT$(1) ; GOSUB 2060
630 IF X$(1) = "Y" THEN 640 ; IF X$(1) = "#" THEN 650 ; GOTO 610
640 Y$(U) = BLANK$(1) ; Y$(U) = COMNT$(1)
650 W = W + P ; P = 2 ; GOSUB 1930
660 INPUT "NUMBER OF CHANNELS BEING RECORDED (1 - 8) - ", CHS$(1)
670 XX$(1) = CHS$(1) ; GOSUB 2060
680 IF X$(1) = "Y" THEN 690 ; IF X$(1) = "#" THEN 710 ; GOTO 660
685 REM CHECK VALUE OF THE NUMBER OF CHANNELS.
690 IF CHS$(1) < "1" OR CHS$(1) > "8" THEN 660
700 Y$(U) = BLANK$(1) ; Y$(U) = CHS$(1)
710 W = W + P
715 REM INPUT A CHANNELS MONTAGE FOR THE EXPERIMENT.
720 IF CHS$(1) = "1" THEN 1380 ; IF CHS$(1) = "2" THEN 1290
730 IF CHS$(1) = "3" THEN 1200 ; IF CHS$(1) = "4" THEN 1110
740 IF CHS$(1) = "5" THEN 1020 ; IF CHS$(1) = "6" THEN 930
750 IF CHS$(1) = "7" THEN 840
760 P = 5 ; GOSUB 1930
770 PRINT "INPUT MONTAGE FOR CHANNEL 8"
780 INPUT " INPUT A - ", FROMB$(1)
790 Y$(U) = BLANK$(1) ; Y$(U) = FROMB$(1)
800 W = W + P ; P = 5 ; GOSUB 1930
810 INPUT " INPUT B - ", TDB$(1)
820 Y$(U) = BLANK$(1) ; Y$(U) = TDB$(1)
830 W = W + P ; GOTO 850
840 P = 10 ; GOSUB 1930 ; W = W + P
850 PRINT "INPUT MONTAGE FOR CHANNEL 7"
860 P = 5 ; GOSUB 1930
870 INPUT " INPUT A - ", FROM7$(1)
880 Y$(U) = BLANK$(1) ; Y$(U) = FROM7$(1)
890 W = W + P ; P = 5 ; GOSUB 1930
900 INPUT " INPUT B - ", T07$(1)
910 Y$(W) = BLANK$(1) ; Y$(W) = T07$(1)
920 W = W + P ; GOTO 940
930 P = 20 ; GOSUB 1930 ; W = W + P
940 PRINT "INPUT MONTAGE FOR CHANNEL 6"
950 P = 5 ; GOSUB 1930
960 INPUT " INPUT A - ", FROM6$(1)
970 Y$(W) = BLANK$(1) ; Y$(W) = FROM6$(1)
980 W = W + P ; P = 5 ; GOSUB 1930
990 INPUT " INPUT B - ", T06$(1)
1000 Y$(W) = BLANK$(1) ; Y$(W) = T06$(1)
1010 U = U + P ; GOTO 1030
1020 P = 30 ; GOSUB 1930 ; W = W + P
1030 PRINT "INPUT MONTAGE FOR CHANNEL 5"
1040 P = 5 ; GOSUB 1930
1050 INPUT " INPUT A - ", FROM5$(1)
1060 Y$(W) = BLANK$(1) ; Y$(W) = FROM5$(1)
1070 W = W + P ; P = 5 ; GOSUB 1930
1080 INPUT " INPUT B - ", T05$(1)
1090 Y$(W) = BLANK$(1) ; Y$(W) = T05$(1)
1100 W = W + P ; GOTO 1120
1110 P = 40 ; GOSUB 1930 ; W = W + P
1120 PRINT "INPUT MONTAGE FOR CHANNEL 4"
1130 P = 5 ; GOSUB 1930
1140 INPUT " INPUT A - ", FROM4$(1)
1150 Y$(W) = BLANK$(1) ; Y$(W) = FROM4$(1)
1160 W = W + P ; P = 5 ; GOSUB 1930
1170 INPUT " INPUT B - ", T04$(1)
1180 Y$(W) = BLANK$(1) ; Y$(W) = T04$(1)
1190 W = W + P ; GOTO 1210
1200 P = 50 ; GOSUB 1930 ; W = W + P
1210 PRINT "INPUT MONTAGE FOR CHANNEL 3"
1220 P = 5 ; GOSUB 1930
1230 INPUT " INPUT A - ", FROM3$(1)
1240 Y$(W) = BLANK$(1) ; Y$(W) = FROM3$(1)
1250 W = W + P ; P = 5 ; GOSUB 1930
1260 INPUT " INPUT B - ", T03$(1)
1270 Y$(W) = BLANK$(1) ; Y$(W) = T03$(1)
1280 U = U + P ; GOTO 1320
1290 P = 60 ; GOSUB 1930
1300 PRINT "INPUT MONTAGE FOR CHANNEL 2"
1310 P = 5 ; GOSUB 1930
1320 INPUT " INPUT A - ", FROM2$(1)
1330 Y$(W) = BLANK$(1) ; Y$(W) = FROM2$(1)
1340 W = W + P ; P = 5 ; GOSUB 1930
1350 INPUT " INPUT B - ", T02$(1)
1360 Y$(W) = BLANK$(1) ; Y$(W) = T02$(1)
1370 W = W + P ; GOTO 1390
1380 P = 70 ; GOSUB 1930 ; W = W + P
1390 PRINT "INPUT MONTAGE FOR CHANNEL 1"
1400 P = 5 ; GOSUB 1930
INPUT " INPUT A - ", FORM$(1)
Y$(W) = BLANK$(1); Y$(W) = FROM$(1)
W = W + P; P = 5; GOSUB 1930
INPUT " INPUT B - ", TO$(1)
Y$(W) = BLANK$(1); Y$(W) = TO$(1)
W = W + P; P = 5; GOSUB 1930
REM PRINT OUT THE MONTAGE INFORMATION FOR ALL CHANNELS.
PRINT; PRINT " CH. N", "INPUT A", "INPUT B"
SS(1) = 225
FOR I = 1 TO 8
SS(1) = SS(1) - 10
PRINT I IN 4,4 , Y$(SS(1),SS(1)+4), Y$(SS(1)+5,SS(1)+9)
NEXT I
REM ENABLES USER TO CHANGE A MONTAGES VALUE.
PRINT; PRINT "ARE THE ABOVE MONTAGES CORRECT - ", X$
IF X$ = "Y" THEN 1720; IF X$ = "N" THEN 1550; GOTO 1530
REM USER SELECTS CHANNEL WITH ERRORS IN ITS MONTAGE VALUES.
INPUT " WHICH CHANNEL IS INCORRECT - ", RCH$(1)
PRINT "IS ", RCH$(1), " CORRECT?"
INPUT X$(1)
IF X$(1) = "Y" THEN 1590; GOTO 1550
REM REPLACEMENT OF THE BAD MONTAGES WITH THE CORRECT VALUE.
IF RCH$(1) = "1" THEN 1640; IF RCH$(1) = "2" THEN 1650
IF RCH$(1) = "3" THEN 1660; IF RCH$(1) = "4" THEN 1670
IF RCH$(1) = "5" THEN 1680; IF RCH$(1) = "6" THEN 1690
IF RCH$(1) = "7" THEN 1700; IF RCH$(1) = "8" THEN 1710
IF RCH$(1) = RCH$(1) THEN 1550
N1(1)=215; N2(1)=219; N3(1)=220; N4(1)=224; GOSUB 1970; GOTO 1470
N1(1)=205; N2(1)=209; N3(1)=210; N4(1)=214; GOSUB 1970; GOTO 1470
N1(1)=195; N2(1)=199; N3(1)=200; N4(1)=204; GOSUB 1970; GOTO 1470
N1(1)=185; N2(1)=189; N3(1)=190; N4(1)=194; GOSUB 1970; GOTO 1470
N1(1)=175; N2(1)=179; N3(1)=180; N4(1)=184; GOSUB 1970; GOTO 1470
N1(1)=165; N2(1)=169; N3(1)=170; N4(1)=174; GOSUB 1970; GOTO 1470
N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164; GOSUB 1970; GOTO 1470
N1(1)=145; N2(1)=149; N3(1)=150; N4(1)=154; GOSUB 1970; GOTO 1470
REM ENABLES USER TO STORE THE PERSONAL DATA FILE
PRINT; PRINT "STORE THIS PERSONAL DATA - ", X$
IF X$ = "Y" THEN 1740; IF X$ = "N" THEN 1890; GOTO 1720
REM SELECTS DISK DRIVE NUMBER 2 AND SETS THE START AND SIZE OF DMEM.
PUT 2 IN PDV; PUT 0 IN PST; PUT 1024 IN PSZ
PRINT; PRINT " INPUT FILE SUBSCRIPT (3 0'S 0 - 999) - ", S
PRINT "IS ", S, " CORRECT?"
INPUT X$
REM CHANGE NORMAL DATA INTO ASCII CODE.
FOR I = 1 TO W
A$ = Y$(I,I); DMEH(I) = ASC(A$); NEXT I
REM WRITE FILE ON DISK.
XED DVR
REM ENABLES USER TO GO DIRECTLY TO THE PROGRAM WHICH COLLECTS E.P. DATA
PRINT; INPUT "COLLECT EVOKED POTENTIAL DATA NOW - ", X$
IF X$ = "Y" THEN 1880; IF X$ = "N" THEN 1890; GOTO 1860
RUN EPDATA
REM ENABLES USER TO RETURN TO THE POINTER PROG. AND RUN ANOTHER PROGRAM
PRINT; INPUT "RUN ANOTHER PROGRAM - ", X$
IF X$ = "Y" THEN 1910; IF X$ = "N" THEN 2090; GOTO 1890
RUN EPPROG
REM STOP THE FOLLOWING ARE SUBROUTINES.
STOP
REM PLACES A NULL FACTOR '#' IN A VARIABLE'S LOCATION.
FOR S = W TO W + (P - 1)
YS(SS) = "0"
NEXT S
RETURN; STOP
REM ENABLES USER TO CHANGE A CHANNEL'S MONTAGE VALUE.
INPUT "CORRECT 'INPUT A' MONTAGE - ", FROM$(1)
XX$(1) = FROM$(1); GOSUB 2060
IF X$(1) = "Y" THEN 2000; GOTO 1970
Y$(N1(1),N2(1)) = FROM$(1)
INPUT "CORRECT 'INPUT B' MONTAGE - ", TO$(1)
XX$(1) = TO$(1); GOSUB 2060
IF X$(1) = "Y" THEN 2040; GOTO 2010
Y$(N3(1),N4(1)) = TO$(1)
RETURN; STOP
REM ENABLES USER TO RESPOND TO A VISUAL CHECK ON A VARIABLES VALUE.
PRINT "IS ", XX$(1), " CORRECT?"
INPUT X$(1)
RETURN; STOP
REM END OF THE PROGRAM
END

