UNITED STATES DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

PROGRAMS FOR THE AUTOMATIC PROCESSING AND
INTERPRETATION OF SCHLUMBERGER Sounding CURVES
IN QUICKBASIC 4.0

By

Adel A.R. Zohdy and Robert J. Bisdorf

Open-File Report 89-137 A

1989

89-137A Manual, examples, and program listings (Paper Copy).
89-137B Disk with computer programs and test example.

This report is preliminary and has not been reviewed for conformity with the U.S.
Geological Survey editorial standards. Any use of trade names is for descriptive purposes only and does not constitute endorsement by the USGS.
PROGRAMS FOR AUTOMATIC PROCESSING AND INTERPRETATION OF SCHLUMBERGER SOUNDING CURVES
in QuickBASIC 4.0

By
Adel A.R. Zohdy and Robert J. Bisdorf

INTRODUCTION

This open file is composed of two parts: (a) The manual you are reading now (including examples and program listings which are given in the appendix), and (b) the disk with computer programs and test examples.

Two computer programs are presented. The first program (ATO.EXE) is based on a new method for the automatic interpretation of Schlumberger sounding curves obtained over horizontally stratified media (Zohdy, in press). The second program (PICKCONT.EXE) is a utility program that reads layering files created by ATO.EXE and automatically interpolates the depths at preselected resistivity contour values. The resulting list of depths and resistivities facilitates the construction of contoured geoelectric cross sections.

The programs were written in Microsoft QuickBASIC 4.0. Each program is composed of several modules. The complete listings of the source code for the various modules are given in appendices A and B.
DISCLAIMER

Although these computer programs have been tested extensively and every effort has been made to assure their accuracy and performance, no guarantees are expressed or implied. Furthermore, any use of trade names is for descriptive purposes only and does not constitute endorsement by the U. S. Geological Survey.

SYSTEM REQUIREMENTS

This version of the program is for IBM and compatible computers running under MS-DOS version 2.1 or higher. The computer must be equipped with an EGA (Enhanced Graphics Adaptor) or CGA (Color Graphics Adaptor) card and at least 256 K RAM (random access memory). A math co-processor chip is strongly recommended but is not required. If the proper graphics card is not available the program displays a message to that effect and then exits.

An Epson or compatible printer is required to dump graphics and obtain a hard copy.

DISK CONTENTS

The accompanying disk contains the following:

1) ATO.EXE is the executable form of the automatic interpretation program.

2) ATO.BAS is the main module of the automatic interpretation program written in QuickBASIC version 4.0. It requires two modules: ATOSUB.BAS, DRCT.BAS. (See appendix A for a complete
listing of ATO.BAS, ATOSUB.BAS, and DRCT.BAS).

3) ATOSUB.BAS is a module containing several sub-programs that are used with ATO.BAS.

4) DRCT.BAS is a second module containing several sub-programs that are used with ATO.BAS.

5) ATO.MAK is a make utility created by QuickBASIC 4.0 to call the modules ATO.BAS, ATOSUB.BAS and DRCT.BAS.

6) EGADUMPL.COM is a public domain (PD) program for dumping EGA graphics on an Epson (or compatible) printer.

7) PICKCONT.EXE is the executable form of the pick contour program.

8) PICKCONT.BAS is the main module of the pick contour program written in QuickBASIC 4.0. It requires the following modules: ATOSUB.BAS, CONTSUB.BAS, and DRCT.BAS. The first and third modules are the same as those used with ATO.BAS. (see appendix B for a complete listing of PICKCONT.BAS and CONTSUB.BAS).

9) CONTSUB.BAS is a module consisting of one subprogram that contains the preselected resistivity contour values.

10) PICKCONT.MAK is a make utility created by QuickBASIC 4.0 to call the modules: PICKCONT.BAS, ATOSUB.BAS, CONTSUB.BAS, and DRCT.BAS.

11) Directory named TEST. This directory contains example data files.
GETTING STARTED

Before running the automatic interpretation program, make a backup copy of the disk. If you do not have a hard disk then make sure that you have a formatted floppy disk on which to save files. If you wish to copy the executable program ATO.EXE to a hard disk, make sure to copy the PD program EGADUMPL.COM also.

To run the program, type: ATO and press <ENTER>. In this manual, and on screen displays, we use <ENTER> to indicate the <ENTER> or <RETURN> key, and we use <Esc> to indicate the Esc (or Escape) key.

The program is menu driven and most prompts require a single key stroke: Y, N, a letter, a number, <Esc>, or <ENTER>. Prompts requiring a single key stroke, begin with the word PRESS. Prompts requiring an entry followed by a carriage return, end with the word <ENTER>. Furthermore, almost every menu or prompt includes a way (press <Esc>) or choice (press an appropriate number) to get out of that particular menu or prompt. Pressing <Esc> will return you to the MAIN MENU or to a similar menu.

After the program is loaded into memory, a screen is displayed showing the program title, credits, the program version, and the message "Press any Key to Continue". After you press a key, the program will proceed and test the computer system for the presence of an EGA. If successful, the program will load the PD program "EGADUMPL.COM". If unsuccessful, the program will test the system for the
presence of CGA. If successful, it will load "GRAPHICS.COM" from DOS. If unsuccessful it will display an appropriate message and then exit.

NOTE: For the program to find the file "GRAPHICS.COM", the appropriate path should have been set up by the user in the AUTOEXEC.BAT file.

DISK DRIVE MENU

Under MS-DOS, disk drives are designated by the letters A and B (for floppy drives) or C (for hard drive). Hard disks may be partitioned by the user and designated by other letters.

The Disk Drive Menu is a single prompt that asks you to press the appropriate letter (A, B, C, etc.) for the disk drive that you wish to save data to or to retrieve data from. If you do not wish to save the sounding data, press <Esc>. NOTE: If you press <Esc> and you change your mind later, you cannot save the sounding data you entered without returning to the DISK DRIVE MENU, designating a drive, and reentering the data.

MAIN MENU

The Main Menu has four options:

1. ENTER new sounding from keyboard.
2. LOAD a sounding from disk.
3. RETURN to DISK MENU.
4. QUIT.
Press the appropriate number to proceed.

Whether you wish to ENTER or LOAD data, the next prompt will request the name of the Survey Area. Soundings are grouped by Survey Area.

Individual files for: field sounding data, digitized sounding data, and interpreted layering are automatically saved under a single directory. This directory is automatically created by the program. Its name is based on the name of the survey area, and it is always placed under the Root Directory.

SURVEY AREA

The name of the survey area can be more than one word, contain a space, and can be up to 25 characters long. For example: LONE MOUNTAIN, Tule Valley, Alamo road, etc., are all valid names. Regardless of how you type the name of the Survey Area, it is automatically capitalized.

In order to minimize the amount of typing required by the user, the name of the Survey Area is entered only once (after selecting ENTER or LOAD from the MAIN MENU). Furthermore, the name of the survey area is automatically used in the TITLE of the sounding, in the name of the DIRECTORY, and in the FILE NAME. The following paragraphs describe how this is done.

Survey Area and Sounding Title:

The complete name of the survey area (including spaces, apostrophes, hyphens, etc) is used as the first part of the
title of a sounding. The second part of the sounding title is the sounding number (which is provided by the user at a subsequent prompt).

Example: the TITLE of sounding number 5 obtained in the Lone Mountain area will be automatically generated and saved by the program as: LONE MOUNTAIN 5.

**Survey Area and Name of Directory:**

The program will capitalize all the letters in the name of the SURVEY AREA, delete the delimiters (spaces, apostrophes, hyphens, commas, etc) and use the first 8 characters as the name of a directory to save data to or to retrieve data from.

Example: the name of the directory for data obtained in the Lone Mountain survey area is LONEMOUN. If a directory with the name LONEMOUN does not exist under the root directory, you will be asked if you wish to create a new directory with that name.

**Survey Area and File Name:**

In MS-DOS, each file name can be up to 8 characters long plus 3 extension characters (for a total of 11 characters).

The first 4 characters in the file name are taken from the first four characters of the Survey Area. The next four characters in the file name are reserved for the sounding number (e.g. 125). A letter indicating that the sounding was smoothed (S), extended (X), or corrected (C) for the effect of a winding road (Zohdy and Bisdorf, 1982) may be used as
part of the sounding number in the file name (e.g. 125X). The three extension characters in a file name are automatically designated as FLD (for field), DGT (for digitized), or LYR (for layering).

Example: the FIELD data for sounding 5 obtained in the Survey Area named Lone Mountain will be saved in a sequential file named LONE5.FLD. This file will be listed in the LONEMOUN directory, which is listed under the Root Directory.

UNITS

The Schlumberger current-electrode spacings (AB/2) may be entered in feet or meters. Thus when the prompt:

"Electrode spacings: in Feet or Meters? (F/M)"

is displayed, press F for feet or M for meters. If you enter the AB/2 spacings in feet, you will have the option later to convert them to meters and to display the sounding curve in the units you decided upon. The apparent resistivities are assumed to be in ohm-meters. The units for interpreted layer depths are the same as those used for the electrode spacings.

SOUNDING NUMBER

When prompted to enter a sounding number, enter only the appropriate number. You need not type the name of the survey area repeatedly for every sounding you wish to enter or load. As mentioned earlier, the program will combine the name of the SURVEY AREA and the sounding number to form the sounding title.
ENTERING SOUNDING DATA

Two methods are provided to enter sounding data. You may enter the data as obtained in the field (in AB/2 and apparent resistivity values), or you may enter a digitized sounding curve. In either method, the AB/2 units may be in feet or meters, the apparent resistivities are assumed to be in ohm-meters, the values of the potential electrode spacings (MN/2) are not required.

Standard Field Data:

For entering field data, enter all the AB/2 values first, all the apparent resistivities second, and then you may edit any errors using the provided edit submenu.

The AB/2 values must be entered in an ascending manner, one segment following another. Each segment is a set of AB/2 values used with a constant MN/2 value. The MN/2 values, which are assumed to be constant for each segment, are not entered (see example). When done entering all the AB/2 values, type 0 (zero) and press <ENTER>. Now you may enter the corresponding apparent resistivities.

The edit submenu allows you to Continue, Edit, Delete, Insert, Append, or Escape. You can not Continue unless the number of AB/2 equals the number of apparent resistivities.

Digitized Data:

You may wish to process the sounding curve and smooth it manually prior to entering it. If you do, you must shift the
various segments upward or downward to form a continuous sounding curve, then digitize the sounding curve at the rate of 6 points per logarithmic cycle.

To enter a digitized sounding, the program will prompt you to enter the smallest AB/2 value, and then all you have to do is to enter the digitized apparent resistivity values, as the program automatically increments the AB/2 values at the rate of 6 points per logarithmic cycle.

GRAPHIC DISPLAY

When the data for a given sounding are entered and you press C (to Continue) the sounding curve will be displayed on a 4 x 3 log-log graph with the message:

"CURVE LOOKS OK? (Y/N/<Esc>)."

If you press N, the program returns to the edit screen. If you press Y, you will be asked:

"Save this Sounding? (Y/N/<Esc>)"

if you press Y, then you will be asked if you wish to:

"Enter another sounding (Y/N/<Esc>)?"

If you press Y, the program returns to SOUNDOING NUMBER to enter another sounding. If you press N to either of the above two prompts, the program proceeds with processing the entered sounding data as follows.

If you entered the AB/2's in FEET you will be asked if you wish to convert them to METERS. Next you will be asked if you wish to:

"LIST field data and DUMP GRAPHICS? (Y/N/<Esc>)"
If you press Y you will be prompted with a message to make sure that the printer is turned on, to adjust the paper and to press <ENTER> when ready. Figure 1 shows an example of a field curve named TEST 2. The AB/2 values were entered in feet and converted to meters. The listing of the field data, and the EGA screen dump were made on an Epson FX-185 dot matrix printer.

DATA PROCESSING

A field sounding curve is processed by shifting the various sounding segments, upward or downward, with respect to a fixed segment, to form a continuous sounding curve. The segments on the field curve are numbered from right to left. That is, the segment obtained with the largest MN spacing is segment 1. Segment 1 is the default fixed segment in the shift prompt.

The shifted curve is digitized (or sampled), from right to left, at the rate of 6 points per logarithmic cycle. The digitized curve is displayed with a prompt to save it. If the digitized curve is saved the file will automatically have the extension DGT.
OPTIONS

Three options are offered prior to the automatic interpretation of a sounding curve. These are:

* Fix DEPTH-SHIFT factor at (..., 0.7, 0.8, ...) ?
* Change Number of Layers per cycle from 6 to ...?
* Fix last layer resistivity at ?

It is not recommended to use any of these options before becoming familiar with the program.

* The DEPTH-SHIFT factor refers to the shifting of the digitized electrode spacings to determine depths (Zohdy, in press). This (horizontal) depth-shift factor should not be confused with the (vertical) shifting of the various segments, as discussed on the previous page. The program is written so that an optimum depth-shift factor, usually in the range between .35 and .5, is determined. A fixed depth-shift factor may be selected by the user to fix the depths or to help in using layer compression.

* Changing the number of layers per log cycle determines the amount of layer compression or expansion. The default value is 6. For values greater than 6 (6.5, 7, 8, etc) the number of layers per logarithmic cycles is increased, hence the layers are compressed. Conversely, the number of layers per log cycle may be made less than 6 (5.5, 5, etc) to expand the layer thicknesses. Layer compression/expansion may be used alone or in conjunction with fixing the depth-shift factor. For sounding curves with flat left branches,
compression may be applied with a depth-shift factor that is equal to or greater than unity to produce highly compressed layers.

* The resistivity of the last layer may be fixed by the user at any positive value.

NOTE: the use of any or all of the above options does not guarantee a successful fit. Experimentation with these options is useful in imposing some constraints and in generating generally unlikely models.

AUTOMATIC INTERPRETATION

The automatic interpretation is generally done in two sets of iterations (two passes). In the first set of iterations, the best fitting theoretical sounding curve is determined. If the least root mean square (between digitized and calculated apparent resistivities) is greater than 2 percent then this means that the digitized curve is noisy and therefore anomalous layers may have been created. To eliminate these anomalous layers, the best fitting theoretical curve is reinterpreted as if it is a smoothed version of the digitized curve. If the digitized curve, under interpretation, is an exceptionally smooth and complete curve and is fitted within 2 percent in the first pass, then a second set of iterations is not performed (Zohdy, in press).

The user is kept aware of what is going on with a display of the root mean square values. When the interpretation is completed, a plot of the digitized data, the calculated
sounding curve, and the corresponding layering is displayed. The following prompt is displayed at the top of the screen:

"SAVE LAYERING ? (Y/N/<Esc>)"

If you press Y the layering (depths and resistivities) is automatically saved in a file with the extension LYR.

The next prompt is:

"LIST RESULTS and DUMP GRAPHICS ? (Y/N/<Esc>)"

If you press Y then a prompt is displayed to urge you to make sure that the printer is turned on, to adjust the paper, and to press <ENTER> when ready. Figure 2 shows the results of interpreting sounding TEST 2. The table of depths and resistivities is followed by a screen dump of the graphics.

Following the graphics dump, or if the answer to the above prompt was N, a menu with six options (including Quit) is displayed. The items on this menu are self explanatory.

LOADING AN EXAMPLE

Several test examples, including the sounding named TEST 2, are given under the directory TEST. To run one of the test examples type TEST for the name of the Survey Area then select LOAD from the Main Menu. The TEST Directory contains few files. When prompted for the sounding number you wish to load, you may type FILE to find out what sounding numbers exist under the directory TEST (which represents the name of a survey area). If you type FILE, the program will load the files with extensions FLD and DGT and display them on the screen. You may use the cursor keys to highlight the file
names. Highlight the file name you want and then press <ENTER>, or highlight EXIT to exit without making a selection.

If you know the sounding number you wish to load (for example sounding 2), then type 2 to load sounding TEST 2. The next prompt is:

"LOAD Field or Digitized sounding ? (F/D)"

Press the appropriate letter.

When a sounding is loaded and graphically displayed you will be prompted:

" Is this the sounding you want ? (Y/N) "

If you press N, you will go back to sounding number. If you press Y, you will be prompted:

" CURVE LOOKS OK ? (Y/N/<Esc>) "

If you press N, you get a chance to edit the sounding, re-save it, print it, etc. From here on, the program follows the same steps and prompts as if you entered a new sounding.

PICK CONTOUR

The program PICKCONT.EXE is a separate program that reads layering files created by ATO.EXE and automatically interpolates the depths at preselected resistivity values. The purpose is to facilitate the manual contouring of resistivity on geoelectric cross sections. The preselected resistivity contour values are: 1, 1.5, 2, 3, 4.5, 7, 10, 15, 20, 30, 45, 70, 100, 140, 200, 300, 450, ...., etc.
To run the PICKCONT.EXE program you must exit the program ATO.EXE and then type PICKCONT. The program will prompt you to select the drive to retrieve data from, then it will display the directories of the various survey areas so that you may select the one you wish by highlighting it. The next screen displays the file names that end with the extension LYR (layering files). Make sure the printer is turned on, select the file you want, and press <ENTER>. A list of the depths and corresponding, preselected, resistivity values is printed.

Lists of depths and preselected resistivities for soundings along a given cross section are very useful in the manual construction of geoelectric cross sections of contoured interpreted true resistivity. Table 1 shows the results of running PICKCONT for sounding TEST 2.

REFERENCES


<table>
<thead>
<tr>
<th>AB/2, m (ft)</th>
<th>App. Res.</th>
<th>AB/2, m (ft)</th>
<th>App. Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.05 (10.00)</td>
<td>3540.00</td>
<td>121.92 (400.00)</td>
<td>555.00</td>
</tr>
<tr>
<td>4.27 (14.00)</td>
<td>2940.00</td>
<td>182.88 (600.00)</td>
<td>400.00</td>
</tr>
<tr>
<td>6.10 (20.00)</td>
<td>1950.00</td>
<td>243.84 (800.00)</td>
<td>265.00</td>
</tr>
<tr>
<td>9.14 (30.00)</td>
<td>1130.00</td>
<td>304.80 (1000.00)</td>
<td>200.00</td>
</tr>
<tr>
<td>9.14 (30.00)</td>
<td>1030.00</td>
<td>426.72 (1400.00)</td>
<td>124.00</td>
</tr>
<tr>
<td>12.19 (40.00)</td>
<td>700.00</td>
<td>304.80 (1000.00)</td>
<td>220.00</td>
</tr>
<tr>
<td>18.29 (60.00)</td>
<td>620.00</td>
<td>426.72 (1400.00)</td>
<td>130.00</td>
</tr>
<tr>
<td>24.30 (80.00)</td>
<td>650.00</td>
<td>609.60 (2000.00)</td>
<td>120.00</td>
</tr>
<tr>
<td>30.48 (100.00)</td>
<td>630.00</td>
<td>914.40 (3000.00)</td>
<td>107.00</td>
</tr>
<tr>
<td>42.67 (140.00)</td>
<td>600.00</td>
<td>1219.20 (4000.00)</td>
<td>120.00</td>
</tr>
<tr>
<td>30.48 (100.00)</td>
<td>707.00</td>
<td>914.40 (3000.00)</td>
<td>98.80</td>
</tr>
<tr>
<td>42.67 (140.00)</td>
<td>690.00</td>
<td>1219.20 (4000.00)</td>
<td>121.00</td>
</tr>
<tr>
<td>60.96 (200.00)</td>
<td>645.00</td>
<td>1828.80 (6000.00)</td>
<td>149.00</td>
</tr>
<tr>
<td>91.44 (300.00)</td>
<td>640.00</td>
<td>2438.40 (8000.00)</td>
<td>218.00</td>
</tr>
<tr>
<td>91.44 (300.00)</td>
<td>620.00</td>
<td>3048.00 (10000.00)</td>
<td>252.00</td>
</tr>
</tbody>
</table>
### TEST 2

<table>
<thead>
<tr>
<th>DEPTH IN METERS</th>
<th>CONTOUR VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>4500.0</td>
</tr>
<tr>
<td>2.6</td>
<td>3000.0</td>
</tr>
<tr>
<td>4.4</td>
<td>2000.0</td>
</tr>
<tr>
<td>4.3</td>
<td>1500.0</td>
</tr>
<tr>
<td>4.6</td>
<td>250.0</td>
</tr>
<tr>
<td>8.6</td>
<td>700.0</td>
</tr>
<tr>
<td>17.4</td>
<td>700.0</td>
</tr>
<tr>
<td>26.1</td>
<td>700.0</td>
</tr>
<tr>
<td>51.3</td>
<td>450.0</td>
</tr>
<tr>
<td>112.7</td>
<td>200.0</td>
</tr>
<tr>
<td>141.6</td>
<td>150.0</td>
</tr>
<tr>
<td>260.1</td>
<td>100.0</td>
</tr>
<tr>
<td>248.9</td>
<td>70.0</td>
</tr>
<tr>
<td>487.5</td>
<td>70.0</td>
</tr>
<tr>
<td>245.8</td>
<td>70.0</td>
</tr>
<tr>
<td>755.3</td>
<td>150.0</td>
</tr>
<tr>
<td>849.4</td>
<td>150.0</td>
</tr>
<tr>
<td>979.4</td>
<td>500.0</td>
</tr>
<tr>
<td>1129.2</td>
<td>450.0</td>
</tr>
</tbody>
</table>

LAST DEPTH = 1222.7
DECLARE SUB CHKGRAPH (E!, GR$)
DECLARE SUB GRAPHSSET (GR$)
DECLARE SUB PRTRSCR ()
DECLARE SUB UPSORT (A$!, N$)
DECLARE SUB FILESCT (name$, f1$(!), N$)
DECLARE SUB DIRECT (A$!, N$, p$, k$)
DECLARE SUB LVES (A$!, B$!, N$, t$, U$, k$)
DECLARE SUB CLRSL ()
DECLARE SUB CLRNE (k$)
DECLARE SUB PRESSNYK ()
DECLARE FUNCTION PressNbrZ (N$)
DECLARE SUB ONEILL (Xk!, Nk$)
DECLARE SUB SETSCTN ()
DECLARE SUB Logplt (AX$!, AY$!)
DECLARE FUNCTION CAPNHT* ()
DECLARE SUB CNTR (YT!, A$)
DECLARE SUB HIGHL (A$)
DECLARE FUNCTION ALGT! (X!)
DECLARE FUNCTION DROUND! (tx!, ty!)
DECLARE SUB Kernel (L!, X!, t!, R!, V!, N$)
DECLARE SUB Convexes (Vv$!, Ves$!, Nrad$, Xk!, Nk$)
DECLARE SUB Splines (Wz!, X!, Y!, A$!, B$!, C$!, t!, D$!, p$!, S$!, k$)
DECLARE SUB Cubic! (Wz!, X!, Y!, A!, B!, C!, D!, p!, S!, k$)
DECLARE SUB Pltbox (AX$!, AY$!)
DECLARE SUB RHSs (RhoDiglO, Ves$!, Rho$, S$, Layers$)
DECLARE SUB Shft (ab$!, ob$!, npts$, absh$, obsh$!, nsh$, Isgn$)

DEFINT I-N
CONST FALSE = 0, true = NOT FALSE, conl = .3048, Xratio = 1.467799267581
ON ERROR GOTO ChkError
DIM AB2(45), PsOb(45), ABX(45), PSY(45), AX(5), AY(5), FILENAHE$(300)
DIM AbDig(45), RhoDig(45), RhoDigl(45), Rho(45), Thick(45), depth(45)
DIM Ves(45), Rho(45), Thick(45), Depth(45), Ves(45)
DIM Xaida(65), Vv(5), Xk(20), PsO(45), absh(45), obsh(45), A(45)
DIM B(45), C(45), D(2), S(45), p(45), Ps(45), It(65), Ss(65)
DIM F$((300), F2$(!50)
ONEILL Xk!
SETSCRN
WIDTH 80
LOCATE , , 0
Es$ = CHR*(27)

CNTR 6, " SCHLUMBERGER SOUNDING DATA PROCESSING AND INTERPRETATION PROGRAM "
CNTR 10, " By "
CNTR 12, " Adel A.R. Zohdy "
CNTR 14, " Robert J. Bisdorf "
CNTR 16, " U.S. GEOLOGICAL SURVEY, DENVER, CO. "
CNTR 20, " 1989 "
CNTR 22, " Version 1.8" 
PRESSANYK
CRVPLT = true
CHKGRAPH E, GR$ 
CRVPLT = FALSE
SETSCRN
IF GR$ = "N" THEN
    CLS
    BEEP
    CNTR 10, " SORRY, GRAPHICS NOT AVAILABLE ... "
    CNTR 12, " CAN'T USE THIS PROGRAM !. "
    PRSSANYKY
    CLS
    SYSTEM
END IF

CLS
CNTR 10, " PLEASE WAIT ....... "
AS = " LOADING " + GR$ + "BA GRAPHICS PRINTER DRIVER. "
CNTR 12, AS
IF GR$ = "E" THEN SHELL "EBADUHPL"
IF GR$ = "C" THEN SHELL "GRAPHICS"

DrvMnu:
    SETSCRN
    CNTR 23, " # If you DO NOT wish to save or retrieve press <Esc>. "
    CNTR 7, " DISK-DRIVE MENU "
    LOCATE 10, 20: PRINT " Please type letter of DISK DRIVE "
    LOCATE 12, 21
    HIGHL " (A, B, C, etc. ) "
    LOCATE 14, 20, 1
    PRINT " to SAVE or RETRIEVE DATA from: ";
    AS = CAPNMU$;
    IF AS = CHR$(13) THEN BEEP: GOTO DrvMnu
    IF AS = Es$ THEN DDS = "": GOTO MainMnu
    DDS = AS
    LOCATE 14, 52: PRINT DDS$: DDS$ = DDS$ + ":"
    LOCATE 16, 20: PRINT " Please wait ... ";
    chkdrive$ = "CHKDRIVE"
    OPEN DDS$ + chkdrive$ FOR INPUT AS #1
    CLOSE #1
    chkdrive$ = ""