OK
LIST EPDATA

10 PUT 0 IN PSI; PUT 8192 IN PSZ; XEQ ZEM
20 DIM DELAY(3), STSEC(3), SWFSZ(7), SWPS(5), CHS(2), STMEM(2), WRISt(6)
30 DIM S$H$:5, DWEEl(5), STPEIt(6), REPTS(5), Y$(70), AN$70)
40 DIM LIEW$(4), WBBAND(5), SENTY(5), STINT(5), THRES(4), SUBNO$(5), VAL$(7)
50 DIM LIEW$(5), %$(1); N$(1) = 0; M$(1) = 0
60 W = 1; SWFSZ(1) = 0; GOTO 70
70 SUBNO$(1) = " "
80 PRINT; INPUT "INPUT THE SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
90 PRINT "IS ", SUBNO$(1), " CORRECT?"
100 INPUT X$
110 IF X$ = "Y" THEN 120; IF X$ = "N" THEN 70
120 BASE(1) = VAL(SUBNO$(3,4))
130 SUB(1) = 0
140 PRINT; INPUT "THE FILE SUBSCRIPT IS (3 #'S 0 - 999) - ", SUB(1)
150 PRINT "IS ", SUB$(1), " CORRECT?"
160 INPUT X$
170 IF X$ = "Y" THEN 180; IF X$ = "N" THEN 130
180 IF SUB(1) < 0 OR SUB(1) > 999 THEN 130
190 SSUB$(1) = SUB$(1)
200 PUT 0 IN PD$
210 PRINT; INPUT "STIMULATING THE RIGHT OR LEFT WRIST? - ", WRISt$(1)
220 Y$(W) = WRISt$(1); P = 6; W = W + P
230 PRINT; INPUT "WHAT IS THE STIMULUS DELAY TIME (IN MS)? - ", DELAY$(1)
240 DELAY$(1) = STR$(DELAY$(1)); "P = 3; W = W + P
250 PRINT; PRINT "USER CANNOT REQUEST MORE THAN 13 STIMULI PER SECOND."
260 INPUT "NUMBER OF STIMULI PER SECOND - ", STSEC(1)
270 IF STSEC(1) < 1 OR STSEC(1) > 13 THEN 250
280 STSEC$(1) = STR$(STSEC$(1)); Y$(W) = STSEC$(1); P = 3; W = W + P
290 STPEIt(1) = 1/STSEC(1); REPTS(1) = STPEIt(1) * 1000
300 REPTS$(1) = STR$(REPTS$(1)); "P = 5; W = W + P
310 PRINT; PRINT "MINIMUM LENGTH OF THE SWEEP (IN MS) - ", SWFSZ$(1)
320 GOSUB 1510
330 STMEM$(1) = "O"; Y$(W) = STMEM$(1); P = 2; W = W + P
340 Y$(W) = SWMEM$(1); P = 5; W = W + P; Y$(W) = LIEW$(1); P = 4
350 W = W + P; Y$(W) = SWFSZ$(1); P = 7; W = W + P
360 GOSUB 1620
370 PUT 2;12 IN LDL
380 PRINT
390 INPUT "WHAT IS THE SUBJECT'S THRESHOLD (IN MILLIAMS)? - ", THRES$(1)
400 Y$(W) = THRES$(1); P = 4; W = W + P; PRINT
410 INPUT "WHAT IS THE STIMULUS INTENSITY (IN MILLIAMS)? - ", STINT$(1)
420 Y$(W) = STINT$(1); P = 5; W = W + P
430 PRINT; PRINT "INPUT THE SENSITIVITY - ", SENTY$(1)
440 IF SENTY$(1) = 1.25 OR SENTY$(1) = 2.5 OR SENTY$(1) = 5 THEN 480
450 IF SENTY$(1) = 12.5 OR SENTY$(1) = 25 OR SENTY$(1) = 50 THEN 480
460 IF SENTY$(1) = 125 OR SENTY$(1) = 250 OR SENTY$(1) = 500 THEN 480
470 GOTO 430
480 SENTY$(1) = STR$(SENTY$(1)); Y$(W) = SENTY$(1); P = 5; W = W + P