MainMnu:
    SETSCRN
    LodF = 0
    Edt = 0
    Mnu$ = "ENTRY"
    ABUnit$ = ""
    LOCATE , , 0
    CNTR 7, " MAIN MENU "
    LOCATE 10, 26
    PRINT "1. "; HIGHL " ENTER "
    PRINT " SOUNDING FROM KEYBOARD."
    LOCATE 12, 26
    PRINT "2. "; HIGHL " LOAD "
    PRINT " SOUNDING FROM DATA DISK."
    LOCATE 14, 26
    PRINT "3. "; HIGHL " RETURN "
    PRINT " TO DISK-DRIVE MENU."
    LOCATE 16, 26
    PRINT "4. "; HIGHL " QUIT. "

    22
k = PressNumbr(4)
ON k GOTO SrvyArea, LodFile, DrvMnu, Kwit

SrvyArea:
  IF DDI$ = "" THEN GOTO ABUnit
  Dir$ = ""
  CLEAR
  file = FALSE
  LOCATE 10, 16: PRINT "Type name of " ; HIGHL " SURVEY AREA "
  PRINT " <ENTER>: "
  LOCATE 13, 16: PRINT "# If NO NAME press <ENTER>."
  LOCATE 14, 16: PRINT "# You may use SPACES in name of survey area."
  LOCATE 15, 16: PRINT "# Automatic Assignments:"
  LOCATE 16, 16: PRINT " Sounding Title = Survey Area + Sounding Number."
  LOCATE 17, 16: PRINT " Directory name = First 8 Letters of Survey Name."
  LOCATE 19, 52
  LINE INPUT "", SurveyName$
  IF SurveyName$ = "" THEN 60 TO ABUnit
  SurveyName$ = UCase$(SurveyName$)
  ABunit$ = SurveyName$
  GOSUB CapNTrim
  IF ABunit$ = "" THEN BEEP: GOTO SrvyArea
  chkdir:
    Dir$ = LEFT$(ABunit$, 8)
    LOCATE 10, 52: PRINT Dir$;
    LOCATE 22, 16: PRINT "Please wait, checking directory ..."
    IF LEN(Dir$) = 4 AND Dir$ = "FILE" THEN
      BEEP
      LOCATE 22, 16
      PRINT "Can't use the name FILE for Survey area!!"
      PRESS ANY KEY
      GOTO SrvyArea
    END IF
  chkdir$ = "dir"
  CHDIR DDI$ + "\" + Dir$
  IF chkdir$ <> "" THEN GOTO ABUnit
  BEEP
  LOCATE 22, 16
  PRINT "Above directory not found, wish to create a new one ? (Y/N)"
  GOSUB YsNo
  IF ABunit$ = "Y" THEN
    MKDIR DDI$ + "\" + Dir$
    GOTO ABUnit
  END IF
  GOTO SrvyArea

ABUnit:
  CLEAR
  ABunit$ = ""
  CNTR 12, " Electrode spacings: in Feet or Meters ? (F/M) "
  ABunit$ = CAPNTRIM
  IF ABunit$ <> "F" AND ABunit$ <> "M" THEN BEEP: GOTO ABUnit
  PRINT ABunit$
  IF ABunit$ = "F" THEN ABunit$ = "FEET"
  IF ABunit$ = "M" THEN ABunit$ = "METERS"
VesNabr:

SETSCRN
IF convert$ = "Y" THEN
  convert$ = "M"
  ABUnit$ = "FEET"
FOR i = 1 TO NAB2
  AB2(i) = AB2(i) / con1
NEXT i
END IF
CNTR B, " ENTER SOUNDING ... "
LOCATE 12, 16: PRINT " Type Sounding ";
HI6HL " NUMBER ": PRINT " (ENTER): "
LOCATE 14, 16: PRINT " (4 Digits and/or Characters max)"
LOCATE 16, 16: PRINT " * If no number type 0 (ENTER). "
LOCATE 18, 16: PRINT " * To return to MAIN MENU press (ENTER)."
LOCATE 12, 48: LINE INPUT ", VesNuir
IF VesNuir = " THEN GOTO MainHnu
IF VesNuir = "O" THEN VesNuir = ": GOTO DataEntryHnu
A* = VesNuir*
60SUB CapNTrii
VesNuir* = A$
LOCATE 14, 23
PRINT VesNuir; 
TITLE! = SurveyNai + * + VesNuir
DataEntryHnu:

SETSCRN
CNTR 4, " DATA ENTRY MENU 
VesTyp$ = "
LOCATE 11, 15
PRINT "1. Enter AB/2 and "; HI6HL " FIELD 
PRINT " APPARENT RESISTIVITIES.
LOCATE 13, 15
PRINT "2. Enter SMALLEST AB/2 and "; HI6HL " DIGITIZED 
PRINT " APPARENT RESISTIVITIES.
LOCATE 15, 15
PRINT "3. Return to "; HI6HL " MAIN MENU. 
k = PrssNbr(3)
IF k = 3 THEN GOTO MainHnu
IF k = 1 THEN
  VesTyp$ = "FLD"
  IF Dir$ = " THEN GOTO InptFld
ELSE
  VesTyp$ = "DGT"
  IF Dir$ = " THEN GOTO InptDgt
END IF
FILENAME$ = LEFT(Dir$, 4) + LEFT$(VesNuir$, 4) + "." + VesTyp$
LOCATE 23, 16: PRINT "Please wait ..."
CLOSE
ChkFil = 1
OPEN Dir$ + "\" + Dir$ + "\" + FILENAME$ FOR INPUT AS #1
CLOSE #1
IF ChkFil THEN
  BEEP
CLS
LOCATE 10, 12
PRINT "File for: "
LOCATE 12, 12
HIGHL * " + Dir$ + " + VesNum$ + " (" + VesTyp$ + ") already Exists!"
LOCATE 14, 12
PRINT "Do you wish to replace it ? (Y/N/\<Esc>)"
605UB YsNo
IF A$ = Es$ THEN GOTO MainMnu
IF A$ = "N" THEN GOTO VesNabr
END IF
IF VesTyp$ = "DGT" THEN GOTO InptDgt

InptFId:
COLOR 7, 1: CLS
LOCATE 21, 22: COLOR 0, 7
PRINT "ENTER ALL DATA NOW ........ EDIT LATER! "
LOCATE 22, 22
PRINT " Type AB/2 values & press <ENTER>.
LOCATE 23, 22
PRINT " When done, type 0 & press <ENTER>.
LOCATE 25, 22
PRINT " TO RETURN TO MENU TYPE -1 <ENTER>. "
COLOR 7, 1
xx = 5: y = 4: X = &: i = 1

AB2Hdr1:
LOCATE 2, xx - 2
PRINT 
LOCATE 2, xx
HIGHL * AB/2 *

AB2Nabr:
LOCATE y, X - 4: PRINT i
LOCATE y, X: INPUT "", ab$
IF ab$ = "" AND AB2(i) <> 0 THEN ab$ = STR$(AB2(i))
LOCATE y, X: HIGHL ab$
AB2(i) = VAL(ab$)
IF AB2(i) < 0 THEN AB2(i) = 0: GOTO DataEntryMnu
IF ab$ = "0" AND i = 1 THEN BEEP: GOTO AB2Nabr
IF ab$ = "0" THEN GOTO DonAB
IF ab$ <> "" AND AB2(i) <> 0 AND AB2(i) <> -1 THEN
  LOCATE y, X
  PRINT *
  LOCATE y, X
  HIGHL STR$(AB2(i))
END IF
IF AB2(i) = 0 THEN
  BEEP
  LOCATE y, X
  PRINT *
  GOTO AB2Nabr
END IF
i = i + 1
IF y = 18 THEN
  y = 4
  xx = xx + 24
  X = X + 24
ELSE
    y = y + 1
END IF
GOTO AB2Nibr

DonAB:
CLR5L
NAB2 = i - 1
xx = 15
y = 4
X = 16
i = 1

RoFldPrapt:
   LOCATE 21, 22: COLOR 0, 7
   PRINT " ENTER ALL DATA NOW ......... EDIT LATER! "
   LOCATE 22, 22
   PRINT " Type APP. RESISTIVITY & press <ENTER>. "
   LOCATE 23, 22
   PRINT " When done type 0 & press <ENTER>. "
   COLOR 7, 1
25 LOCATE 2, xx - 2
   PRINT "* 
   LOCATE 2, xx
   HIGHL " APP.RHO "

InptRo:
   LOCATE y, X - 4: PRINT i
   LOCATE y, X: INPUT "*, Ps$"
   IF Ps$ = "*" AND PsOb(i) <> 0 THEN Ps$ = STR$(PsOb(i))
   LOCATE y, X: HIGHL Ps$ 
   PsOb(i) = VAL(Ps$)
   IF PsOb(i) < 0 THEN 
     PsOb(i) = 0
   END IF
   IF Ps$ = "0" AND i = 1 THEN 
     BEEP
     GOTO InptRo
   END IF
   IF Ps$ = "0" THEN 
     NAPR = i - 1
     GOTO ChkNibr
   END IF
   IF Ps$ <> "*" AND PsOb(i) <> 0 AND PsOb(i) <> -1 THEN
     LOCATE y, X
     PRINT 
     LOCATE y, X
     HIGHL STR$(PsOb(i))
   END IF
   IF PsOb(i) = 0 THEN BEEP: GOTO InptRo
   i = i + 1
   IF y = 18 THEN
     y = 4
     xx = xx + 24
   X = X + 24
GOTO 25
END IF
y = y + 1
GOTO InptRo

ChkNabr:
IF NAB2 <> NAPR THEN GOSUB NtEq
GOSUB Disp
GOTO EdtPrompt

InptDgt:
VesTyp$ = "GDT"
CLS
LOCATE 12, 12
PRINT "PLEASE TYPE SMALLEST DIGITIZED AB/2 <ENTER>"
LOCATE 16, 20
HIGHL * TO RETURN TO MENU PRESS <ENTER>"

ABMIN:
LOCATE 12, 60
INPUT **, ABBIN$
IF ABBIN$ = "" THEN GOTO DataEntryMnu
IF VAL(ABBIN$) = 0 THEN BEEP: GOTO ABBIN$
AbDgt(i) = VAL(ABBIN$)
COLOR 7, 4: CLS
LOCATE 21, 16: COLOR 0, 7
PRINT "ENTER ALL DATA NOW ................. EDIT LATER!"
LOCATE 22, 16
PRINT "INPUT DIGITIZED APP. RESISTIVITIES & PRESS <ENTER>"
LOCATE 23, 16
PRINT "WHEN DONE, TYPE 0 AND PRESS .......... <ENTER>"
LOCATE 25, 16
PRINT "TO RETURN TO MENU TYPE -1 AND PRESS <ENTER>"
COLOR 7, 4
xx = 5: y = 4: X = 6: i = 1

DgtHdr:
LOCATE 2, xx - 2: PRINT "#"
LOCATE 2, xx
COLOR 0, 7: PRINT "AB/2": COLOR 7, 4
GOSUB PrtAbDgt
xx = xx + 10: y = 4: X = X + 10
LOCATE 2, xx - 2: PRINT "#"
LOCATE 2, xx
COLOR 0, 7: PRINT "APP.RHO": COLOR 7, 4

InptRoDgt:
LOCATE y, X - 4: PRINT i
LOCATE y, X: INPUT **, RhoDgt$
IF RhoDgt$ = "" AND RhoDgt(i) > 0 THEN RhoDgt$ = STR$(RhoDgt(i))
LOCATE y, X
COLOR 0, 7: PRINT RhoDgt$: COLOR 7, 4
RhoDgt(i) = VAL(RhoDgt$)
IF RhoDgt(i) < 0 THEN RhoDgt(i) = 0: GOTO DataEntryMnu
IF RhoDgt$ = "0" AND i = 1 THEN BEEP: GOTO InptRoDgt
IF RhoDgt$ = "0" THEN NAPR = i - 1: GOTO DnRhoDgt
IF RhoDig* <> " AND RhoDig(i) <> 0 AND RhoDig(i) <> -1 THEN
  LOCATE y, X
  PRINT *
  LOCATE y, X
  COLOR 0, 7: PRINT RhoDig(i): COLOR 7, 4
END IF
IF RhoDig(i) = 0 THEN BEEP: GOTO InptRoDgt
  y = y + 1
  i = i + 1
  AbDig(i) = AbDig(i - 1) * Xratio
IF y = 19 THEN
  yM
ELSEIF i = 16 THEN
  xx = 30
  X = 30
ELSEIF i = 31 THEN
  xx = 60
  X = 60
END IF
GOTO DgtHdr
ELSEIF y < 19 AND i < 16 THEN
  xx = 5
  X = 6
ELSEIF y < 19 AND i > 16 AND i < 31 THEN
  xx = 30
  X = 30
ELSEIF y < 19 AND i > 31 THEN
  xx = 60
  X = 60
END IF
GOSUB PrtAbDgt
  xx = xx + 10
  X = X + 10
GOTO InptRoDgt

DnRhoDig:
  NAB2 = NAPR
  FOR i = 1 TO NAPR
    AB2(i) = AbDig(i)
    PsOb(i) = RhoDig(i)
  NEXT i
GOSUB Dsply

EdtPrEp:
  Edt = -1
  CLR5L
  LOCATE 21, 11
  PRINT "Press: "; ; HIGHL " C "; PRINT " to Continue, ";
  HIGHL " E "; PRINT " to Edit, ";
  HIGHL " D "; PRINT " to Delete, "
  LOCATE 23, 18
  HIGHL " I "; PRINT " to Insert, ";
  HIGHL " A "; PRINT " to Append, ";
  HIGHL " <Esc> "; PRINT " to Main Menu. "