8.15
490 PRINT; INPUT "INPUT THE LOW BANDPASS -", LBand$(1)
500 IF LBAND$(1) = 1 OR LBAND$(1) = 5 OR LBAND$(1) = 30 THEN 530
510 IF LBAND$(1) = 150 OR LBAND$(1) = 300 THEN 530
520 GOTO 490
530 LBAND$(1) = STR$(LBAND$(1)); Y$(W) = LBAND$(1); P = 4; W = W + P
540 PRINT; INPUT "INPUT THE HIGH BANDPASS -", HBAND$(1)
550 IF HBAND$(1) = 30 OR HBAND$(1) = 100 OR HBAND$(1) = 250 THEN 580
560 IF HBAND$(1) = 1500 OR HBAND$(1) = 3000 OR HBAND$(1) = 8000 THEN 580
570 GOTO 540
580 HBAND$(1) = STR$(HBAND$(1)); Y$(W) = HBAND$(1); P = 5; W = W + P
590 PRINT; INPUT "NUMBER OF SWEEPS TO BE COLLECTED -", SWPS$(1)
600 SWPS$(1) = STR$(SWPS$(1)); Y$(W) = SWPS$(1); F = 5; W = W + P
610 PUT SWPS$(1) IN FSW; GET SWPS$(1) FROM 6SW
620 PRINT; PRINT "USER CAN REQUEST 1, 2, 4, OR 8 CHANNELS TO BE RECORD."
630 INPUT "NUMBER OF CHANNELS COLLECTED? -", CHS$(1)
640 IF CHS$(1) < 1 OR CHS$(1) > 8 THEN 620
650 IF CHS$(1) = 3 OR CHS$(1) = 5 OR CHS$(1) = 6 OR CHS$(1) = 7 THEN 620
660 PUT CHS$(1) IN FPI; GET CHS$(1) FROM FPI; CHS$(1) = STR$(CHS$(1))
670 Y$(W) = CH$(1); R = 2; W = W + P; GOTO 660
680 PRINT; INPUT "DO YOU NEED A LISTING OF THE PARAMETERS - ", X$
690 IF X$ = "Y" THEN 700; IF X$ = "N" THEN 850; IF X$ = X$ THEN 680
700 PRINT; PRINT "1. WRIST STIMULATION - ", WSR$I$(1)
710 PRINT "2. SUBJECTS THRESHOLD - ", TR$(1)
720 PRINT "3. STIMULUS INTENSITY - ", STIN$(1)
730 PRINT; PRINT "4. STIMULUS DELAY TIME - ", DEL$(1)
740 PRINT "5. STIMULUS PER SECOND - ", STSEC$(1)
750 PRINT "6. LENGTH OF SWEEP - ", SWPS$(1)
760 PRINT "7. WREW TIME - ", RREW$(1)
770 PRINT "8. STIMULUS PERIOD - ", REP$(1)
780 PRINT; PRINT "9. NUMBER OF SWEEPS - ", SWPS$(1)
790 PRINT "10. NUMBER OF CHANNELS - ", CHS$(1)
800 PRINT "11. START OF MEMORY - ", SMEM$(1)
810 PRINT "12. SIZE OF MEMORY - ", SZMEM$(1)
820 PRINT; PRINT "13. AMPLIFIER/Filter SENSITIVITY - ", SENTRY$(1)
830 PRINT "14. AMPLIFIER/Filter LOW BANDPASS - ", LBAND$(1)
840 PRINT "15. AMPLIFIER/Filter HIGH BANDPASS - ", HBAND$(1)
850 PRINT
860 INPUT "ARE ANY OF THE ABOVE PARAMETER VALUES INCORRECT - ", X$
870 IF X$ = "Y" THEN 880; IF X$ = "N" THEN 1200; GOTO 850
880 PRINT; INPUT "GIVE THE # OF THE PARAMETER TO BE CHANGED - ", NO$(1)
890 PRINT "IS ", NO$(1), " CORRECT?"
900 INPUT X$
910 IF X$ = "Y" THEN 920; IF X$ = X$ THEN 880
920 IF NO$(1) = 1 THEN 990; IF NO$(1) = 2 THEN 1000; IF NO$(1) = 3 THEN 1010
930 IF NO$(1) = 4 THEN 1020; IF NO$(1) = 5 THEN 1040; IF NO$(1) = 6 THEN 1080
940 IF NO$(1) = 7 OR NO$(1) = 8 OR NO$(1) = 11 OR NO$(1) = 12 THEN 970
950 IF NO$(1) = 9 THEN 1130; IF NO$(1) = 10 THEN 1150; IF NO$(1) = 13 THEN 1170
960 IF NO$(1) = 14 THEN 1180; IF NO$(1) = 15 THEN 1190; IF NO$(1)=NO$(1) THEN 880
970 PRINT; PRINT "THIS VARIABLE IS NOT UNDER USER CONTROL."
980 GOTO 850
990 N1(1) = 1; N2(1) = 5; GSUS(1)=1750; WSR$(1) = VAL$(1); GOTO 680
1000 M1(1) = 36 ; N2(1) = 38 ; GOSUB 1750 ; THRES$(1) = VAL$(1) ; GOTO 680
1010 M1(1) = 40 ; N2(1) = 43 ; GOSUB 1750 ; STINT$(1) = VAL$(1) ; GOTO 680
1020 M1(1) = 8 ; N2(1) = 9 ; GOSUB 1750 ; DELAY$(1) = VAL$(1);
1030 DELAY(1) = VAL(DELAY$(1)) ; GOTO 680
1040 M1(1) = 11 ; N2(1) = 12 ; GOSUB 1750 ; STSEC$(1) = VAL$(1)
1050 STSEC(1) = VAL(STSEC$(1)) ; STPER(1) = 1/STSEC(1) ; REPTS(1) = STPER(1)*1000
1060 M1(1) = 13 ; N2(1) = 14 ; VAL$(1) = STR$(REPTS(1)) ; GOSUB 1790
1070 REPTS$(1) = VAL$(1) ; GOTO 680
1080 M1(1) = 30 ; N2(1) = 34 ; GOSUB 1750 ; SWPSZ(1) = VAL(VAL$(1))
1090 GOSUB 1510
1100 M1(1) = 20 ; N2(1) = 24 ; VAL$(1) = SIN$(1) ; GOSUB 1790
1110 M1(1) = 25 ; N2(1) = 28 ; VAL$(1) = LLEV$(1) ; GOSUB 1790
1120 M1(1) = 29 ; N2(1) = 34 ; VAL$(1) = SWPSZ$(1) ; GOSUB 1790
1130 M1(1) = 60 ; N2(1) = 63 ; GOSUB 1720 ; SWPS$(1) = VAL$(1)
1140 SWPS$(1) = VAL(SWPS$(1)) ; PUT SWPS$(1) IN PSW ; GET SWPS$(1) FROM GSU ; GOTO 680
1150 M1(1) = 65 ; N2(1) = 65 ; GOSUB 1750 ; CHS$(1) = VAL$(1)
1160 CHS$(1) = VAL(CHS$(1)) ; PUT CHS$(1) IN P1N ; GET CHS$(1) FROM CIT ; GOTO 680
1170 M1(1) = 46 ; N2(1) = 49 ; GOSUB 1750 ; SENT$(1) = VAL$(1) ; GOTO 680
1180 M1(1) = 51 ; N2(1) = 53 ; GOSUB 1750 ; LBand$(1) = VAL$(1) ; GOTO 680
1190 M1(1) = 55 ; N2(1) = 58 ; GOSUB 1750 ; HBand$(1) = VAL$(1) ; GOTO 680
1200 PUT 2 IN PDV ; PUT 208 IN FFH ; PUT BASE(1) IN FFB ; PUT SSUB(1) IN FFS
1210 FOR S = 1 TO W
1220 AS = Y$(S,S) ; DMEN(S) = ASC(AS)
1230 NEXT S
1240 XEQ DUR
1250 XEQ ZEM
1260 PRINT ; GOSUB 1650
1270 PRINT ; PRINT "HIT ANY KEY TO START AVERAGING WAVEFORMS."
1280 GET X FROM AEN ; PRINT
1290 XEQ AVG ; XEQ BSS ; PUT 2 12 IN LDL
1300 PRINT ; PRINT "DO YOU WISH TO STORE THESE WAVEFORMS? " ; Y
1310 IF X$ = "Y" THEN 1320 ; IF X$ = "N" THEN 1330 ; IF X$ = X$ THEN 1300
1320 PUT 196 IN FFH ; XEQ DUR ; SSUB$(1) = SSUB$(1) + 1
1330 PRINT ; PRINT "DO YOU WANT A PLOT OF THE WAVEFORMS AS THEY APPEAR ON THE"
1340 PRINT "SCREEN. NOTE: NO PARAMETER INFORMATION OR SCALES INCLUDED."
1350 INPUT X$
1360 IF X$ = "Y" THEN 1370 ; IF X$ = "N" THEN 1410 ; IF X$ = X$ THEN 1300
1370 PUT 0 IN FFH ; XEQ PLT
1380 PRINT "WILL YOU LIKE ANOTHER PLOT OF THE SAME WAVEFORM? MAKE ANY"  
1390 INPUT " CHANGES BEFORE TYPING A 'Y' RESPONSE. ", X$
1400 IF X$ = "Y" THEN 1370 ; IF X$ = X$ THEN 1410
1410 PRINT ; PRINT "IF ANY PARAMETERS ARE TO BE CHANGED REMEMBER TO WRITE A"  
1420 PRINT " NEW PERSONAL FILE."
1430 PRINT ; INPUT "DO YOU WISH TO CONTINUE WITH THIS PROGRAM - ", X$
1440 IF X$ = "N" THEN 1470 ; IF X$ = "Y" THEN 1450 ; IF X$ = X$ THEN 1410
1450 PRINT ; INPUT "DO YOU NEED TO CHANGE ANY PARAMETER VALUES? - ", X$
1460 IF X$ = "N" THEN 1250 ; IF X$ = "Y" THEN 1480 ; IF X$ = X$ THEN 1470
1470 PRINT ; INPUT "DO YOU WISH TO CONTINUE WITH ANOTHER PROGRAM? - ", X$
1480 IF X$ = "Y" THEN 1490 ; IF X$ = "N" THEN 1800 ; IF X$ = X$ THEN 1470
1490 PUT 1 IN PDV ; RUN EPPROG
1500 STOP
1510 SZMEM(1) = SWPSZ(1) / .04 ; FOR S = 1 TO 14
1520 IF SZMEM(1) < 2'S THEN 1550 ; NEXT S
1530 PRINT ; PRINT "SWEEP LENGTH ERROR - TRY ANOTHER VALUE."
1540 GOTO 310
1550 SZMEM(1) = 2'S
1560 SZMEM$(1) = STR$(SZMEM(1)) ; DWELL(1) = SWPSZ(1) / SZMEM(1) * 1000
1570 IF DWELL(1) < 40 THEN 1580 ; GOTO 1590
1580 DWELL(1) = 40
1590 PUT DWELL(1) IN P$W ; GET DWELL(1) FROM GDW ; LLEWD(1) = DWELL(1) / 1000
1600 LLEWD$(1) = STR$(LLEWD(1)) ; SWPSZ(1) = LLEWD(1) * SZMEM(1)
1610 SWPSZ$(1) = STR$(SWPSZ(1)) ; RETURN
1620 PRINT ; PRINT "FIND SUBJECT'S THRESHOLD AND DETERMINE STIMULUS INTENSITY."
1630 PRINT ; PRINT "EXAMINE THE SIGNAL AND SET THE AMPLIFIER/FILTER'S"
1640 PRINT "SENSITIVITY AND BANDPASS WHILE IN THE VIEW MODE."
1650 PRINT ; PRINT "TYPE 0 WHEN READY TO CONTINUE."
1660 PUT 0 IN PST ; PUT SZMEM(1) IN P$Z
1670 PUT 2'12 IN LIL
1680 PUT 1 IN STH ; PUT REPTS(1) IN LDT
1690 PUT 2 IN STW ; PUT STPER(1) IN LDT
1700 PUT 3 IN STM ; PUT DELAY(1) IN LDT
1710 PUT 2'12 IN CLR
1720 PUT 1 IN UNT
1730 XEO NTR ; XEO VIW
1740 RETURN
1750 VAL$(1) = "" ; PRINT ; INPUT "INPUT THE CORRECT VALUE - ", VAL$(1)
1760 PRINT "IS ", VAL$(1), " CORRECT?"
1770 INPUT X$
1780 IF X$ = '1' THEN 1790 ; IF X$ = X$ THEN 1790
1790 Y$(1) = N$(1) = VAL$(1) ; RETURN
1800 END

OK
LIST DATABU

5 REM ZERO DATA MEMORY
10 PUT 0 IN PST ; PUT 6192 IN PSZ ; XEQ ZEN
15 REM DIMENSION VARIABLES AND ASSIGN VALUES TO SOME VARIABLES.
20 DIN DELAYS(3), STSEC*(3), SUPSZ$(7), SUPSS(5), CHSS(2), STMEM$(2), WRIST*(6)
25 DIN DMEM*(5), DUELLS(5), STSEC*(6), REPTS*(5), Y$(70), A*$ (70)
30 DIN LLEUDS(5), X$(1); N1(1) = 0 ; N2(1) = 0
35 U = 1 ; SUPSZ$(1) = 0 ; GOTO 70
65 REM INPUT SUBJECT'S I.D. NUMBER.
70 SUBNO$(1) = " 
80 PRINT ; INPUT "INPUT THE SUBJECT’S I.D. NUMBER - ", SUBNO$(1)
85 REM VALIDATE VALUE
90 PRINT "IS ", SUBNO$(1), " CORRECT?"
100 INPUT X$
110 IF X$ = "Y" THEN 120 ; IF X$ = X$ THEN 70
115 REM FIND FILE BASE AND FILE HEADER.
120 BASE(1) = VAL(SUBNO$(3,4))
130 SUB(1) = 0
135 REM INPUT FILE SUBSCRIPT
140 PRINT ; INPUT "THE FILE SUBSCRIPT IS (3 #'S 0 - 999) - ", SUB(1)
145 REM VALIDATE VALUE
150 PRINT "IS ", SUB(1), " CORRECT?"
160 INPUT X$
170 IF X$ = "Y" THEN 180 ; IF X$ = X$ THEN 130
175 REM CHECK TO INSURE THAT SUBSCRIPT IS IN THE CORRECT RANGE.
180 IF SUB(1) < 0 OR SUB(1) > 999 THEN 130
185 REM SET VARIABLE EQUAL TO THE FILE SUBSCRIPT
190 SSSUB(1) = SUB(1)
195 REM SET DELAY TIME TO ZERO
200 PUT 0 IN PDY
205 REM INPUT A VARIABLE'S VALUE AND THE PROGRAM WILL PLACE THAT VALUE INTO
210 REM ITS PROPER LOCATION IN THE FILE.
215 PRINT ; INPUT "STIMULATING THE RIGHT OR LEFT WRIST? - ", WRIST*(1)
220 Y$(W) = WRIST*(1) ; P = 6 ; W = W + P
230 PRINT ; INPUT "WHAT IS THE STIMULUS DELAY TIME (IN MS)? - ", DELAY(1)
240 DELAY*(1) = STR$(DELAY(1)) ; Y$(W) = DELAY*(1) ; P = 3 ; W = W + P
250 PRINT ; PRINT "USER CANNOT REQUEST MORE THAN 13 STIMULI PER SECOND."
260 INPUT "NUMBER OF STIMULI PER SECOND - ", STSEC(1)
265 REM CHECK TO INSURE THAT STIMULI PER SECOND IS IN THE CORRECT DATA RANGE.
270 IF STSEC(1) < 1 OR STSEC(1) > 13 THEN 250
275 REM MULTIPLY STIMULUS PERIOD BY 1000 SO IT WILL BE AT THE SAME DATA
280 STSEC*(1) = STR$(STSEC(1)) ; Y$(W) = STSEC*(1) ; P = 3 ; W = W + P
285 REM LEVEL AS THE OTHER DATA.
290 STPER(1) = 1/STSEC(1) ; REPTS*(1) = STPER(1) * 1000
300 REPTS*(1) = STR$(REPTS*(1)) ; Y$(W) = REPTS*(1) ; P = 5 ; W = W + P
310 PRINT ; INPUT "MINIMUM LENGTH OF THE SVEEP (IN MS) - ", SUPSZ$(1)
320 GOSUB 1510
325 REM SET START OF DATA MEMORY TO 0.
STMEMS(1) = "0" ; Y$(U) = STMEMS(1) ; P = 2 ; U = U + P
340 Y$(U) = STMEMS(1) ; P = 5 ; U = U + P ; Y$(U) = LLEND(1) ; P = 4
345 REM TURN STIMULUS OFF
350 U = U + P ; Y$(U) = SWPSZ(1) ; P = 7 ; U = U + P
355 REM TURN STIMULUS ON TO FIND SUBJECT'S THRESHOLD.
360 GOSUB 1620
370 PUT 2`12 IN LDL
380 PRINT
385 REM INPUT A VARIABLE'S VALUE AND THE PROGRAM WILL PLACE THAT VALUE INTO
386 REM ITS PROPER LOCATION IN THE FILE.
390 INPUT "WHAT IS THE SUBJECT'S THRESHOLD (IN MILLIAMPS)? - ", THRES(1)
395 PRINT "Y$(U) = THRES(1); P = 4; U = U + P; PRINT"
400 Y$(U) = THRES(1) ; P = 4 ; U = U + P ; PRINT
410 INPUT "WHAT IS THE STIMULUS INTENSITY (IN MILLIAMPS)? - ", SENTS(1)
415 PRINT ; INPUT "INPUT THE SENSITIVITY - ", SENTS(1)
420 Y$(U) = SENTS(1) ; P = 5 ; U = U + P
425 PRINT ; INPUT "INPUT THE LOW BANDPASS - "; LBAND(1)
430 REM CHECK THE VALUE OF SENSITIVITY.
435 IF SENTS(1) = 1.25 OR SENTS(1) = 2.5 OR SENTS(1) = 5 THEN 480
440 IF SENTS(1) = 12.5 OR SENTS(1) = 25 OR SENTS(1) = 50 THEN 480
445 IF SENTS(1) = 125 OR SENTS(1) = 250 OR SENTS(1) = 500 THEN 480
450 GOTO 430
455 PRINT "SENTS(1) = STR$(SENTS(1)) ; Y$(U) = SENTS(1) ; P = 5 ; U = U + P"
460 PRINT ; INPUT "INPUT THE LOW BANDPASS - "; LBAND(1)
465 REM CHECK THE VALUE OF low BANDPASS.
470 IF LBAND(1) = 1 OR LBAND(1) = 5 OR LBAND(1) = 30 THEN 530
475 IF LBAND(1) = 150 OR LBAND(1) = 300 THEN 530
480 GOTO 490
485 LBAND(1) = STR$(LBAND(1)) ; Y$(U) = LBAND(1) ; P = 4 ; U = U + P
490 PRINT ; INPUT "INPUT THE HIGH BANDPASS - "; HBAND(1)
495 REM CHECK THE VALUE OF HIGH BANDPASS.
500 IF HBAND(1) = 30 OR HBAND(1) = 100 OR HBAND(1) = 250 THEN 580
505 IF HBAND(1) = 1500 OR HBAND(1) = 3000 OR HBAND(1) = 8000 THEN 580
510 GOTO 540
515 HBAND(1) = STR$(HBAND(1)) ; Y$(U) = HBAND(1) ; P = 5 ; U = U + P
520 PRINT ; INPUT "NUMBER OF SWEEPS TO BE COLLECTED - "; SWPS(1)
525 PBSPS(1) = STR$(SWPS(1)) ; Y$(U) = PBSPS(1) ; P = 5 ; U = U + P
530 PUT PBSPS(1) IN PSW ; GET PBSPS(1) FROM GSU
535 PRINT ; PRINT "USER CAN REQUEST 1, 2, 4, OR 8 CHANNELS TO BE RECORDED."
540 INPUT "NUMBER OF CHANNELS COLLECTED? - "; CHS(1)
545 REM CHECK THE RANGE OF THE NUMBER OF CHANNELS.
550 IF CHS(1) < 1 OR CHS(1) > 8 THEN 620
555 IF CHS(1) = 3 OR CHS(1) = 5 OR CHS(1) = 6 OR CHS(1) = 7 THEN 620
560 PUT CHS(1) IN PIN ; GET CHS(1) FROM GIN ; CHS(1) = STR$(CHS(1))
565 Y$(U) = CHS(1) ; P = 2 ; U = U + P ; GOTO 680
570 REM ENABLES USER TO GET A LISTING OF PARAMETERS AND THEIR VALUES.
575 PRINT ; INPUT "DO YOU NEED A LISTING OF THE PARAMETERS - ", X$'
580 IF X$ = "Y" THEN 700 ; IF X$ = "N" THEN 850 ; IF X$ = X$ THEN 680
585 PRINT ; PRINT "1. WRIST STIMULATION - ", WRIST($1)
590 PRINT "2. SUBJECTS THRESHOLD - ", THRES($1)
595 PRINT "3. STIMULUS INTENSITY - ", STIM($1)
600 PRINT "4. STIMULUS DELAY TIME - ", DELAY($1)
605 PRINT "5. STIMULUS PER SECOND - ", STSEC($1)
PRINT "6. LENGTH OF SWEEP - ", SUPSZ$(1)
760 PRINT "7. DUELL TIME - ", LLEUD$(1)
770 PRINT "8. STIMULUS PERIOD - ", REPTS$(1)
780 PRINT ; PRINT "9. NUMBER OF SWEEPS - ", SUPS$(1)
790 PRINT "10. NUMBER OF CHANNELS - ", CHS$(1)
800 PRINT "11. START OF MEMORY - ", SMEM$(1)
810 PRINT "12. SIZE OF MEMORY - ", SZMEM$(1)
820 PRINT ; PRINT "13. AMPLIFIER/FILTER SENSITIVITY - ", SENTS$(1)
830 PRINT "14. AMPLIFIER/FILTER LOW BANDPASS - ", LBAND$(1)
840 PRINT "15. AMPLIFIER/FILTER HIGH BANDPASS - ", HBM$(1)
850 PRINT
855 REM ENABLES USER TO CHANGE THE VALUE OF A VARIABLE.
860 INPUT "ARE ANY OF THE ABOVE PARAMETER VALUES INCORRECT - ", X$
870 IF X$ = "Y" THEN 880 ; IF X$ = "N" THEN 1200 ; GOTO 850
880 PRINT ; PRINT "GIVE THE # OF THE PARAMETER TO BE CHANGED - ", NO(1)
890 PRINT "IS ", NO(1), " CORRECT?"
900 INPUT X$
910 IF X$ = "Y" THEN 920 ; IF X$ = X$ THEN 880
920 IF NO(1) = 1 THEN 990 ; IF NO(1) = 2 THEN 1000 ; IF NO(1) = 3 THEN 1010
930 IF NO(1) = 4 THEN 1020 ; IF NO(1) = 5 THEN 1040 ; IF NO(1) = 6 THEN 1080
940 IF NO(1) = 7 OR NO(1) = 8 OR NO(1) = 11 OR NO(1) = 12 THEN 970
950 IF NO(1) = 9 THEN 1130 ; IF NO(1) = 10 THEN 1150 ; IF NO(1) = 13 THEN 1170
960 IF NO(1) = 14 THEN 1180 ; IF NO(1) = 15 THEN 1190 ; IF NO(1)=NO(1) THEN 880
970 PRINT ; PRINT "THIS VARIABLE IS NOT UNDER USER CONTROL."
980 GOTO 850
990 N1(1) = 1 ; N2(1) = 5 ; GOSUB 1750 ; WRIST$(1) = VAL$(1) ; GOTO 680
1000 N1(1) = 36 ; N2(1) = 38 ; GOSUB 1750 ; THRES$(1) = VAL$(1) ; GOTO 680
1010 N1(1) = 40 ; N2(1) = 43 ; GOSUB 1750 ; STINT$(1) = VAL$(1) ; GOTO 680
1020 N1(1) = 8 ; N2(1) = 9 ; GOSUB 1750 ; DELAY$(1) = VAL$(1)
1030 DELAY$(1) = VAL$(DELAYS$(1)) ; GOTO 680
1040 N1(1) = 11 ; N2(1) = 12 ; GOSUB 1750 ; STSEC$(1) = VAL$(1)
1050 STSEC$(1) = VAL$(STSEC$(1)) ; STPER$(1) = STSEC$(1)/1000
1060 REPTS$(1) = 13 ; N2(1) = 14 ; VAL$(1) = STR$(REPTS$(1)) ; GOSUB 1790
1070 REPTS$(1) = VAL$(1) ; GOTO 680
1080 N1(1) = 30 ; N2(1) = 34 ; GOSUB 1750 ; SUPSZ$(1) = VAL$(VAL$(1))
1090 GOSUB 1510
1100 N1(1) = 20 ; N2(1) = 24 ; VAL$(1) = SZMEM$(1) ; GOSUB 1790
1110 N1(1) = 25 ; N2(1) = 28 ; VAL$(1) = LLEUD$(1) ; GOSUB 1790
1120 N1(1) = 29 ; N2(1) = 34 ; VAL$(1) = SUPSZ$(1) ; GOSUB 1790
1130 N1(1) = 60 ; N2(1) = 63 ; GOSUB 1720 ; SUPS$(1) = VAL$(1)
1140 SUPS$(1)=VAL(SUPS$(1)); PUT SUPS$(1) IN PSW ; GET SUPS$(1) FROM GSU ; GOTO 680
1150 N1(1) = 65 ; N2(1) = 65 ; GOSUB 1750 ; CHS$(1) = VAL$(1)
1160 CHS$(1)=VAL(CHS$(1)); PUT CHS$(1) IN PIN ; GET CHS$(1) FROM GIN; GOTO 680
1170 N1(1) = 46 ; N2(1) = 49 ; GOSUB 1750 ; SENTS$(1) = VAL$(1) ; GOTO 680
1180 N1(1) = 51 ; N2(1) = 53 ; GOSUB 1750 ; LBAND$(1) = VAL$(1) ; GOTO 680
1190 N1(1) = 55 ; N2(1) = 58 ; GOSUB 1750 ; HBAND$(1) = VAL$(1) ; GOTO 680
1195 REM SELECT DISK DRIVE # 2 AND THE FILENAME FOR THE PARAMETER FILE.
1200 PUT 2 IN PIN ; PUT 200 IN PHF ; PUT BASE$(1) IN PFH ; PUT SSUB$(1) IN PFS
1205 REM TRANSLATE PARAMETER DATA FILE INTO ASCII CODE.
1210 FOR S = 1 TO W
1220 AS = Y$(S,S) ; DNEM$(S) = ASC(AS)
REM SAVE THE PARAMETER DATA FILE ON DISK.
XEQ DUR
REM CLEAR MEMORY.
XEQ ZEN
REM TURN STIMULUS ON.
PRINT ; GOSUB 1650
PRINT "HIT ANY KEY TO START AVERAGING WAVEFORMS."
GET X FROM AEN ; PRINT
REM AVERAGE E.P.'S, BASELINE ZERO, AND TURN STIMULUS OFF.
XEQ AVG ; XEQ DSC ; PUT 2'12 IN LDL
REM ENABLES USER TO STORE WAVEFORMS ON DISK.
PRINT ; PRINT "DO YOU WANT TO STORE THESE WAVEFORMS? - ", X$
IF X$ = "Y" THEN 1320 ; IF X$ = "N" THEN 1300
REM STORE WAVEFORMS AND INCREMENT THE SUBSCRIPT BY 1.
PUT 196 IN PFH ; XEQ DUR ; SSUB(1) = SSUB(1) + 1
REM ENABLES USER TO PLOT EVOKED POTENTIALS.
PRINT ; PRINT "DO YOU WANT A PLOT OF THE WAVEFORMS AS THEY APPEAR ON THE"
PRINT "SCREEN. NOTE: NO PARAMETER INFORMATION OR SCALES INCLUDED."
INPUT X$
IF X$ = "Y" THEN 1370 ; IF X$ = "N" THEN 1410 ; IF X$ = X$ THEN 1330
PUT 0 IN PPM ; XEQ PLT
REM ENABLES USER TO GET A SECOND PLOT OF THE EVOKED POTENTIALS.
PRINT "WOULD YOU LIKE ANOTHER PLOT OF THE SAME WAVEFORM? MAKE ANY"
INPUT X$ ; IF X$ = "Y" THEN 1370 ; IF X$ = X$ THEN 1410
PRINT ; PRINT "IF ANY PARAMETERS ARE TO BE CHANGED REMEMBER TO WRITE A"
PRINT "NEW PERSONAL FILE."
REM ENABLES USER TO RERUN PROGRAM.
INPUT "DO YOU NEED TO CHANGE ANY PARAMETER VALUES? - ", X$
IF X$ = "Y" THEN 1450 ; IF X$ = "N" THEN 1440 ; IF X$ = X$ THEN 1410
REM RERUN THE PROGRAM BUT CHANGE SOME PARAMETER DATA.
PRINT ; INPUT "DO YOU WISH TO CONTINUE WITH THIS PROGRAM? - ", X$
IF X$ = "Y" THEN 1470 ; IF X$ = "N" THEN 1460 ; IF X$ = X$ THEN 1440
REM RERUN THE PROGRAM BUT CHANGE SOME PARAMETER DATA.
PRINT "DO YOU WISH TO CONTINUE WITH ANOTHER PROGRAM? - ", X$
IF X$ = "Y" THEN 1490 ; IF X$ = "N" THEN 1480 ; IF X$ = X$ THEN 1470
PUT 1 IN PDV ; RUN EPPROG
REM STOP MAIN PROGRAM, THE FOLLOWING ARE SUBROUTINES.
STOP
REM SUBROUTINE TO DETERMINE VALUE OF SWEELength, SIZE OF DMEN, AND DWELL.
SZMEM(1) = SUPSZ(1) / .04 ; FOR S = 1 TO 14
REM INSURES THAT THE SIZE OF DMEN IS A FACTOR OF 2.
IF SZMEM(1) < 2'S THEN 1550 ; NEXT S
PRINT ; PRINT "Sweep Length Error - Try Another Value."
GOTO 310
SZMEM(1) = 2'S
SZMEM(1) = STR$(SZMEM(1)) ; DUEL(1) = SUPSZ(1) / SZMEM(1) * 1000
REM IF DWELL IS LESS THAN 40 SET DWELL EQUAL TO 40.
IF DUEL(1) < 40 THEN 1580 ; GOTO 1590
DUEL(1) = 40
PUT DUELL(1) IN PDU; GET DUELL(1) FROM GDU; LLEWD(1) = DUELL(1) / 1000
LLEWD$(1) = STR$(LLEWD(1)); SWPSZ(1) = LLEWD(1) * SZMEM(1)
SWPSZ$(1) = STR$(SWPSZ(1)); RETURN
REM SUBROUTINE TO TURN ON STIMULUS.
PRINT ; PRINT "FIND SUBJECT'S THRESHOLD AND DETERMINE STIMULUS INTENSITY."
PRINT ; PRINT "EXAMINE THE SIGNAL AND SET THE AMPLIFIER/FILTER'S"
PRINT ; PRINT "SENSITIVITY AND BANDPASS WHILE IN THE VIEW MODE."
PRINT ; PRINT "TYPE Q WHEN READY TO CONTINUE."
PUT 0 IN PST ; PUT SZMEM(1) IN PSZ
PUT 2'12 IN LDL
REM INPUT PARAMETERS FOR THE STIMULUS THROUGH NIC-293.
PUT 1 IN STN ; PUT REPTS(1) IN LDT
PUT 2 IN STN ; PUT STPER(1) IN LDT
PUT 3 IN STN ; PUT DELAY(1) IN LDT
REM TURN THE STIMULUS ON.
PUT 2'12 IN CLR
PUT 1 IN ONT
REM SUBROUTINE TO CHANGE A VARIABLE'S VALUE.
XEQ NTR; XEQ VIU
RETURN
VALS(1) = " " ; PRINT ; INPUT "INPUT THE CORRECT VALUE - ", VAL$(1)
PRINT "IS ", VAL$(1), " CORRECT?"
INPUT X$
IF X$ = "Y" THEN 1790 ; IF X$ =X$ THEN 1750
Y$(N1(1),N2(1)) = VAL$(1); RETURN
REM END OF THE PROGRAM.
END