WhrTo:
A$ = CAPNHT$  
IF A$ = "E" THEN B$ = "EDIT" : GOTO EdtDel  
IF A$ = "D" THEN B$ = "DELETE" : GOTO EdtDel  
IF A$ = "I" THEN B$ = "INSERT" : GOTO Insrt  
IF A$ = "A" THEN GOTO Apnd  
IF A$ = Es$ THEN GOTO MainMnu  
IF A$ = "C" THEN  
  IF NAB2 <> NAPR THEN  
    GOSUB NtEql  
    GOTO EdtPmpkt  
  ELSEIF VesTyp$ = "DBT" THEN  
    FOR i = 1 TO NAB2  
      AbDig(i) = AB2(i)  
      RhoDig(i) = PsOb(i)  
    NEXT i  
  END IF  
  IF NAB2 = NAPR THEN GOTO Pnt  
END IF  
BEEP  
GOTO WhrTo

EdtDel:

  C$ = * TYPE ENTRY NUMBER TO *: A$ = C$ + B$  
  CLR$  
  LOCATE 21, 1B  
  COLOR 3, 1: PRINT C$;  
  HIGHL B$  
  COLOR 3, 1: PRINT "<ENTER>": COLOR 7, 1  
  LOCATE 22, 1B  
  COLOR 7, 1: PRINT "If NO CHANGE........ press <ENTER>"  
  LOCATE 21, 18 + LEN(A$) + 12  
  COLOR 7, 1: INPUT ", N$": COLOR 7, 1  
  IF N$ = "" THEN  
    GOSUB Dsply  
    GOTO EdtPmpkt  
  END IF  
  IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO EdtDel  
  N = VAL(N$)  
  IF N = 0 THEN  
    BEEP  
    GOTO EdtDel  
  END IF  
  IF N > Q THEN  
    GOSUB WrngNbr  
    GOSUB Dsply  
    GOTO EdtDel  
  END IF  
  IF B$ <> "EDIT" THEN GOTO Del  
  IF N > NAB2 THEN  
    y = 3 + N  
    IF N <= 15 THEN LOCATE y, 7  
    IF N > 15 AND N <= 30 THEN LOCATE y - 15, 33  
    IF N > 30 AND N <= 45 THEN LOCATE y - 30, 59  
    AK = AB2(N)  
    COLOR 0, 7: INPUT "", ab$  
    IF ab$ = "" THEN AB2(N) = AK ELSE AB2(N) = VAL(ab$)
IF AB2(N) <= 0 THEN BEEP: AB2(N) = AK
GOSUB Dsply
GOTO EdtDel

1440  
    y = 3 + N - NAB2
    IF N <= 15 + NAB2 THEN LOCATE y, 19: GOTO 1480
    IF N <= 30 + NAB2 THEN LOCATE y - 15, 44: GOTO 1460
    IF N <= 45 + NAB2 THEN LOCATE y - 30, 69: GOTO 1460

1460  
    AKK = PsOb(N - NAB2)
    COLOR 0, 7
    INPUT ": Ps$"
    IF Ps$ = " THEN PsOb(IN - NAB2) = AKK: GOTO 1500
    PsOb(N - NAB2) = VAL(Ps$)
    IF PsOb(N - NAB2) <= 0 THEN BEEP: PsOb(N - NAB2) = AKK

1500  
    GOSUB Dsply
    GOTO EdtDel

Del:
    IF B$ <> " DELETE " THEN GOTO Insrt
    IF N > NAB2 THEN GOTO DelRo

DelAB2:
    FOR i = N TO NAB2
        AB2(i) = AB2(i + 1)
    NEXT i
    NAB2 = NAB2 - 1
    GOSUB Dsply
    GOTO EdtDel

DelRo:
    IF N > 0 THEN
        GOSUB WrngNabr
        GOSUB Dsply
    END IF
    FOR i = N TO Q
        PsOb(i - NAB2) = PsOb(i - NAB2 + 1)
    NEXT i
    NAPR = NAPR - 1
    GOSUB Dsply
    GOTO EdtDel

Insrt:
    CLRSL
    LOCATE 21, 6
        COLOR 3, 1: PRINT "Type entry number ";: COLOR 1, 3
        PRINT " ABOVE WHICH ";
        COLOR 3, 1: PRINT " you wish to ";
        COLOR 0, 7: PRINT B$: : COLOR 3, 1
        PRINT " & press <ENTER> "
    LOCATE 22, 22
        COLOR 7, 1: PRINT "IF NO CHANGE ...... press <ENTER> "
    LOCATE 21, 75
        INPUT ":, N$
    N = VAL(N$)
IF N$ = "" THEN GOTO EdtPrmpt
IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO Insr
IF N = 0 THEN BEEP: GOTO Insr
IF N > NAB2 THEN GOTO InsrRo

InsrtAB:
LOCATE 23, 1: PRINT STRING$(79, " ")
LOCATE 23, 21
HIGHL " TYPE VALUE OF AB/2 TO BE INSERTED "
PRINT " ";
INPUT ": N$
IF N$ = "" THEN GOTO EdtPrmpt
IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO InsrtAB
AH = VAL(N$)
IF AH = 0 THEN BEEP: GOTO InsrtAB
NAB2 = NAB2 + 1
FOR i = NAB2 TO N + 1 STEP -1
   AB2(i) = AB2(i - 1)
NEXT i
AB2(N) = AH
GOSUB Dsply
GOTO Insr

InsrtRo:
IF N > Q THEN
   GOSUB WrngNbr
   GOSUB Dsply
   GOTO Insr
END IF
LOCATE 23, 1: PRINT STRING$(79, " ")
LOCATE 23, 22
HIGHL " TYPE VALUE OF APP.RHD TO BE INSERTED "
PRINT " ";
INPUT ": N$
IF N$ = "" THEN GOTO EdtPrmpt
IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO InsrtRo
AM = VAL(N$)
IF AM = 0 THEN BEEP: GOTO InsrtRo
NAPR = NAPR + 1: Q = Q + 1
FOR i = Q TO N + 1 STEP -1
   PsOb(i - NAB2) = PsOb(i - NAB2 - 1)
NEXT i
PsOb(N - NAB2) = AM
GOSUB Dsply
GOTO Insr

WrngNbr:
BEEP
CLRSL
CNTR 24, " THERE IS NO SUCH NUMBER! "
PRESSANYKY
CLRSL
RETURN

Apnd:
CLRSL
LOCATE 21, B
PRINT " Press ";
HIGHL * A *: PRINT " to append ";
HIGHL * AB/2 *: PRINT " , or press ";
HIGHL * R *: PRINT " to append ";
HIGHL * APP.RHO *
LOCATE 23, 22
PRINT " If NO CHANGE ...... Press <ENTER> "
A$ = CAPNW$;
IF A$ = CHR$(13) THEN GOTO EdtPrapt
IF A$ = "A" THEN GOTO ApndAB
IF A$ = "R" THEN GOTO ApndRo
BEEP: GOTO Apnd

ApndAB:
CLR5L
LOCATE 23, 22
PRINT " If NO CHANGE ...... Press <ENTER> "
LOCATE 22, 18
HIGHL "Type VALUE OF AB/2 to be added <ENTER>"
INPUT "", N$
IF N$ = " " THEN GOTO EdtPrapt
IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO ApndAB
AN = VAL(N$)
IF AN <= 0 THEN BEEP: GOTO ApndAB
NAB2 = NAB2 + 1
AB2(NAB2) = AN
60SUB Dsply
GOTO Apnd

ApndRo:
CLR5L
LOCATE 23, 22
PRINT " If NO CHANGE ...... Press <ENTER> "
LOCATE 22, 18
HIGHL "Type VALUE OF APP.RHO to be added <ENTER>"
INPUT "", N$
IF N$ = " " THEN GOTO EdtPrapt
IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO ApndRo
AN = VAL(N$)
NAPR = NAPR + 1
PsoB(NAPR) = AN
60SUB Dsply
GOTO Apnd

Pit:
60SUB FldPit

CrvOk:
CLRME 1
LOCATE 1, 24
PRINT " CURVE LOOKS OK ? (Y/N/<Esc>) "
60SUB YsNo
IF A$ = Es$ THEN GOTO MainMnu
IF A$ = "N" THEN

SETRCRN
GOSUB Dispy
GOTO EdtPrapt
END IF
IF DD$ = "" OR Dir$ = "" THEN GOTO FtToM
CLRLINE 1
IF Edt = -1 THEN
LOCATE 1, 20
PRINT "Save this Sounding? (Y/N/<Esc>)"
GOSUB YsNo
IF A$ = Es$ THEN GOTO MainMnu
IF A$ = "N" THEN GOTO FtToM
GOSUB SaveIt
END IF
IF LodF <> -1 THEN
CLRLINE 1
LOCATE 1, 15
PRINT "Enter another sounding? (Y/N/<Esc>)"
GOSUB YsNo
IF A$ = Es$ THEN GOTO MainMnu
IF A$ = "Y" THEN GOTO VesMabr ELSE GOTO FtToM
END IF

FtToM:
convert$ = ""
CLRLINE 1
LOCATE 1, 15
IF ABUnit$ = "METERS" THEN
GOSUB Lb12
GOTO DipBrfx
END IF
IF ABUnit$ = "FEET" THEN
CLRLINE 1
LOCATE 1, 15
PRINT "CONVERT AB/2 from FEET to METERS? (Y/N/<Esc>)"
GOSUB YsNo
IF A$ = Es$ THEN GOTO MainMnu
IF A$ = "N" THEN
convert$ = "N"
GOSUB FldFlt
GOTO DipBrfx
END IF
convert$ = "Y"
ABUnit$ = "METERS"
IF VesTyp$ = "DGT" THEN
FOR i = 1 TO NAB2
    AbDig(i) = AbDig(i) * con1
    AB2(i) = AbDig(i)
NEXT i
ELSE
FOR i = 1 TO NAB2
    AB2(i) = AB2(i) * con1
NEXT i
END IF
GOSUB FldFlt
GOTO DipBrfx
END IF
file = FALSE
CNTR B, " LOAD FILE ..... "
LOCATE 12, 16: PRINT "Type name of ";
HIGHL " SURVEY AREA ": PRINT " <ENTER>: ">
LOCATE 15, 16: PRINT "* You may use SPACES in name of survey area.*"
LOCATE 17, 16: PRINT "* To return to MAIN MENU press <ENTER>frei"
LOCATE 12, 52
LINE INPUT ", SurveyName$
IF SurveyName$ = " THEN GOTO MainMnu
SurveyName$ = UCASE$(SurveyName$)
At = SurveyName$
GOSUB CapNTria
IF At = " THEN BEEP: GOTO LodFile
Dir$ = LEFT$(At, 8)
LOCATE 12, 52: PRINT Dir$;
LOCATE 20, 16: PRINT "Please wait, checking directory ...
IF LEN(Dirt) - 4 AND Dirl = "FILE 1 THEN
BEEP
LOCATE 20, 16
PRINT "Can't use the name FILE for Survey area !!!!
PRESSANYKY
GOTO LodFile
END IF
chkdir$ = "LOD"
CHDIR "\"
MKDIR D$: + "\" + Dir$
IF chkdir$ = "LOD" THEN
BEEP
LOCATE 20, 16
PRINT "Above directory not found !!
RMDIR D$: + Dir$
PRESSANYKY
GOTO LodFile
END IF
LodVes:
SETSCRN
IF D$: = "" OR Dir$ = "" THEN
BEEP
CLS
LOCATE 10, 10
PRINT " Can't LOAD, you did not select a Drive and/or a Directory !!!
PRESSANYKY
GOTO MainMnu
END IF
Lod1:
CNTR B, " LOAD FILE ...... "
LOCATE 15, 16: PRINT "* To see file names type FILE.*"
LOCATE 16, 16: PRINT "* To return to MAIN MENU press <ENTER>."
LOCATE 12, 16: PRINT " Type Sounding ";
HIGHL " NUMBER ": PRINT " <ENTER>: ";
LINE INPUT At
IF At = " THEN GOTO MainMnu
GOSUB CapNTria
VesNum$ = At$
IF LEFT$(VesNum$, 4) = "FILE" THEN
  CLS
  CNTR 12, "LOADING DISK DIRECTORY: .... PLEASE WAIT."
  IF file THEN GOTO file!
  $ = DD$ + "\" + Dir$ + "\.*.FLD"
  DIRECT F1$(1), M1, $, 1
  UPSORT F1$(1), M1
  $ = DD$ + "\" + Dir$ + "\.*.DGT"
  DIRECT F2$(1), M2, $, 1
  UPSORT F2$(1), M2
  FOR i = M1 + 1 TO M1 + M2
    F1$(i) = F2$(i - M1)
  NEXT i
  NF = M1 + M2 + 1
  F1$(NF) = "EXIT"
  file = true

file:
  CNTR 1, "SELECT A FILE OR SELECT EXIT TO RETURN"
  FILESCT FILENAME$, F1$(1), NF
  IF FILENAME$ = "EXIT" THEN GOTO LodFile
  VesTyp$ = RIGHT$(FILENAME$, 3)
  ist = LEN(Dir$)
  IF ist > 4 THEN ist = 4
  VesNum$ = MID$(FILENAME$, ist + 1, LEN(FILENAME$) - ist - 4)
  GOTO Lod4
END IF