OK
LIST PLOTEP

10 DIM A$(9), B$(6), C$(6), D$(16), E$(17), F$(30), G$(10), H$(7), I$(20), J$(16), K$(20)
20 DIM L$(18), M$(1), N$(1), W$(18), X$(5), Y$(225), C$(5), C$(5), C$(5), C$(5), C$(5)
30 DIM C$(5), C$(5), C$(5), Y$(66), SUBNOS$(4), SUB$(3), GAS$(11)
40 DIM GAIN$(6), NOS$(15), FN$(10), SUB$(4), NOFS$(7)
50 B$="DATE: "; C$="NAME: "; D$="SUBJECT NUMBER: "; G$="(IN MSEC.) "; H$="(IN UV)"
60 E$="SLED RUN NUMBER: "; F$="SONATOSENSORY EVOKED POTENTIAL "; I$="DELAY TIME: "; J$="STIMULI/SECOND: "; K$="DUELL TIME (IN US): "; L$="SLEEP LENGTH: "; M$="21 .. 6"
70 GA$="PLOT GAIN: "; NOS$="NUMBER SLEEP: "; FNS$="FILENAME: "; N$=-.3; C$=-.9; CH$(1)=O
80 US$="URIST STIMULATED: "; C1$="CH.#1" ; C2$="CH.#2" ; C3$="CH.13"; C4$="CH.#4" ; C5$="CH.#5" ; C6$="CH.6l" ; C7$="CH.7" ; C8$="CH.N8"
90 X$(1)=0 ; D$(1)=0 ; N1$(1)=0 ; N2$(1)=0 ; N3$(1)=0 ; N4$(1)=0
100 BASE$(1)=VAL(SUBNO$(3,4)) ; NEAD$(1)=ASC(SUBNOS$(1))
110 PRINT "IS ", SUB$(1); " CORRECT?"
120 X$(1)=0 ; D$(1)=0 ; N1$(1)=0 ; N2$(1)=0 ; N3$(1)=0 ; N4$(1)=0
130 PUT 0 IN PST ; PUT 1024 IN PSZ ; PUT 2 IN PDV
140 PRINT "IS ", SUB$(1); " CORRECT?"
150 INPUT X$
160 IF X$="Y" THEN 170 ; GOTO 130
170 BASE$(1)=VAL(SUBNO$(3,4)) ; NEAD$(1)=ASC(SUBNOS$(1))
180 PRINT "IS ", SUB$(1); " CORRECT?"
190 INPUT X$
200 IF X$="Y" THEN 210 ; GOTO 180
210 IF SUB$(1)<0 OR SUB$(1)>999 THEN 180
220 IF SUB$(1)<0 OR SUB$(1)>999 THEN 180
230 IF HEAD$(1)=40 THEN 240 ; IF HEAD$(1)=50 THEN 250
240 PUT 200 IN PFH ; PUT BASE$(1) IN PFB ; PUT SUB$(1) IN PFS ; GOTO 260
250 PUT 210 IN PFH ; PUT BASE$(1) IN PFB ; PUT SUB$(1) IN PFS
260 XEO DRD
270 FOR S = 1 TO 225
280 U = DMEM$(S) ; V$(S,S)=CHR$(U) ; NEXT S
290 PUT 208 IN PFH ; XEO DRD
300 FOR S = 1 TO 66
310 UU = DMEM$(S) ; V$(S,S)=CHR$(U) ; NEXT S
320 HS$(1)=VAL(Y$(30,34)); OS$(1)=V$(143); CH$(1)=VAL(D$(1)); HR$(1)=HS$(1)/10
330 PRINT "WHAT IS THE MIN. VOLTAGE LEVEL (A NEG. #) - ", VA$(1)
340 PRINT "IS ", VA$(1); " CORRECT?"
350 INPUT X$
360 IF X$="Y" THEN 370 ; GOTO 330
370 VR$(1)=-VA$(1)*2/10
380 INPUT "INPUT GAIN FOR THE PLOTTING OF E.P. WAVEFORMS - ", GAIN$(1)
390 PRINT "IS ", GAIN$(1); " CORRECT?"
400 INPUT X$
410 IF X$="Y" THEN 420 ; GOTO 380
420 XEO DEF ; PUT 1.0 IN ROT
430 SUB$(1)="STR$(SUB$(1))"
440 NOFS$(1)=SUBNO$(1,1) ; NOFS$(2,3)=SUBNO$(3,4) ; NOFS$(4,6)=SUB$(2,4)
450 IF OS$(1)="1" THEN 490 ; IF OS$(1)="2" THEN 720
460 IF OS$(1)="3" THEN 510 ; IF OS$(1)="4" THEN 680
470 IF OS$(1)="5" THEN 540 ; IF OS$(1)="6" THEN 640
480 IF OS$(1)="7" THEN 570 ; IF OS$(1)="8" THEN 660

8.24
490  D(1)=.2; CH$(1)=C1%; N1(1)=215; N2(1)=219; N3(1)=220; N4(1)=224
500  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 750
510  SIZE 0,-8.5 ; VECT PENUP, 1,-1 ; XEO DEF ; PUT 1.0 IN ROT
520  D(1)=.2; CH$(1)=C3%; N1(1)=195; N2(1)=199; N3(1)=200; N4(1)=204
530  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 810
540  SIZE 0,-17 ; VECT PENUP, 1,-1 ; XEO DEF
550  D(1)=.2; CH$(1)=C5%; N1(1)=175; N2(1)=175; N3(1)=180; N4(1)=184
560  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 690
570  SIZE 0,-25.5 ; VECT PENUP, 1,-1 ; XEO DEF
580  D(1)=.2; CH$(1)=C7%; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
590  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 650
600  SIZE 0,-25.5 ; VECT PENUP, 1,-1 ; XEO DEF
610  D(1)=.2; CH$(1)=C9%; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
620  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 690
630  SIZE 0,-8.5 ; VECT PENUP, 1,-1 ; XEO DEF ; PUT 1.0 IN ROT
640  D(1)=.2; CH$(1)=C11%; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
650  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 650
660  SIZE 0,-17 ; VECT PENUP, 1,-1 ; XEO DEF
670  D(1)=.2; CH$(1)=C13%; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
680  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 690
690  SIZE 0,-34 ; VECT PENUP, 1,-1 ; XEO DEF ; PUT 1.0 IN ROT
700  D(1)=.2; CH$(1)=C15%; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
710  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 690
720  SIZE 0,-34 ; VECT PENUP, 1,-1 ; XEO DEF ; PUT 1.0 IN ROT
730  D(1)=.2; CH$(1)=C17%; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
740  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 690
750  SIZE 0,-34 ; VECT PENUP, 1,-1 ; XEO DEF
760  D(1)=.2; CH$(1)=C19%; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
770  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 690
780  SIZE 0,-34 ; VECT PENUP, 1,-1 ; XEO DEF
790  D(1)=.2; CH$(1)=C21%; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
800  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 690
810  SIZE 0,-34 ; VECT PENUP, 1,-1 ; XEO DEF
820  D(1)=.2; CH$(1)=C23%; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
830  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 690
840  SIZE 0,-34 ; VECT PENUP, 1,-1 ; XEO DEF
850  GOTO 880
860  SIZE 0,-34 ; VECT PENUP, 1,-1 ; XEO DEF
870  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 890
880  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 900
890  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 910
900  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 920
910  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 930
920  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 940
930  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 950
940  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 960
950  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 970
960  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 980
970  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 990
980  SIZE 0,-34 ; VECT PENUP, 1,-1 ; GOTO 8.25
1000 SIZE 0,-17 ; VECT PENUP, 1,-1
1010 XEO DEF ; SIZE -1,-.8 ; VECT PENUP, 1,1
1020 PUT PLTSZ(1)* 3 IN PST ; SIZE -9,-3 ; XEO DPL
1030 GOTO 1060
1040 SIZE 0,-17 ; VECT PENUP, 1,-1 ; XEO DEF
1050 SIZE -1,-.8 ; VECT PENUP, 1,1 ; GOTO 1070
1060 SIZE -1,-4 ; VECT PENUP, 1,1
1070 PUT PLTSZ(1) + 2 IN PST ; SIZE -9,-3 ; XEO DPL
1080 SIZE 0,-8.5 ; VECT PENUP, 1,1 ; GOTO 1100
1090 SIZE 0,-8.5 ; VECT PENUP, 1,-1
1100 XEO DEF ; SIZE -1,-.8 ; VECT PENUP, 1,1
1110 PUT PLTSZ(1) IN PST ; SIZE -9,-3 ; XEO DPL
1120 GOTO 1150
1130 SIZE 0,-8.5 ; VECT PENUP, 1,-1 ; XEO DEF
1140 SIZE -1,-8 ; VECT PENUP, 1,1 ; GOTO 1160
1150 SIZE -1,-.4 ; VECT PENUP, 1,1
1160 PUT 0 IN PST ; SIZE -9,-3 ; XEO DPL
1170 VECT PENUP, 0,0
1180 PRINT ; INPUT "DO YOU WISH TO PLOT ANOTHER FILE? - ", X$
1190 IF X$ = "Y" THEN 1210 ; IF X$ = "N" THEN 1310 ; GOTO 1180
1200 PRINT ; INPUT "IS THE FILE IN THIS SERIES - ", X$
1210 IF X$ = "Y" THEN 1230 ; IF X$ = "N" THEN 1220 ; GOTO 1180
1220 GOSUB 2100 ; GOTO 120
1230 INPUT "IS THE FILE NEXT IN THIS SEQUENCE? - ", X$
1240 IF X$ = "Y" THEN 1250 ; IF X$ = "N" THEN 1260 ; GOTO 1230
1250 SUB(1) = SUB(1) + 1 ; GOSUB 2100 ; GOTO 420
1260 INPUT "INPUT THE FULL SUBSCRIPT OF THE NEW FILE - ", SUB(1)
1270 PRINT "IS ", SUB(1), " CORRECT?"
1280 INPUT X$
1290 IF X$ = "Y" THEN 1300 ; GOTO 1260
1300 GOSUB 2100 ; GOTO 420
1310 PRINT ; INPUT "DO YOU WANT TO CONTINUE WITH ANOTHER PROGRAM? - ", X$
1320 IF X$ = "Y" THEN 1330 ; IF X$ = "N" THEN 2190 ; GOTO 1210
1330 RUN EPPROG
1335 STOP
1340 SIZE -1.2,-D(1) ; VECT PENUP, 1,1 ; PLOT -.1, CH$(1)
1350 SIZE -2.0,-D(1) ; VECT PENUP, 1,1 ; PLOT -.1, V$(N1(1),N2(1))
1360 SIZE -2.4,-D(1) ; VECT PENUP, 1,1 ; PLOT -.1, V$(1)
1370 SIZE -2.6,-D(1) ; VECT PENUP, 1,1 ; PLOT -.1, V$(N3(1),N4(1))
1380 RETURN ; STOP
1390 VECT PENUP, 0,0 ; PUT 2.0 IN ROT ; SIZE -3,-2 ; VECT PENUP, 1,-1
1400 PLOT -.1, M$
1410 SIZE -8.5,0 ; VECT PENUP, 1,-1 ; XEO DEF ; PUT 1.0 IN ROT
1420 RETURN ; STOP
1430 XX = -10 ; SIZE -1,-3.8 ; VECT PENUP, 1,1
1440 SIZE XX,-3.8 ; VECT PENUP, 1,1 ; FOR S = 0 TO 10
1450 SIZE XX,-3.7 ; VECT PENUP, 1,1 ; SIZE XX,-3.9 ; VECT PENUP, 1,1
1460 XX = XX - C ; NEXT S
1470 SIZE -2.4,-B.2 ; VECT PENUP, 1,1 ; PLOT -.2, F$
1480 SIZE -.5,-7.9 ; VECT PENUP, 1,1 ; PLOT -.1, C$
1490 SIZE -1.2,-7.9 ; VECT PENUP, 1,1 ; PLOT -.1, V$(1,34)

8.26
1500 SIZE -.5,-7.7 ; VECT PENUP, 1,1 ; PLOT -.1, FM$
1510 SIZE -1.7,-7.7 ; VECT PENUP, 1,1 ; PLOT -.1, NOF$(1)$
1520 SIZE -.5,-7.5 ; VECT PENUP, 1,1 ; PLOT -.1, D$
1530 SIZE -2.2,-7.5 ; VECT PENUP, 1,1 ; PLOT -.1, V$(35,39)$
1540 SIZE -.5,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, E$
1550 SIZE -2.3,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, V$(56,61)$
1560 SIZE -4.75,-7.9 ; VECT PENUP, 1,1 ; PLOT -.1, BS$
1570 SIZE -5.7,-7.9 ; VECT PENUP, 1,1 ; PLOT -.1, V$(50,55)$
1580 SIZE -4.75,-7.7 ; VECT PENUP, 1,1 ; PLOT -.1, VS$(35,39)$
1590 SIZE -2.3,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, VS$(56,61)$
1600 SIZE -4.75,-7.5 ; VECT PENUP, 1,1 ; PLOT -.1, V6$(50,55)$
1610 SIZE -6.6,-7.5 ; VECT PENUP, 1,1 ; PLOT -.1, V$(1,5)$
1620 SIZE -4.75,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, IS$
1630 SIZE -5.95,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, VS$(41,48)$
1640 SIZE -8,-7.9 VECT PENUP, 1,1 ; PLOT -.1, Y$(8,9)$
1650 SIZE -8.6,-3.85 ; VECT PENUP, 1,1 ; PLOT -.1, G5$
1660 SIZE -10,-.8 ; VECT PENUP, 1,1 ; PLOT -.1, GM$
1670 XX = -1 ; UHA(1) = 0 ; FOR S = 0 TO 10
1680 SIZE XX,-.7 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1690 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1700 SIZE XX,-.35 ; VECT PENUP, 1,1 ; PLOT -.1, IS$
1710 SIZE -1,-.8 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1720 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1730 SIZE XX,-.35 ; VECT PENUP, 1,1 ; PLOT -.1, G5$
1740 SIZE XX,-.7 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1750 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1760 SIZE XX,-.9 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1770 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1780 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1790 SIZE -.9,XX ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1800 SIZE -.9,XX ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1810 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1820 SIZE XX,-.9 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1830 SIZE XX,-.9 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1840 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1850 SIZE XX,-.9 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1860 SIZE XX,-.9 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1870 SIZE XX,-.9 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1880 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1890 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1900 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1910 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1920 RETURN ; STOP
1930 SIZE XX,-.9 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1940 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1950 SIZE XX,-.9 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1960 SIZE XX,-.9 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1970 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1980 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1990 XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
2000 XX = XX,-.9 ; VECT PENUP, 1,1 ; SIZE XX,-.7 ; VECT PENDN, 1,1

8.27
2010 XX = XX - C; NEXT S
2020 SIZE -1, -5.5; VECT PENUP, 1, 1; P = -.1; Z = -.1; FOR S = 1 TO 45
2030 SIZE Z, -5.5; VECT PENDN, 1, 1; Z = Z + P
2040 SIZE Z, -5.5; VECT PENUP, 1, 1; Z = Z + P; NEXT S
2050 SIZE -10, -4; VECT PENUP, 1, 1; SIZE -10, -7; VECT PENDN, 1, 1
2060 XX = -.7; FOR S = 0 TO 10
2070 SIZE -9.9, XX; VECT PENUP, 1, 1; SIZE -10.1, XX; VECT PENDN, 1, 1
2080 XX = XX - M; NEXT S
2090 RETURN; STOP
2100 IF $1(1) = "1" OR $1(1) = "2" THEN 2140
2110 IF $1(1) = "3" OR $1(1) = "4" THEN 2150
2120 IF $1(1) = "5" OR $1(1) = "6" THEN 2160
2130 IF $1(1) = "7" OR $1(1) = "8" THEN 2170
2140 SIZE 0, -8.5; VECT PENUP, 1, 1; GOTO 2180
2150 SIZE 0, -17; VECT PENUP, 1, 1; GOTO 2180
2160 SIZE 0, -25.5; VECT PENUP, 1, 1; GOTO 2180
2170 SIZE 0, -34; VECT PENUP, 1, 1
2180 RETURN; STOP
2190 END