Lod2:
  LOCATE 20, 10: PRINT STRING$(60, " "); LOCATE 20, 10
  PRINT "Type of sounding: FIELD or DIGITIZED? (F/D/<Esc>) ":
  A$ = CAPNMT$
  IF A$ = Es$ THEN GOTO MainNnu
  IF A$ = "F" THEN VesTyp$ = "FLD": GOTO Lod3
  IF A$ = "D" THEN VesTyp$ = "DGT": GOTO Lod3
  BEEP
  GOTO Lod2

Lod3:
  FILENAME$ = LEFT$(Dir$, 4) + LEFT$(VesNum$, 4) + "." + VesTyp$

Lod4:
  OPEN DD$ + "\" + Dir$ + "\" + FILENAME$ FOR INPUT AS #1
  IF VesTyp$ = "FLD" THEN
    LVES AB20, PsOb0, NAB2, TITLE*, ABUnit*, 1
  ELSE
    LVES AbDig0, RhoDig0, NAB2, TITLE*, ABUnit*, 1
    FOR i = 1 TO NAB2
      AB2(i) = AbDig(i)
      PsOb(i) = RhoDig(i)
    NEXT i
  END IF
  NAPR = NAB2
  CLOSE #1
  LodF = -1
  Edt = 0

36
GOSUB FldPlt
CLRLNE 1
LOCATE 1, 15
PRINT "Is this the sounding you want? (Y/N)"
GOSUB YsNo
IF A$ = Es$ OR A$ = "N" THEN GOTO LodVes
GOTO CrvOk

Ato:

SETSCRN
Replough = 0
Ft = 2 'First FITTING TOLERANCE
Layers = NAB2
Nrad = 20 + Layers 'Nrad = # of Kernel Function Spacings
IF VesTyp$ = "DST" THEN
    Radmin = AbDig(1)
    FOR i = 1 TO Layers
        RhoDigl(i) = RhoDig(i)
    NEXT i
    GOTO KrnlSpcing
END IF
FOR i = 1 TO Layers
    RhoDigl(i) = RhoDig(i)
NEXT i

KrnlSpcing:
Xmin = Radmin / Xratio ^ 14 / 1.1396 'Xmin = Smallest KERNEL FUNCTION SPACING
Xama(1) = Xmin
FOR i = 2 TO Nrad ' COMPUTE KERNEL FUNCTION SPACINGS (XAMA)
    Xama(i) = Xratio * Xama(i - 1)
NEXT i

Options:
Compress = 6 ' NUMBER OF LAYERS PER CYCLE
Shift = 0 ' DEPTH-SHIFT factor
Iterate = 1 ' NUMBER OF ITERATIONS
Max = 60 ' MAX NUMBER OF ITERATIONS
FixRhoN = 0
CLS
CNTR 6, "OPTIONS"
LOCATE 10, 10
PRINT "Do You wish to use any of the following options? (Y/N)"
LOCATE 12, 10
PRINT "* Fix DEPTH-SHIFT factor at (...,0.7, 0.8, ...) ?*"
LOCATE 14, 10
PRINT "* Change Number of Layers/Cycle from 6 to (5.5, 6.5, etc.)?*
LOCATE 16, 10
PRINT "* Fix Last Layer Resistivity at ........?*
GOSUB YsNo
IF A$ = Es$ THEN GOTO MainMnu
IF A$ = "Y" THEN
    LOCATE 20, 10
    COLOR 7, 0
    PRINT "If no change press <ENTER>, else type REQUIRED VALUE."
    COLOR 7, 1
LOCATE 12, 67
LINE INPUT A$
IF A$ <> "" THEN Shift = VAL(A$)
LOCATE 14, 72
LINE INPUT A$
IF A$ <> "" THEN Compress = VAL(A$)
LOCATE 16, 60
LINE INPUT A$
IF A$ <> "" THEN FixRhoN = VAL(A$)
END IF

DoAto:
CLS
CNTR 3, " AUTOMATIC INTERPRETATION WORKING "
LOCATE 5, 1

Init:
Squish = 10 ^ (1 / Compress) ' MULTIPLIER FOR GENERATING COMPRESSED DEPTHS
RMSmin = 99999999 ' RMSmin= SMALLEST RMS PERCENT
FOR j = 1 TO Layers
   Rho(j) = RhoDiglj)
NEXT j
IF Shift = 0 THEN
   Shift = .6
   FOR j = 1 TO 10
      Shift = Shift * .9
      GOSUB LCompress
      GOSUB LVES
      IF RMS < RMSmin THEN
         ShiftMin = Shift
         RMSmin = RMS
         PRINT USING "Shift = ###.### RMS = ###.###"; ShiftMin, RMSmin
      ELSE
         EXIT FOR
      END IF
   NEXT j
Shift = ShiftMin
END IF
GOSUB LCompress 'COMPUTE SHIFTED AND COMPRESSED DEPTHS AND THICKNESSES
GOSUB LVES 'COMPUTE KERNEL & VES & COMPARE VES TO RhoDig
GOSUB SaveLyrNVes
RMSmin = RMS

FlipFlop:
FOR i = 1 TO Layers
   Rhod) = RhoDigd) / Ves(i) * Rho(i)
NEXT i
GOSUB LVES 'COMPUTE AND COMPARE
PRINT USING " RMS = ###.###"; RMS
IF RMS > RMSmin THEN GOTO Replough 'FLIP FAILS TO IMPROVE
GOSUB SaveLyrNVes
IF (RMSmin / RMS - 1) * 100 < 5 THEN GOTO Replough
RMSmin = RMS
IF RMSmin < Ft OR Iterate > Max THEN GOTO Graphics
Iterate = Iterate + 1
GOTO FlipFlop ' FLIP-FLOP SOME MORE
Replough:
   IF Replough = 1 THEN GOTO Graphics
   CLS
   CIb TR 3, " INTERPRETING SMOOTH CURVE ...... ",
   LOCATE 5, 1
   FOR i = 1 TO Layers
      RhoDig(i) = Vesi(i)
   NEXT i
   Replough = 1
   Ft = 1
   GOTO Init

LCompress:
   depth(1) = AbDig(i) * Shift
   Thick(1) = depth(1)
   FOR i = 2 TO Layers
      depth(i) = AbDig(i) * Shift
      depth(i) = depth(i - 1) * Squish
      Thick(i) = depth(i) - depth(i - 1)
   NEXT i
   depth(Layers) = 99999
   RETURN

LVES:
   IF FixRhoN <> 0 THEN Rho(Layers) = FixRhoN
   CALL Kernel(Layers, Xada(), Thick(), Rho(), Vv(), Nrad)
   CALL Conves(Vv(), Vesi(), Nrad, Xk(), Xk)
   CALL RHSHRhoDig(), Vesi(), RKS, Layers)
   RETURN

SaveLyrNVes:
   FOR i = 1 TO Layers
      Vesfd(i) = Ves(i)
      Rhof(i) = Rho(i)
      Depthf(i) = depth(i)
      Thickf(i) = Thick(i)
   NEXT i
   RETURN

Graphics:
   GOSUB IntrPlt
   IF Dir$ = "" THEN GOTO Listea
   CLRLME 1
   LOCATE 1, 10
   PRINT " SAVE LAYERING ? (Y/N/<Esc>) ",
   GOSUB YsNo
   IF A$ = Es$ THEN GOTO MainMnu
   IF A$ = "Y" THEN
      CLRLME 1
      LOCATE 1, 20
      PRINT "SAVING ......."
FILENAME$ = LEFT$(FILENAME$, INSTR(FILENAME$, ".") + "LYR")
OPEN DDS + "\" + Dir$ + "\" + FILENAME$ FOR OUTPUT AS 02
PRINT 02, TITLE$
PRINT 02, ABUnit$
PRINT 02, Layers
FOR i = 1 TO Layers
  PRINT 02, Depthf(i), Rhof(i)
NEXT i
CLOSE 02
CLRLINE 1
LOCATE 1, 1
PRINT TITLE$; " saved as: ";
PRINT " "; FILENAME$; " "; Press Any Key to Continue"
WHILE INKEY$ = "": WEND
END IF
Listen:
CLRLINE 1
LOCATE 1, 10
PRINT " LIST RESULTS & DUMP GRAPHICS on printer? (Y/N/<Esc>) ".
GOSUB YsNo
IF A$ = Es$ OR A$ = "N" THEN GOTO Whatnow
GOSUB PrntrRdy
nspc = 14
IF convert$ = "Y" THEN nspc = 6
LPRINT: LPRINT SPC(nspc); TITLE$; " (INTERPRETATION) ": LPRINT
XNu = INT(Layers / 2)
XNn = Layers / 2
IF convert$ = "N" OR convert$ = "" THEN
  forma$ = "
  form$ = "
  LPRINT
  FOR i = 1 TO XNu
    LPRINT USING forma$; Depthf(i), Rhof(i), Depthf(i + XNu), Rhof(i + XNu) NEXT i
  IF XNn > XNu THEN LPRINT USING forma$; Depthf(Layers), Rhof(Layers)
ELSE
  forma$ = " DEPTH, ( ft ) RESIS. DEPTH, ( ft ) RESIS."
  form$ = "
  forma$ = "
  form$ = "
  LPRINT forma$: LPRINT
  FOR i = 1 TO XNu
    LPRINT USING forma$; Depthf(i), Depthf(i) / conl, Rhof(i), Depthf(i + XNu) / conl, Rhof(i + XNu) NEXT i
  IF XNn > XNu THEN LPRINT USING forma$; Depthf(Layers), Depthf(Layers), Rhof(Layers)
END IF
GOSUB DmpBrfxMsg

Whatnow:

VesTyp$ = "DGT"
NAPR = NA62
Replough = 0
FOR i = 1 TO NA62
  RhoDig(i) = RhoDigl(i)
  AB2(i) = AbDig(i)
  PsDb(i) = RhoDigl(i)
NEXT i
SETSCRN
LOCATE 6, 12
PRINT "Do you wish to:"
LOCATE 8, 12
PRINT "1. " : HIBHL " ENTER "
PRINT "sounding from the ";
HIBHL " SAME AREA."
LOCATE 10, 12
PRINT "2. " : HIBHL " LOAD "
PRINT "sounding from the ";
HIBHL " SAME AREA."
LOCATE 12, 12
PRINT "3. " : HIBHL " DISPLAY "
PRINT "digitized sounding to edit & reinterpret."
LOCATE 14, 12
PRINT "4. " : HIBHL " REINTERPRET "
PRINT "sounding from the ";
HIBHL " DIFFERENT OPTIONS."
LOCATE 16, 12
PRINT "5. " : HIBHL " RETURN TO MAIN MENU."
LOCATE 18, 12
PRINT "6. " : HIBHL " QUIT."
k = Prsn#br(6)
ON k GOTO ABUnit, LodVes, Plt, Options, MainMnu, Kwit

ChVsNembr:
IF Edt = 0 THEN RETURN
CLRLNE 1
LOCATE 1, 10: PRINT "Modify sounding number = "; VesNum$; " ? (Y/N/<Esc>)"
GOSUB YesNo
IF A$ = Es$ THEN GOTO MainMnu
IF A$ = "Y" THEN
    CLRLNE 1
    LOCATE 1, 10
    LINE INPUT "Enter NEW sounding number (4 char. max) ", A$
    GOSUB CapNTrim
    IF A$ = " " THEN BEEP: GOTO ChVsNembr
    VesNum$ = LEFT$(A$, 4)
    FOR i = LEN(TITLE$) TO 1 STEP -1
        IF MID$(TITLE$, i, 1) = " " THEN
            TITLE$ = LEFT$(TITLE$, i) + VesNum$
            i = 1
    END IF
    NEXT i
    FILENAME$ = LEFT$(Dir$, 4) + VesNum$ + "." + VesTyp$
END IF
RETURN

SaveIt:
IF DD$ = " " OR Dir$ = " " THEN
    CLRLNE 1
    BEEP
    LOCATE 1, 10
    PRINT " Can't Save! Drive or Directory not specified."
RETURN
END IF
GOSUB ChVsNabr
CLRLNE 1
LOCATE 1, 10: PRINT "SAVING ......."
OPEN DD$ + "\" + Dir$ + "\" + FILENAME$ FOR OUTPUT AS #1
PRINT #1, TITLE$
PRINT #1, ABUnit$
FOR i = 1 TO NAB2
    PRINT #1, AB2(i), PsOb(i)
NEXT i
CLOSE #1
CLRLNE 1
LOCATE 1, 1
PRINT TITLE$; " saved as: ";
PRINT "; FILENAME$; " Press Any Key to Continue"
WHILE INKEY$ = "": WEND
RETURN

FldPlt:
CRVPLT = true
GRAPHSET GR$
SETCORD AX(), AY(), AB2(), PsOb(i), NAB2
CALL Logplt(AX(), AY())
ABX(i) = ALBT(AB2(i)); PSY(i) = ALBT(PsOb(i))