OK
LIST PLOTBU

5 REM (10 - 110) - DIMENSION VARIABLES AND ASSIGN VALUES TO SOME VARIABLES
115 REM (120) - SELECT START AND SIZE OF DMEM AND THE DISK DRIVE TO BE USED.
116 REM - ALSO REPLACES FILES BACK INTO THE SAME LOCATION AS FOUND.
125 REM (130 - 160) - INPUT THE I.D. NUMBER OF THE SUBJECTS WHOSE FILE IS
126 REM TO BE PLOTTED. USER IS ASKED TO VALIDATE VARIABLE'S VALUE.
165 REM (170) - THE FILE BASE IS THE LAST TWO #'S OF THE SUBJECT'S I.D. NUMBER.
166 REM THE FILE HEADER IS THE FIRST DIGIT OF THE SUBJECT'S I.D. NUMBER.
175 REM (180 - 220) - INPUT THE FILE SUBSCRIPT. USER IS ASKED TO VALIDATE
176 REM VARIABLE'S VALUE. FOLLOWED BY A CHECK ON VALUES RANGE.
225 REM (230 - 250) - THE FILE HEADER IS CHECKED TO DETERMINE IF THE SUBJECT
226 REM WAS HUMAN OR A RHESUS. THE CORRECT RESPONSE IS THEN PLACED INTO THE
227 REM FILE HEADER. THE BASE AND SUBSCRIPT NUMBER ARE ALSO SET.
255 REM (260) - THE FILE SELECTED IS READ FROM DISK TO DMEM.
265 REM (270 - 280) - THE SUBJECT'S PERSONAL FILE IS READ AND CHANGED FROM
26 REM ASCII CODE TO NORMAL CHARACTERS.
285 REM (290 - 310) - THE SUBJECT'S PARAMETER FILE IS READ FROM DISK TO DMEM AND
286 REM CHANGES FROM ASCII CODE TO NORMAL CHARACTERS.
315 REM (320) - THE SWEEP LENGTH AND THE NUMBER OF CHANNELS RECORDED ARE
316 REM TAKEN FROM THE PARAMETER FILE AND THE PERSONAL FILE RESPECTFULLY.
317 REM THE VARIABLE WITH THE CHANNEL INFORMATION IS ALSO PLACED INTO A
318 REM NON-STRING VARIABLE. THE TIME SCALE INCERIMENTATION IS DETERMINED.
325 REM (330 - 370) - INPUT THE MIN. VOLTAGE LEVEL OF THE WAVEFORM. USER IS
326 REM ASKED TO VALIDATE VARIABLE'S VALUE. THE VOLTAGE INCERIMENTATION IS
327 REM DETERMINED.
375 REM (380 - 410) - INPUT THE GAIN TO BE USED IN PLOTTING. USER IS ASKED TO
376 REM VALIDATE THE VARIABLE'S VALUE.
415 REM (420) - SETTING THE POINT OF ORIGIN FOR THE ZETA PLOTTER TO THE
416 REM PRESENT PEN LOCATION AND A PEN ROTATION OF 1.0
425 REM (430) - SETTING A STRING VARIABLE EQUAL TO THE SUBJECT NUMBER.
435 REM (440) - FIND FILENAME OF THE WAVEFORMS TO BE PLOTTED.
445 REM (450 - 480) - CONDITIONAL BRANCH FOR PLOTTING OUT THE CORRECT
446 REM NUMBER OF CHANNELS.
485 REM (485 - 810) - PLOTS OUT THE ENCLOSURE FOR THE WAVEFORM AND HEADER INFO.
815 REM (820 - 1170) - PLOTS OUT THE INDIVIDUAL EVOKED POTENTIAL WAVEFORM IN
816 REM IT'S ENCLOSURE.
1175 REM (1180 - 1190) - ENABLES USER TO PLOT ANOTHER FILE.
1195 REM (1200 - 1220) - ASKS IF FILE IS IN THIS SERIES.
1225 REM (1230 - 1240) - CHECKS TO SEE IF FILE IS NEXT IN SEQUENCE.
1245 REM (1250) - INCREASES THE SUBSCRIPT BY 1. ADVANCES GRAPH PAPER, AND
1246 REM RERUNS THE PROGRAM.
1255 REM (1260 - 1300) - IF FILE IS NOT IN SEQUENCE OR IN THIS SERIES INPUT
1256 REM THE CORRECT FILE SUBSCRIPT. ADVANCE GRAPH PAPER AND RERUN PROG.
1305 REM (1310 - 1330) - ENABLES USER TO EXIT PROGRAM OR RUN ANOTHER PROGRAM
1335 REM END OF THE MAIN PROGRAM AND START OF SUBROUTINES.
1345 REM END OF THE MAIN PROGRAM AND START OF SUBROUTINES
1346 REM (1350 - 1380) - PLOTTING OF THE CHANNEL NUMBER AND ITS MONTAGE INFO.
1385 REM (1390 - 1420) - PLOTS OUT THE HEADER 'IN UV)' AT A RIGHT ANGLE TO
1386 REM THE OTHER HEADER INFORMATION.

8.29
1425 REM (1430 - 1460) - IF A ODD NUMBER OF CHANNELS ARE TO BE PLOTTED THIS
1426 REM PLOTS OUT THE TOP HORIZONTAL LINE AND INCRIMENTS THE TICK MARKS FOR
1427 REM THE LAST CHANNEL.
1465 REM (1470 - 1730) - PLOTS OUT THE HEADER INFORMATION.
1736 REM INCORRECTED TICK MARKS.
1785 REM (1790) - PLOTS OUT THE HEADER INFORMATION 'IN MSEC'.
1795 REM (1800 - 1840) - PLOTS OUT THE INCORRECTED TIME SCALE AND ITS TICK
1796 REM MARKS.
1845 REM (1850 - 1870) - PLOTS OUT THE CENTER LINE FOR THE LOWER ENCLOSURE.
1875 REM (1880 - 1920) - PLOTS OUT THE RIGHT VERTICAL LINE AND ITS
1876 REM INCORRECTED TICK MARKS FOR THE LOWER RIGHT VERTICAL LINE.
1925 REM (1930 - 1970) - PLOTS OUT THE LEFT VERTICAL LINE WITH ITS
1926 REM INCORRECTED VOLTAGE SCALE AND THE TICK MARKS.
1975 REM (1980 - 2010) - PLOTS OUT THE TOP HORIZONTAL LINE FOR THE UPPER
1976 REM ENCLOSURE.
2015 REM (2020 - 2040) - PLOTS OUT THE CENTER LINE FOR THE UPPER ENCLOSURE.
2045 REM (2050 - 2090) - PLOTS OUT THE UPPER RIGHT VERTICAL LINE AND ITS TICK
2046 REM MARKS.
2095 REM (2100 - 2180) - ADVANCES THE PLOTTING PAPER OF THE ZETA PLOTTER FOR THE
2096 REM PLOTTING OF ANOTHER EVOKED POTENTIAL FILE.
2185 REM END OF THE PROGRAM
9999 END

$ OK
LIST LSTPER
*
10 DIM Y$(225), V$(66), SUBNO$(4)
20 PUT 0 IN PST; PUT 1024 IN PSZ; PUT 2 IN PDU
30 PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT
40 PRINT; INPUT "WHAT IS THE SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
50 PRINT "IS ", SUBNO$(1), " CORRECT?"
60 INPUT X$
70 IF X$ = "Y" THEN 80 ; IF X$ = X$ THEN 30
80 BASE(1) = VAL(SUBNO$(3,4)); HEAD(1) = ASC(SUBNO$(1))
90 PRINT; INPUT "WHAT IS THE FILE SUBSCRIPT (3 #'S 0 - 999) - ", SUB(1)
100 PRINT "IS ", SUB(1), " CORRECT?"
110 INPUT X$
120 IF X$ = "Y" THEN 130 ; IF X$ = X$ THEN 90
130 IF SUB(1) < 0 OR SUB(1) > 999 THEN 90
140 IF HEAD(1) = 40 THEN 150 ; IF HEAD(1) = 50 THEN 160
150 PUT 200 IN PFH; PUT BASE(1) IN PFB; PUT SUB(1) IN PFS; GOTO 170
160 PUT 210 IN PFH; PUT BASE(1) IN PFB; PUT SUB(1) IN PFS
170 XEO DRD
180 FOR S = 1 TO 225
190 X = DMEM(S); Y$(S,S) = CHR$(X)
200 NEXT S
210 XEO DRD
220 FOR S = 1 TO 66
230 X = DMEM(S); V$(S,S) = CHR$(X)
240 NEXT S
250 PRINT "TURN PAPER TO THE TOP OF A FRESH PAGE SO THAT ALL"
260 PRINT "DATA WILL BE PRINTED ON IT'S OWN PAGE."
270 PRINT "AFTER FIXING PAPER TYPE Y TO CONTINUE. ", X$
280 IF X$ = "Y" THEN 300 ; IF X$ = X$ THEN 260
290 PRINT; PRINT; PRINT; PRINT "SUBJECT'S NAME - ", Y$(1,34)
300 PRINT; PRINT "SUBJECT'S I.D. NUMBER - ", Y$(35,39)
310 PRINT; PRINT "DATE OF EXPERIMENT - ", Y$(41,48)
320 PRINT; PRINT "6-FORCE OF EXPERIMENT - ", Y$(50,55)
330 PRINT; PRINT "TIME-OF-DAY - ", Y$(63,68)
340 PRINT; PRINT "AIR TEMP IN ROOM (IN F) - ", Y$(69,73)
350 PRINT; PRINT "SUBJECT'S TEMP (IN F) - ", Y$(74,78)
360 PRINT; PRINT "SUBJECT'S HEART RATE (PER MIN) - ", Y$(79,82)
370 PRINT; PRINT "COMMENTS - ", Y$(83,142)
380 PRINT; PRINT "WRIST STIMULATED - ", V$(1,5)
390 PRINT; PRINT "SUBJECT'S THRESHOLD - ", V$(36,38)
400 PRINT; PRINT "STIMULUS INTENSITY - ", V$(40,43)
410 PRINT; PRINT "DELAY TIME (IN MS) - ", V$(8,9)
420 PRINT; PRINT "STIMULUS PER SECOND - ", V$(11,12)
430 PRINT; PRINT "LENGTH OF SWEEP - ", V$(30,34)
440 PRINT; PRINT "Dwell Time - ", V$(25,28)
450 PRINT; PRINT "STIMULUS PERIOD - ", V$(13,14)
460 PRINT; PRINT "NUMBER OF SWEEPS - ", V$(60,63)
490 PRINT; PRINT "NUMBER OF CHANNELS - ", YS(143,144)
500 PRINT; PRINT " MONTAGE FOR CHANNEL 1"
510 PRINT; PRINT " INPUT A", YS(215,219), " INPUT B", YS(220,224)
520 PRINT; PRINT " MONTAGE FOR CHANNEL 2"
530 PRINT; PRINT " INPUT A", YS(205,209), " INPUT B", YS(210,214)
540 PRINT; PRINT
550 PRINT; PRINT " MONTAGE FOR CHANNEL 3"
560 PRINT; PRINT " INPUT A", YS(195,199), " INPUT B", YS(200 - 204)
570 PRINT; PRINT " MONTAGE FOR CHANNEL 4"
580 PRINT; PRINT " INPUT A", YS(185,189), " INPUT B", YS(190,194)
590 PRINT; PRINT " MONTAGE FOR CHANNEL 5"
600 PRINT; PRINT " INPUT A", YS(175,179), " INPUT B", YS(180,184)
610 PRINT; PRINT " MONTAGE FOR CHANNEL 6"
620 PRINT; PRINT " INPUT A", YS(165,169), " INPUT B", YS(170,174)
630 PRINT; PRINT " MONTAGE FOR CHANNEL 7"
640 PRINT; PRINT " INPUT A", YS(155,159), " INPUT B", YS(160,164)
650 PRINT; PRINT " MONTAGE FOR CHANNEL 8"
660 PRINT; PRINT " INPUT A", YS(145,149), " INPUT B", YS(150,154)
670 PRINT; PRINT " START OF DATA MEMORY - ", "0"
680 PRINT; PRINT " SIZE OF DATA MEMORY - ", V$(20,24)
690 PRINT; PRINT " AMPLIFIER/FILTER SENSITIVITY (IN UV) - ", V$(46,49)
700 PRINT; PRINT " AMPLIFIER/FILTER LOW BANDPASS (IN HZ) - ", V$(51,53)
710 PRINT; PRINT " AMPLIFIER/FILTER HIGH BANDPASS (IN HZ) - ", V$(55,58)
720 PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT
730 PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT
740 PRINT; PRINT " DO YOU WISH TO LIST ANOTHER FILE? - ", X$
750 IF X$ = "Y" THEN 30 ; IF X$ = "N" THEN 760 ; IF X$ = X$ THEN 740
760 PRINT; PRINT " DO YOU WISH TO RUN ANOTHER PROGRAM? - ", X$
770 IF X$ = "Y" THEN 780 ; IF X$ = "N" THEN 790 ; IF X$ = X$ THEN 760
780 RUN EPPROG
790 STOP ; END