PltCrv:
FOR i = 2 TO NAB2
    IF AB2(i - 1) < AB2(i) THEN
        ABX(i) = ALBT(AB2(i))
        PSY(i) = ALBT(PsOb(i))
        LINE (ABX(i) - 1), PSY(i) - i) - (ABX(i), PSY(i))
    ELSEIF AB2(i - 1) >* AB2(i) THEN
        ABX(i) = ALBT(AB2(i))
        PSY(i) = ALBT(PsOb(i))
    END IF
NEXT i
GOSUB PltCrl
LOCATE 2, 28
PRINT "o-o-o"
LOCATE 3, 26
IF VesTyp$ = "DBT" THEN PRINT "DIGITIZED CURVE" ELSE PRINT "FIELD CURVE"
GOSUB Lbl
CRVPLT = FALSE
RETURN

Wtr: A$ = INKEY$: IF A$ = "" THEN GOTO Wtr
RETURN

CapNtrm:
A$ = UCASE$(A$)
TEST$ = "" 'remove delimiters
FOR i = 1 TO LEN(A$)
Z% = ASC(MID$(A$, i, 1))
IF (Z% >= 48 AND Z% <= 57) OR (Z% >= 65 AND Z% <= 90) THEN TEST$ = TEST$ + CHR$(Z%)
NEXT i
A$ = TEST$
RETURN

'---------------------------------------------------------------

YsNo:
A$ = CAP$(A$)
IF A$ = "Y" OR A$ = "N" OR A$ = Es$ THEN RETURN
BEEP
GOTO YsNo

'---------------------------------------------------------------

NtEq1:
BEEP
CLR$5
CNTR 24, " Number of AB/2 is NOT EQUAL to number of APP.RHO!! "
PRT$ANY$Y
CLR$5
RETURN

'---------------------------------------------------------------

PrntrRdy:
CLRLNE 1
LOCATE 1, 5; PRINT "Make sure printer is on, Adjust Paper, press <ENTER> when ready"
WHILE INKEY$ <> CHR$(13): WEND
RETURN

'---------------------------------------------------------------

DipGrfxMsg:
CLRLNE 1
PRT$CR
RETURN

'---------------------------------------------------------------

IntrPlt:
Intrp = 1
GRAPH$ SET GR$
SETOORD AX(), AY(), AbDig(), Vesf(), Layers
CALL Logpl$(AX(), AY())
ABX(1) = ALGT(AbDig(1))): PSY(1) = ALGT(Vesf(1))
FOR i = 2 TO Layers
   ABX(i) = ALGT(AbDig(i))
   PSY(i) = ALGT(Vesf(i))
   LINE (ABX(i - 1), PSY(i - 1))-(ABX(i), PSY(i))
NEXT i
RAD = .03
FOR i = 1 TO Layers
   CIRCLE (ABX(i), ALGT(RhoDig(i))), RAD
NEXT i
LINE (ALGT(Depthf(i) / 10), ALGT(Rhof(i)))- (ALGT(Depthf(i)), ALGT(Rhof(i)))
FOR i = 2 TO Layers
   LINE -(ALGT(Depthf(i - 1)), ALGT(Rhof(i)))
   LINE -(ALGT(Depthf(i)), ALGT(Rhof(i)))
NEXT i
LOCATE 2, 28
PRINT "o o o", " "
LOCATE 3, 26
PRINT "DIGITIZED", "CALCULATED"
GO TO Lbl
Intrp = 0

RETURN

Lbl:

FOR i = 1 TO NAB2
    CIRCLE (ABX(i), PSY(i)), RAD
NEXT i
RETURN

Lbl2:

LOCATE 23, 79 - LEN(TITLE$) + 1: PRINT TITLE$ + "modified title"
LOCATE 25, 35: PRINT "ELECTRODE SPACING (AB/2)";
IF Intrp = 1 THEN PRINT ", OR DEPTH,  ";
PRINT "IN " + ABUnit$;
LOCATE 5, 17: PRINT "R"
LOCATE 6, 17: PRINT "E"
LOCATE 7, 17: PRINT "S"
LOCATE 8, 17: PRINT "I"
LOCATE 9, 17: PRINT "S"
LOCATE 10, 17: PRINT "T"
LOCATE 11, 17: PRINT "I"
LOCATE 12, 17: PRINT "V"
LOCATE 13, 17: PRINT "I"
LOCATE 14, 17: PRINT "T"
LOCATE 15, 17: PRINT "Y"
LOCATE 16, 17: PRINT "O"
LOCATE 18, 17: PRINT "H"
LOCATE 19, 17: PRINT "H"
LOCATE 20, 17: PRINT "M"
LOCATE 21, 17: PRINT "I"
LOCATE 22, 17: PRINT "M"
RETURN

DISPLAY:

SETSCRN
xx = 5: y = 4: i = 6: Q = 1
DsplABHdr:
  LOCATE 2, xx - 2: PRINT "#"
  LOCATE 2, xx: HIGHL " AB/2 ",

DsplABNibr:
  LOCATE y, X - 4: COLOR 3, 1: PRINT Q;
  LOCATE y, X: HIGHL STR#(AB2(i))
  Q = Q + 1
  IF i = NAB2 THEN
    y = 4
    xx = 18
    X = 18
    i = 1
    GOTO DsplRoHdr
  END IF
  GOTO DsplABNibr

DsplRoHdr:
  LOCATE 2, xx - 2: PRINT "#"
  LOCATE 2, xx
  HIGHL " APP.RHO "

DsplRoNibr:
  LOCATE y, X - 4
  COLOR 3, 1: PRINT Q
  LOCATE y, X
  HIGHL STR#(PsOb(i))
  IF i = NAPR THEN RETURN
  i = i + 1
  Q = Q + 1
  IF y = 18 THEN
    y = y + 1
  ELSE
    y = y + 1
  END IF
  GOTO DsplRoNibr

-----------------------------------------------

Kwit:

CLS
LOCATE 12, 20
PRINT "Are you sure you want to QUIT ? (Y/N/<Esc>)"
GOSUB YsNo
IF A$ = "Y" THEN GOTO MainMnu
IF A$ = "N" THEN GOTO MainMnu
ChkError:
  IF ERR = 5 AND CRVPLT THEN
    E = true
    RESUME NEXT
  END IF
  IF ERR = 11 THEN
    BEEP
    SETSCRN
    LOCATE 16, 21: PRINT "Division by Zero!"
    PRSSANYKY
    RESUME MainMnu
  END IF
  IF ERR = 25 THEN
    BEEP
    SETSCRN
    LOCATE 10, 10
    PRINT "PRINTER trouble! ....."
    PRSSANYKY
    RESUME MainMnu
  END IF
  IF ERR = 57 THEN
    CLS
    BEEP
    LOCATE 16, 21
    PRINT "Unformatted disk in selected drive!"
    PRSSANYKY
    RESUME DrvMnu
  END IF
  IF ERR = 64 THEN BEEP: RESUME LodFile
  IF ERR = 68 AND chkdirl = "CHKDRIVE" THEN
    BEEP
    LOCATE 16, 21
    PRINT "Drive (;; DD$; :) not found"
    CLOSE #1
    PRSSANYKY
    RESUME DrvMnu
  END IF
  IF ERR = 75 AND chkdir$ = "dir" OR ERR = 76 AND chkdir$ = "dir" THEN 'dir exists
    chkdir$ = **
    RESUME NEXT
  END IF
  IF ERR = 75 AND chkdir$ = "LOD" OR ERR = 76 AND chkdir$ = "LOD" THEN 'dir exists
    chkdir$ = **
    RESUME LodVes
  END IF
  IF ERR = 70 THEN
    CLS
    BEEP
    LOCATE 10, 20
    PRINT "DISK WRITE PROTECTED?"
    PRSSANYKY
    RESUME MainMnu
  END IF
  IF ERR = 71 THEN
BEEP
LOCATE 16, 21
PRINT "Drive ("; DD$; ") NOT READY, Door Open or No Disk in Drive!"
CLOSE #1
PRSSANYKY
RESUME DrvMnu
END IF
IF ERR = 53 AND chkdrive$ = "CHKDRIVE" THEN RESUME NEXT
IF ERR = 53 AND ChkFil = 1 THEN ChkFil = 0: RESUME NEXT
IF ERR = 53 THEN
BEEP
CLS
LOCATE 10, 16: PRINT "File NOT FOUND ...."
PRSSANYKY
RESUME LodVes
END IF
SETSCRN
PRINT "Can't do what you want. Error number = "; ERR, ERL
PRSSANYKY
RESUME MainMnu

'---------------------------------------------------------------------------

PrtAbDgt:
LOCATE y, X - 4: PRINT i
LOCATE y, X: PRINT AbDig(i)
RETURN

'---------------------------------------------------------------------------

ShftNDig:
CLRLINE 1
Isegint = 1
LOCATE 1, 1
LINE INPUT "Which segment remains fixed (Default is 1) ? <ENTER> ", Isegint$ , Isegint$ = VAL(Isegint$)
IF Isegint$ = "" THEN Isegint = 1
IF Isegint = 0 OR LEN(Isegint$) > 1 THEN BEEP: GOTO ShftNDig
NPTS = NAB2
CALL Shft(AB2(), PsObO(), NPTS, absh(), obsh(), nsh, Isegint)
FOR i = 1 TO nsh
    absh(i) = AL6T(absh(i))
    obsh(i) = ALGT(obsh(i))
NEXT i
temp = 0
CALL Spline1(nsh, temp, absh(), obsh(), A(), B(), C(), temp, D(), p(), S())
Delx = 1 / 6
CALL Cubic1(nsh, absh(), obsh(), A(), B(), C(), Delx, AbDig(), Ps(), NAB2)
FOR i = 1 TO NAB2
    AbDig(i) = 10 ^ AbDig(i)
    RhoDig(i) = 10 ^ Ps(i)
NEXT i
Radmin = AbDig(1)
Layers = NPTS
FOR i = 1 TO NAB2 - 1
    LINE (AL6T(AbDig(i)), ALGT(RhoDig(i)))-(AL6T(AbDig(i + 1)), ALGT(RhoDig(i + 1)))
NEXT i
IF DD$ = "" OR Dir$ = "" THEN
LOCATE 1, 1: PRINT "Save DIGITIZED curve ? (Y/N/ESC)?"  
6DSUB YesNo  
IF A$ = Es$ THEN GOTO MainMu  
IF A$ = "Y" THEN  
   GOSUB ChVsNmbr  
   FILENAME$ = LEFT$(FILENAME$, INSTR(FILENAME$, ")) + "DOT"  
CLRLNE 1  
LOCATE 1, 10: PRINT "SAVING ........."  
OPEN DD$ + "\" + Dir$ + "\" + FILENAME$ FOR OUTPUT AS 01  
PRINT 01, TITLE$  
PRINT 01, ABUnit$  
FOR i = 1 TO NAB2  
   PRINT 01, AbDig(i), RhoDig(i)  
NEXT i  
CLOSE 01  
CLRLNE 1  
LOCATE 1, 1  
PRINT TITLE$ + " saved as: " + FILENAME$ + " Press Any Key To Continue'"  
WHILE INKEY$ = "": WEND  
END IF  
RETURN  

SUB CHKGRAPH (E, GR$) STATIC  
   E = FALSE  
   SCREEN 9  
   IF NOT E THEN GR$ = "E": EXIT SUB  
   E = FALSE  
   SCREEN 10  
   IF NOT E THEN GR$ = "M": EXIT SUB  
   E = FALSE  
   SCREEN 2  
   IF NOT E THEN GR$ = "C": EXIT SUB  
   GR$ = "N"  
END SUB  

SUB GRAPHSET (GR$) STATIC  
IF GR$ = "C" THEN  
   CLS  
   SCREEN 2  
   VIEW (200, 45)-(630, 170)  
ELSE  
   CLS  
   IF GR$ = "E" THEN SCREEN 9: ELSE SCREEN 10  
   VIEW (200, 45)-(630, 300)  
END IF  
END SUB  

SUB UPSORT (A$(), N) STATIC  
   k = 2: M = N - 1  
SORT1: IF M < INT(N / 2) THEN EXIT SUB
IF A$(k - 1) > A$(M + 1) THEN SWAP A$(k - 1), A$(M + 1)
FOR i = k TO M
    IF A$(i) > A$(M + 1) THEN
        SWAP A$(i), A$(M + 1)
    ELSEIF A$(i) < A$(k - 1) THEN
        SWAP A$(i), A$(k - 1)
    END IF
NEXT i
k = k + 1
M = M - 1
GOTO SORT1
END SUB

' *************** ATOSUB.BAS MODULE ***************

DECLARE SUB HIGHL (A$)
DECLARE FUNCTION LOGAV (A$, B$)
DECLARE SUB CLRLNE (II)
DECLARE SUB CNTR (YT$, A$)
DECLARE FUNCTION PrssNbrZ (NZ)
DECLARE FUNCTION CAPNNT$ ()
DECLARE FUNCTION AL6T! (V.)
DECLARE FUNCTION DROUND! (txl!, tyl!)
DEFINT I-N
FUNCTION ALGT (X) STATIC
    ALGT = LDG(X) / LOG(10)
END FUNCTION
FUNCTION CAPNNT$ STATIC
    Cap:
        A$ = INKEY$: IF A$ = " THEN GOTO Cap
        IF A$ = CHR$(27) THEN GOTO cdone
        IF A$ = CHR$(13) THEN GOTO cdone
        A$ = UCASE$(A$)
    IF A$ < "A" OR A$ > "Z" THEN BEEP: GOTO Cap
cdone: CAPNNT$ = A$
END FUNCTION

SUB CLR5L STATIC
    COLOR 7, 1
    CLRLNE 21
    CLRLNE 22
    CLRLNE 23
    CLRLNE 24
    CLRLNE 25
END SUB

SUB CLRLNE (I) STATIC
    LOCATE I, 1: PRINT STRING$(79, " ");
END SUB

SUB CNTR (YT$, A$) STATIC
    IT = (80 - LEN(A$)) / 2
LOCATE YT, XT: HIGHL A$

END SUB

SUB Convex (Vv(), Ves(), Nrad, Xk(), Nc) STATIC
  M = 0
  L = 1
  L1 = Nc
Cnvlv:
  Vs = 0
  FOR J = L TO LI
    Vs = Vs + Vv(J) * Xk(J - M)
  NEXT J
  Ves(L) = Vs
  L = L + 1
  LI = LI + 1
  M = M + 1
  IF LI <= Nrad GOTO Cnvlv
END SUB

SUB Cubicl (H, Xo, Yo, Ao, Bu, Co, Delx, xx(), Yy(), K) STATIC
  Xiax = X(H)
  Xain = X(1)
  K = INT(DROUND((Xmax - Xmin) / Delx, 5)) + 1
  IF K < 1 THEN GOTO L9
  I = M
  xx(K) = Xmax
  Yy(K) = Y(M)
  FOR L = 1 TO K - 1
    J = K - L
    xx(J) = Xmax - (K - J) * Delx
  L7:
    IF DROUND(xx(J), 5) >= DROUND(X(I - 1), 5) THEN GOTO L8
    I = I - 1
    GOTO L7
  L8:
    Z = xx(J) - X(I - 1)
    Yy(J) = Y(I - 1) + ((C(I - 1) * Z + B(I - 1)) + Z + A(I - 1)) * Z
  NEXT L
  EXIT SUB
L9:
  PRINT "ERROR IN CUBIC1"
END SUB

SUB DELAY (B) STATIC
  A = THERT
  WHILE TIMER - A < B: WEND
END SUB

FUNCTION DROUND (tx1, ty1) STATIC
  DROUND = INT(.5 + (tx1 * (10 ^ ty1))) / (10 ^ ty1)
END FUNCTION

SUB HIGHL (A#) STATIC
  COLOR 0, 7
  PRINT A#;

COLOR 7, 1
END SUB

SUB Kernel (L, X(), t(), RO, VI), N) STATIC
  FOR J = 1 TO N
    B = 1
    A = -2 / X(J)
    FOR I = L TO 2 STEP -1
      C = B * R(I)
      D = A * t(I - 1)
      IF D < -10 THEN B = 1: GOTO nxt
      Q = (R(I - 1) - C) / (RI + 1) + C) * EXP(D)
      B = (1 - B) / (1 + Q)
    nxt: NEXT I
    V(J) = B * R(1)
  NEXT J
END SUB

DEFSN I-N
FUNCTION LOGAV (A, B) STATIC
  LOGAV = 10 ^ ((ALGTA(A) + ALGTA(B)) / 2)
END FUNCTION

SUB Logplt (AXO, AYO) STATIC
  xo = ALGTA(AXO): xI = ALGTA(AXIS))
  yo = ALGTA(AYO)): yI = ALGTA(AY(4))
  WINDOW (xo, yo)-(xI, yI)
  t = .065
  FOR I = 1 TO 5
    X = xo + I - 1
    LINE (x, yo)-(x, yI)
    FOR J = 2 TO 9
      x2 = ALGTA(x + J)
      LINE (x2, yo)-(x2, y0 + t)
      LINE (x2, y0 - t)-(x2, yI)
    NEXT J
  NEXT I
  t = .05
  FOR I = 1 TO 4
    Y = yo + I - 1
    yI = 10 ^ Y
    LINE (x0, Y)-(xI, Y)
    FOR J = 2 TO 9
      y2 = ALGTA(y1 + J)
      LINE (x0, y2)-(x0 + t, y2)
      LINE (x0 - t, y2)-(xI, y2)
    NEXT J
  NEXT I
END SUB

DEFINT I-N
SUB LVEO (A(), B(), N, t$, U$, K) STATIC
  INPUT II, t$
  INPUT II, U$
  FOR I = 1 TO 45
    ...
IF EOF(K) THEN EXIT FOR
INPUT #K, A(I), B(I)
NEXT I
N = I - 1
CLOSE #1
END SUB

SUB ONEILL (Xk(), N) STATIC
Xk(1) = -.000318
Xk(2) = .002072
Xk(3) = -.004978
Xk(4) = .01125
Xk(5) = -.02521
Xk(6) = .05812
Xk(7) = -.1436
Xk(8) = .393
Xk(9) = -1.1324
Xk(10) = 2.7044
Xk(11) = -3.4507
Xk(12) = .4248
Xk(13) = 1.1817
Xk(14) = .6194
Xk(15) = .2374
Xk(16) = .08688
Xk(17) = .0235
Xk(18) = .01284
Xk(19) = -.001198
Xk(20) = .003042
N = 20
END SUB

DEFINT I-N
SUB PRSSANYKY
LOCATE 25, 1: PRINT STRING$(78, " ");
CNTR 25, " Press Any Key to Continue ";
WHILE INKEY$ = "": WEND
END SUB

DEFINT I-N
FUNCTION PrssNmbrZ (N) STATIC
Prss:
LOCATE 25, 26
COLOR 0, 7: PRINT " Please Press Appropriate Number ";: COLOR 7, 1
A$ = INKEY$
WHILE A$ = "": A$ = INKEY$: WEND
K = VAL(A$)
IF K < 1 OR K > N THEN BEEP: GOTO Prss
PrssNmbrZ = K
END FUNCTION

SUB RHS1 (RhoDig(), Ves(), RMS, Layers) STATIC
RMS = 0
FOR I = 1 TO Layers
RMS = ((RhoDig(I) - Ves(I)) / RhoDig(I)) ^ 2 + RMS
NEXT I
RMS = (RMS / Layers) ^ .5 * 100
SUB SETCOORD (AX(), AY(), XI(), YI(), N) STATIC
AX(1) = 10 ^ INT(ALGT(XI(1)))
FOR I = 2 TO 5
AX(I) = AX(I - 1) * 10
NEXT I
IF AX(4) > XI(N) THEN
FOR I = 1 TO 5
AX(I) = AX(I) / 10
NEXT I
END IF
RhoMin = YI(1)
RhoMax = YI(1)
FOR I = 1 TO N
IF YI(I) < RhoMin THEN RhoMin = YI(I)
IF YI(I) > RhoMax THEN RhoMax = YI(I)
NEXT I
AVRHO = EXP((LOB(RhoMin) + LOG(RhoMax)) / 2)
AY(1) = 10 ^ INT(ALGT(AVRHO)) / 10
FOR I = 2 TO 4
AY(I) = AY(I - 1) * 10
NEXT I
END SUB

SUB SETSCRN STATIC
SCREEN 0
COLOR 7, 1
CLS
END SUB

SUB Shft (ab(), obO, npts, abshO, obshO, nsh, iseg) STATIC
DIM sgant(15)
J = 1
I = 1
absh(I) = ab(I)
obsh(I) = ob(I)
sgant = 1
amult = 1
sgant(J) = 1
sl = ab(I)
Ishift1:
I = I + 1
IF I > npts THEN GOTO doneshift
IF ab(I) > sl THEN
J = J + 1
absh(J) = ab(I)
obsh(J) = ob(I) * amult
sl = ab(I)
ELSEIF ab(I) = sl THEN
amult = obsh(J) / ob(I)
sgant = sgant + 1
sgant(sgant) = amult
ELSE
END IF
GOTO Ishift1

53
doneshift:

nsh = J
IF iseg >= nsgint OR iseg < 1 THEN EXIT SUB
I = nsgint - iseg + 1
amult = sgint(I)
FOR J = 1 TO nsh
  obsh(J) = obsh(J) / amult
NEXT J
END SUB

SUB Splinel (M, H, X(), Y(), A(), B(), C(), t, D(), P(), S()) STATIC
IF (t < 0) OR (t > 1) OR (H < 0) OR (H < 3) THEN GOTO L999
N = M - 1
IF t = 0 THEN GOTO L20
Ne = N - 1
IF H = 0 THEN GOTO L11
L1:
  Hh = 3 / H
  FOR I = 1 TO Ne
    B(I) = 4
    A(I) = 1
    C(I) = 1
    P(I) = Hh * (Y(I + 2) - Y(I))
  NEXT I
P(1) = P(1) - D(1)
P(Ne) = P(Ne) - D(2)
L3:
  C(I) = C(I) / B(I)
P(I) = P(I) / B(I)
FOR I = 2 TO Ne
  Mul = 1 / (B(I) - A(I) * C(I - 1))
  C(I) = Mul * C(I)
P(I) = Mul * (P(I) - A(I) * P(I - 1))
NEXT I
A(1) = P(1) - C(1) * A(1) + t
I = Ne - 1
L5:
  A(1 + t) = P(1) - C(1) * A(1 + t + 1)
  I = I - 1
IF I = 1 THEN GOTO L5
IF t = 0 THEN GOTO L6
A(1) = D(1)
A(N) = D(2)
L6:
IF H = 0 THEN GOTO L14
Hh = 1 / H
FOR I = 1 TO N
  Mul = Hh * (Y(I + 1) - Y(I))
  B(I) = Hh * (3 * Mul - (A(I + 1) + 2 * A(I)))
  C(I) = Hh * Hh * (-2 * Mul + A(I + 1) + A(I))
NEXT I
GOTO Rtn
L11:
FOR I = 1 TO N
  S(I + 1) = X(I + 1) - X(I)
NEXT I
FOR I = 1 TO Ne
    B(I) = 2 * (S(I + 1) + S(I + 2))
    C(I) = S(I + 1)
    A(I) = S(I + 2)
    P(I) = 3 * (S(I + 1) ^ 2 * (Y(I + 2) - Y(I + 1)) + S(I + 2) ^ 2 * (Y(I + 1) - Y(I))) / (S(I + 1) * S(I + 2))
NEXT I
P(1) = P(1) - S(3) * D(1)
P(Ne) = P(Ne) - S(N) * D(2)
GOTO L3
L14:
    FOR I = 1 TO N
        Hh = 1 / Sd + 1
        MuI = (Y(I + 1) - Y(I)) * Hh ^ 2
        B(I) = 3 * MuI - (A(I + 1) + 2 * A(I)) * Hh
        C(I) = -2 * MuI * Hh + (A(I + 1) + A(I)) * Hh ^ 2
    NEXT I
GOTO Rtn
L20:
    Ne = N + 1
    IF H = 0 THEN GOTO L31
    Hh = 3 / H
    FOR I = 2 TO N
        P(I) = Hh * (Y(I + 1) - Y(I - 1))
        B(I) = 4
        C(I) = 1
        A(I) = 1
    NEXT I
B(1) = 2
B(Ne) = 2
C(Ne) = 1
A(Ne) = 1
P(1) = Hh * (Y(2) - Y(1)) - .5 * H * D(1)
P(Ne) = Hh * (Y(N) - Y(N)) + .5 * H * D(2)
GOTO L3
L31:
    FOR I = 1 TO N
        S(I + 1) = X(I + 1) - X(I)
    NEXT I
    Ni = N - 1
    FOR I = 1 TO Ni
        B(I + 1) = 2 * (S(I + 1) + S(I + 2))
        C(I + 1) = S(I + 1)
        A(I + 1) = S(I + 2)
        P(I + 1) = 3 * (S(I + 1) ^ 2 * (Y(I + 2) - Y(I + 1)) + S(I + 2) ^ 2 * (Y(I + 1) - Y(I))) / (S(I + 1) * S(I + 2))
    NEXT I
B(1) = 2
B(Ne) = 2
C(1) = 1
C(Ne) = 1
A(1) = 1
A(Ne) = 1
P(1) = 3 * (Y(2) - Y(1)) / S(2) - .5 * S(2) * D(1)
P(Ne) = 3 * (Y(N) - Y(N)) / S(N) + .5 * S(N) * D(2)
GOTO L3
L999: PRINT "ERROR IN SPLIN1"

Rtn: END SUB
' *************** This is DRCT.BAS module ***************

DECLARE SUB CNTR (YT!, A$)
DECLARE SUB HIGHL (A$)
DECLARE FUNCTION tri$ (A$)
DECLARE SUB LCATE (iI, ioI, mZ)

TYPE regtype
  AX AS INTEGER
  bx AS INTEGER
  cx AS INTEGER
  dx AS INTEGER
  bp AS INTEGER
  si AS INTEGER
  di AS INTEGER
  flags AS INTEGER
END TYPE

DECLARE SUB interrupt (intnum AS INTEGER, INREG AS regtype, outreg AS regtype)