OK
5 REM DIMENSION VARIABLES
10 DIM Y$(225) , V$(66) , SUBNO$(4)
15 REM SET DATA MEMORY PARAMETERS AND SELECT DISK DRIVE # 2.
20 PUT 0 IN PST ; PUT 1024 IN PSZ ; PUT 2 IN PDV
30 PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
35 REM INPUT THE SUBJECT'S I.D. NUMBER
40 PRINT ; INPUT "WHAT IS THE SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
45 REM VALIDATION CHECK
50 PRINT "IS ", SUBNO$(1) , " CORRECT?"
60 INPUT X$ 70 IF X$ = "Y" THEN 80 ; IF X$ = X$ THEN 30
75 REM FIND THE FILE BASE AND THE FILE HEADER
80 BASE(1) = VAL(SUBNO$(3,4)) ; HEAD(1) = ASC(SUBNO$(1))
85 REM INPUT THE FILE SUBSCRIPT
90 PRINT ; INPUT "WHAT IS THE FILE SUBSCRIPT (3 0'S 0 - 999) - ", SUB(1)
95 REM VALIDATION CHECK
100 PRINT "IS ", SUB(1) , " CORRECT?"
110 INPUT X$ 120 IF X$ = "Y" THEN 130 ; IF X$ = X$ THEN 90
125 REM CHECK THE DATA RANGE OF THE SUBSCRIPT
130 IF SUB(1) < 0 OR SUB(1) > 999 THEN 90
135 REM DETERMINE IF SUBJECT IS HUMAN OR RHESUS AND READ THAT PERSONAL FILE.
140 IF HEAD(1) = 40 THEN 150 ; IF HEAD(1) = 50 THEN 160
150 PUT 200 IN PFH ; PUT BASE(1) IN PFB ; PUT SUB(1) IN PFS ; GOTO 170
160 PUT 210 IN PFH ; PUT BASE(1) IN PFB ; PUT SUB(1) IN PFS
170 XEO DRD
175 REM TRANSLATE DATA FROM ASCII CODE TO NORMAL CHARS. AND PLACE IN VARIABLE.
180 FOR S = 1 TO 225
190 X = DMEN(S) ; Y$(S,S) = CHR$(X)
200 NEXT S
205 REM READ THE PARAMETER FILE
210 PUT 208 IN PFH
220 XEO DRD
225 REM TRANSLATE DATA FROM ASCII CODE TO NORMAL CHARS. AND PLACE IN VARIABLE.
230 FOR S = 1 TO 66
240 X = DMEN(S) ; V$(S,S) = CHR$(X)
250 NEXT S
255 REM ADVANCE PAPER TO TOP OF A CLEAN PAGE.
260 PRINT ; PRINT "TURN PAPER TO THE TOP OF A FRESH PAGE SO THAT ALL"
270 PRINT " DATA WILL BE PRINTED ON IT'S OWN PAGE."
280 INPUT " AFTER FIXING PAPER TYPE Y TO CONTINUE. ", X$ 290 IF X$ = "Y" THEN 300 ; IF X$ = X$ THEN 260
295 REM PRINT OUT THE VARIABLES AND THEIR VALUES
300 PRINT ; PRINT ; PRINT ; PRINT "SUBJECT'S NAME - ", Y$(1,34)
310 PRINT ; PRINT "SUBJECT'S I.D. NUMBER - ", Y$(35,39)
320 PRINT ; PRINT "DATE OF EXPERIMENT - ", Y$(41,48)
330 PRINT ; PRINT "G-FORCE OF EXPERIMENT - ", Y$(50,55)
340 PRINT ; PRINT "EXPERIMENT'S RUN NUMBER - ", Y$(56,61)

8.33
350 PRINT; PRINT "TIME-OF-DAY - ", Y$(63,68)
360 PRINT; PRINT "AIR TEMP IN ROOM (IN F) - ", Y$(69,73)
370 PRINT; PRINT "SUBJECT'S TEMP (IN F) - ", Y$(74,78)
380 PRINT; PRINT "SUBJECT'S HEART RATE (PER MIN) - ", Y$(79,82)
390 PRINT; PRINT "COMMENTS - ", Y$(83,142)
400 PRINT; PRINT "WRIST STIMULATED - ", Y$(1,5)
410 PRINT; PRINT "SUBJECT'S THRESHOLD - ", Y$(36,38)
420 PRINT; PRINT "STIMULUS INTENSITY - ", Y$(40,43)
430 PRINT; PRINT "DELAY TIME (IN MS) - ", Y$(8,9)
440 PRINT; PRINT "STIMULUS PER SECOND - ", Y$(11,12)
450 PRINT; PRINT "LENGTH OF SLEEP - ", Y$(30,34)
460 PRINT; PRINT "Dwell time - ", Y$(25,28)
470 PRINT; PRINT "STIMULUS PERIOD - ", Y$(13,14)
480 PRINT; PRINT "NUMBER OF SLEEPS - ", Y$(60,63)
490 PRINT; PRINT "NUMBER OF CHANNELS - ", Y$(143,144)
500 PRINT; PRINT "MON TAGE FOR CHANNEL # 1"
510 PRINT; PRINT "INPUT A", Y$(215,219), " INPUT B", Y$(220,224)
520 PRINT; PRINT "MON TAGE FOR CHANNEL # 2"
530 PRINT; PRINT "INPUT A", Y$(205,209), " INPUT B", Y$(210,214)
540 PRINT; PRINT "MON TAGE FOR CHANNEL # 3"
550 PRINT; PRINT "INPUT A", Y$(195,199), " INPUT B", Y$(200-204)
560 PRINT; PRINT "MON TAGE FOR CHANNEL # 4"
570 PRINT; PRINT "INPUT A", Y$(185,189), " INPUT B", Y$(190,194)
580 PRINT; PRINT "MON TAGE FOR CHANNEL # 5"
590 PRINT; PRINT "INPUT A", Y$(175,179), " INPUT B", Y$(180,184)
600 PRINT; PRINT "MON TAGE FOR CHANNEL # 6"
610 PRINT; PRINT "INPUT A", Y$(165,169), " INPUT B", Y$(170,174)
620 PRINT; PRINT "MON TAGE FOR CHANNEL # 7"
630 PRINT; PRINT "INPUT A", Y$(155,159), " INPUT B", Y$(160,164)
640 PRINT; PRINT "MON TAGE FOR CHANNEL # 8"
650 PRINT; PRINT "INPUT A", Y$(145,149), " INPUT B", Y$(150,154)
660 PRINT; PRINT "START OF DATA MEMORY - "","0"
670 PRINT; PRINT "SIZE OF DATA MEMORY - ", Y$(20,24)
680 PRINT; PRINT "AMPLIFIER/FILTER SENSITIVITY (IN UV) - ", Y$(46,49)
690 PRINT; PRINT "AMPLIFIER/FILTER LOW BANDPASS (IN HZ) - ", Y$(51,53)
700 PRINT; PRINT "AMPLIFIER/FILTER HIGH BANDPASS (IN HZ) - ", Y$(55,58)
715 REM ADVANCE PAPER
720 PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT
730 PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT
735 REM RE-RUN THE PROGRAM IF WANTED.
740 PRINT; PRINT "DO YOU WISH TO LIST ANOTHER FILE? - ", X$  
750 IF X$ = "Y" THEN 30; IF X$ = "N" THEN 760; IF X$ = X$ THEN 740
755 REM RUN POINTER PROGRAM IF USER WANTS TO RUN ANOTHER PROGRAM.
760 PRINT; PRINT "DO YOU WISH TO RUN ANOTHER PROGRAM? - ", X$  
770 IF X$ = "Y" THEN 760; IF X$ = "N" THEN 790; IF X$ = X$ THEN 760
775 REM RUN EPPROG
780 REM STOP THE PROGRAM
790 STOP; END