DEFINT I-N
SUB DIRECT (All), n, p$, k) STATIC
DIM INREG AS regtype, outreg AS regtype, f AS STRING * 126
i = 0
INREG.dx = VARPTR(f)
INREG.AX = &H1A00
CALL interrupt(&H21, INREG, outreg)
INREG.AX = &H4E00
p1$ = p$ + CHR$(0)
INREG.dx = SADB(p1$)
INREG.cx = &H255
CALL interrupt(&H21, INREG, outreg)
lupe:
IF outreg.AX AND &H1B THEN GOTO done
file$ = f$
at$ = MID$(file$, 22, 1)
m = ASC$(at$)
f$ = LEFT$(f$, 30) + "
  IF k <> 1 AND m <> 16 THEN GOTO c1
  i = i + 1
  A$(i) = tri$(MID$(file$, 31, 13))
c1: INREG.AX = &H4F00
      CALL interrupt(&H21, INREG, outreg)
GOTO lupe
done:
  n = i
END SUB

SUB FILESLCT (naf$, flfO, nf) STATIC
max = 126
IF nf = 0 THEN
  PRINT " no files exist"
  na$ = ""
  EXIT SUB
ELSE
  n = nf
  IF n > max THEN n = max
END IF
a = 1 + INT(n / 6): nof = 2: j = 1: k = 1
IF a > 21 THEN a = 21
ist = 1
iend = n
GOSUB 1st
lupp: A$ = INKEY$
WHILE A$ = "": A$ = INKEY$: WEND
L = LEN(A$): b = ASC(RIGHT$(A$, 1))
IF L = 2 AND b = 72 GOTO up
IF L = 2 AND b = 80 GOTO dwn
IF L = 2 AND b = 75 GOTO lft
IF L = 2 AND b = 77 GOTO rgt
IF L = 2 AND b = 71 GOTO hie
IF L = 2 AND b = 79 GOTO enn
IF L = 2 AND b = 73 GOTO pup
IF L = 2 AND b = 81 GOTO pdn
IF b <> 13 GOTO lupp
nail = ist + i - 1: EXIT SUB
GOTO lupp
up: LCATE i, nof, m: PRINT f1$(ist + i - 1): i = i - 1: IF i < 1 THEN i = n
LCATE i, nof, m: HIGHL f1$(ist + i - 1): GOTO lupp
dwn: LCATE i, nof, m: PRINT f1$(ist + i - 1): i = i + 1: IF i > n THEN i = 1
LCATE i, nof, m: HIGHL f1$(ist + i - 1): GOTO lupp
lft: LCATE i, nof, m: PRINT f1$(ist + i - 1): i = i - m: IF i < 1 THEN i = 1
LCATE i, nof, m: HIGHL f1$(ist + i - 1): GOTO lupp
rgt: LCATE i, nof, m: PRINT f1$(ist + i - 1): i = i + m: IF i > n THEN i = n
LCATE i, nof, m: HIGHL f1$(ist + i - 1): GOTO lupp
hie: LCATE i, nof, m: PRINT f1$(ist + i - 1): i = 1
LCATE i, nof, m: HIGHL f1$(ist + i - 1): GOTO lupp
enn: LCATE i, nof, m: PRINT f1$(ist + i - 1): i = n
LCATE i, nof, m: HIGHL f1$(ist + i - 1): GOTO lupp
pdn: IF iend = nf THEN GOTO lupp
ist = ist + nmax
iend = ist + nmax - 1
IF iend > nf THEN iend = nf
n = iend - ist + 1
m = 1 + INT(n / 6)
IF m > 21 THEN m = 21
GOSUB 1st
GOTO lupp
pup: IF ist = 1 THEN GOTO lupp
ist = ist - nmax: IF ist < 1 THEN ist = 1
iend = ist + nmax - 1
IF iend > nf THEN iend = nf
n = iend - ist + 1
m = 1 + INT(n / 6)
IF m > 21 THEN m = 21
GOSUB 1st
GOTO lupp
1st: CLS
CNTR 1, " SELECT A FILE OR SELECT EXIT TO RETURN ">
FOR i = ist TO iend
LCATE i - ist + 1, nof, m
PRINT f1$(i);
NEXT i
i = 1
SUB LCATE (i, io, a) STATIC
    n = INT((i - 1) / a)
    j = n * 13 + 1
    k = io + i - n * a
    LOCATE k, j
END SUB

SUB PRTSCR STATIC
    DIM INREG AS regtype, outreg AS regtype
    INREG.AI = &HS00
    CALL interrupt(&HS, INREG, outreg)
END SUB

FUNCTION trie$ (A$) STATIC
    A$ = UCASE$(A$)
    T$ = "remove spaces"
    FOR i = 1 TO LEN(A$)
        j = ASC(MID$(A$, i, 1))
        IF j >= 45 AND j <= 57 OR j >= 65 AND j <= 90 THEN T$ = T$ + CHR$(j)
    NEXT i
    trie$ = T$
END FUNCTION
APPENDIX B
DECLARE SUB UPSORT (A$, NZ)
DECLARE SUB ReadLYR (KZ, D!, R!, NZ, T$, u$)
DECLARE SUB LLYR (D!, R!, Layers*, DP!, RS!, NZ)
DECLARE SUB SLCTSOUND (D$, p$, S$, ul, NZ, KZ)
DECLARE SUB FILESCLT (NAMS$, fI$(), NZ)
DECLARE SUB DIRECT (A$, NZ, p$, KZ)
DECLARE SUB PASSANYKY ()
DECLARE SUB CLRME (i$)
DECLARE SUB SETSCRN ()
DECLARE SUB CNTR (YT!, R$)
DECLARE SUB HIGHL (A$)
DECLARE SUB SETCNT (c$(), nz$)
DECLARE SUB SETPATH (D$, p$, ierr$)
DECLARE FUNCTION ALG! (A!)
DECLARE FUNCTION LOGAV! (A!, B!)
DECLARE FUNCTION CAPWTS$ ()
DECLARE FUNCTION tria$ (A$)
DEFINT I-N
CONST FALSE = 0, TRUE = NOT FALSE, ndi = 40
DIM depth(ndi), resist(ndi), CONT(35), DP(100), RSUOO)
DIM CNT(100), DPC(100), FILES$(200)
ON ERROR GOTO Errorproc
SETSCRN
SETCNT CONT(), ncnt
M1: SETUPATH D$, dir$, ierr$,
IFRST = TRUE
M2: SLCTSOUND D$, dir$, FILES$(1, NF, NAM$, IFRST
IFRST = FALSE
IF NAMS = "DONE" THEN
SETSCRN
END
END IF
IF NAMS = "CHANGE DRIVE" THEN GOTO M1
inn = FREEFILE
OPEN D$ + "\" + dir$ + "\" + NAM$ FOR INPUT AS inn
ReadLYR inn, depth(), resist(), Layers, T$, unit$
CNTRES depth(), resist(), Layers, DP(), RS(), M
L = 0: i = 1: j = 1
M3: IF i >= M THEN GOTO M6
SLP = (ALGT(RS(i + 1)) - ALGT(RS(i))) / (ALGT(DP(i + 1)) - ALGT(DP(i)))
IF SLP > 0 THEN
M4: IF CONT(j) > RS(i + 1) THEN
i = i + 1
GOTO M3
ELSEIF CONT(j) > RS(i) AND CONT(j) <= RS(i + 1) THEN
L = L + 1
CNT(L) = CONT(j)
DPC(L) = 10^0 * (ALGT(DP(i)) + (ALGT(CONT(j)) - ALGT(RS(i))) / SLP)
END IF
j = j + 1
IF j > ncnt THEN
j = ncnt
i = i + 1
GO TO M3
ELSE
GO TO M4
END IF
ELSE
4a: IF i = 1 THEN j = ncnt
M5: IF CONT(j) < RS(i + 1) THEN
i = i + 1
GO TO M3
ELSEIF CONT(j) < RS(i) AND CONT(j) >= RS(i + 1) THEN
L = L + 1
CNT(L) = CONT(j)
DPC(L) = 10\* (ALGT(DP(j)) + (ALGT(CONT(j)) - ALGT(RS(i))) / SLP)
END IF
j = j - 1
IF j < 1 THEN
i = i + 1
GO TO M3
ELSE
GO TO M5
END IF
ELSE
5: SETSCRN
LPRINT : LPRINT : LPRINT '   + T*
LPRINT : LPRINT ' DEPTH IN CONTOUR 1
LPRINT '   * unit* + " VALUE": LPRINT
Fi = ' Hilt, I lillt.l*
6$: LPRINT USING F*; DPC(i), CNT(i)
NEXT i
LPRINT : LPRINT USING 6f; DP(H)
GOTO M2
END
Errorproc:
 ierr = TRUE
LOCATE 1, 70
HIGHL STR#(ERR)
SELECT CASE ERR
CASE 71
BEEP
CNTR 16, " DRIVE * + Dd$ + " not ready or door open 
CNTR 17, "press R to retry or S to select another drive"
K = INSTR("RS", UCASE\*INPUT\*1))
ON K = 1 GOTO e1, e2, e3
CASE 76
ierr = ERR
RESUME NEXT
CASE ELSE
PRINT "ERROR NUMBER "; ERR
RESUME M1
SUB CNTRES (D(), R(), N, DP(), RS(), M) STATIC
  j = 1
  DP(j) = 10! ^ (ALGT(D(j)) - .5 # (ALGT(D(2)) - ALGT(D(1))))
  RS(j) = R(1)
  FOR i = 1 TO M - 1
    j = j + 1
    DP(j) = D(i)
    RS(j) = LOGAV(R(i), R(i + 1))
    j = j + 1
    DP(j) = LOGAV(D(i), D(i + 1))
    RS(j) = R(i + 1)
  NEXT i
  M = j
  DP(M) = 10! ^ (ALGT(D(N - 1)) + .5 # (ALGT(D(N - 1)) - ALGT(D(N - 2))))
END SUB

SUB LLYR (D(), R(), N, T$, u$) STATIC
  SETSCRN
  LOCATE 5, 5: HIGHL T$: PRINT * UNIT= " + u$
  PRINT * DEPTH RESISTIVITY": PRINT
  FOR i = 1 TO N
    PRINT D(i), R(i)
  NEXT i
  PRSSANYKY
END SUB

SUB ReadLYR (K, D(), R(), N, T$, u$) STATIC
  INPUT IK, T$
  INPUT IK, u$
  INPUT IK, N
  FOR i = 1 TO N
    INPUT IK, D(i), R(i)
  NEXT i
  CLOSE IK
END SUB

SUB SETPATH (D$, drt, ierr) STATIC
  DIM dir$(ndim)
  setl: SETSCRN
  LOCATE 10, 10
  PRINT "type letter of DISK DRIVE "; HIGHL " (A,B,C...etc) "; PRINT " ";
  D$ = CAPNHTI
  PRINT D$
  CHDIR D$ + ":"
  IF ierr > 0 THEN
    BEEP
    CNTR 15, "DRIVE " + D$ + " not found"
    PRSSANYKY
    ierr = TRUE
    GOTO set1
  END IF
END
set2: SETSCRN
  CNTR 1, "SELECT THE DATA SUB-DIRECTORY"
  p$ = D$ + "\".
  DIRECT dir$(0), ndir, p$, 0
  ndir = ndir + 1
  dir$(ndir) = "CHANGE DRIVE"
  FILESLCT dr$, dir$(0), ndir
  IF dr$ = "CHANGE DRIVE" THEN GOTO set1
  dr$ = trim$(dr$)
END SUB

SUB SLCTSOUND (D$, p$, S$(0), NS, M$, K) STATIC
  IF NOT K THEN GOTO s1
  A$ = D$ + ".:" + RTRIM$(p$) + "\lyr"
  DIRECT S$(0), NS, A$, 1
  UPSORT S$(0), NS
  NS = NS + 1
  S$(NS) = "CHANGE DRIVE"
  NS = NS + 1
  S$(NS) = "DONE"
s1: SETSCRN
  CNTR 1, "SELECT A SOUNDING"
  FILESLCT NS$, S$(0), NS
END SUB

SUB UPSORT (A$(0), N) STATIC
  K = 2: M = N - 1
SOR1:  IF M < INT(N / 2) THEN EXIT SUB
  IF A$(K - 1) > A$(M + 1) THEN SWAP A$(K - 1), A$(M + 1)
  FOR i = K TO M
    IF A$(i) > A$(M + 1) THEN
      SWAP A$(i), A$(M + 1)
    ELSEIF A$(i) < A$(K - 1) THEN
      SWAP A$(i), A$(K - 1)
    END IF
  NEXT i
  K = K + 1
  M = M - 1
  GOTO SOR1
END SUB
DECLARE SUB PRSSAWKY ()
DECLARE SUB SETSCRN ()
DECLARE SUB CNTR (YT!, A#)
DECLARE SUB HI6HL (A#)
DECLARE FUNCTION CAPNUT* ()
DEFINT I-N

SUB SETCNT (c(), nc) STATIC
  c(1) = 1!
c(2) = 1.5
c(3) = 2!
c(4) = 3!
c(5) = 4.5
c(6) = 7!
c(7) = 10!
c(8) = 15!
c(9) = 20!
c(10) = 30!
c(11) = 45!
c(12) = 70!
c(13) = 100!
c(14) = 150!
c(15) = 200!
c(16) = 300!
c(17) = 450!
c(18) = 700!
c(19) = 1000!
c(20) = 1500!
c(21) = 2000!
c(22) = 3000!
c(23) = 4500!
c(24) = 7000!
  nc = 24
END SUB