Southwest Fisheries Center Administrative Report H-87-20

HAWAIIAN ISLAND MAPPING PROGRAM

(HIMP)

Jerry L. Fuqua
Southwest Fisheries Center Honolulu Laboratory
National Marine Fisheries Service, NOAA
Honolulu, Hawaii 96822-2396

December 1987

NOT FOR PUBLICATION

This Administrative Report is issued as an informal document to ensure prompt dissemination of preliminary results, interim reports, and special studies. We recommend that it not be abstracted or cited.

CONTENTS

Page

1 OVERVIEW OF SOFTWARE 1

1.1 Purpose of Software 1

2 PROGRAM DESCRIPTION 3

2.1 General Description 3

2.2 Program Structure 4

3 HIMP OPERATING INSTRUCTIONS. 5

3.1 Overview. 5

3.2 Data-File Format. 5

3.3 Program File Requirements 7

3.4 Program Initiation. 7

3.5 The Opening Menu. 8

3.6 Opening Menu Options. 9

3.6.1 Island Group Options (Options 1-5) 9

3.6.2 Individual Isles (Hi Res) (Options 6-7). 10

3.6.3 Freeform Entry (Option 8). 12

3.7 Setting Map Limits (Scanning) 12

3.8 Adding Depth Contours 15

3.9 Adding Station Locations. 16

3.10 Adding Data 18

3.11 Continuation or Output Menu 20

3.12 Real-Time Plot Initiation 21

3.13 Base Map Plotting 22

CONTENTS (Continued)

Page

3.14 Station Location Plotting 23

3.15 Data Plotting 23

3.16 Plot Enhancements 25

4 MAPPING EXAMPLES 26

4.1 Introduction. 26

4.2 Example 1--Basic Oahu Map 26

4.3 Example 2--Modified Group Map 31

4.4 Example 3--High Resolution Island and Data Display. 37

4.5 Example 4--Freeform Map Development 49

4.6 Example 5--Trackline Map Development. 57

5 I/O DATA FILES 66

5.1 Introduction. 66

5.2 Cartographic Data Files 66

5.3 Map Image Files 67

5.4 Research Data Files 69

5.5 Data Types for Research Data Files 69

5.6 Station Location Option 70

5.7 File Construction and Coordinate System Constraints. 71

LIST OF APPENDIXES

Appendix I
HIMP Detailed Flowchart. 75

Appendix II
HIMP Subroutine List 78

Appendix III
Program Listing 79

Appendix IV
BASIC Digitizing Program 134

Appendix V
HIMP Hard Disk Installation Instructions. 138

1 OVERVIEW OF SOFTWARE

1.1 Purpose of Software

The Hawaiian Island Mapping Program (HIMP) (Fig. 1) is a system of graphics and plotting modules that develops screen images and high quality plots of maps for visually presenting research data in the appropriate geographic setting. As its name indicates, the program is geared specifically for creating maps of the Hawaiian Islands; however, it can be used for mapping other areas. The image of any single island or group of islands in the Hawaiian Archipelago can be retrieved by the HIMP from the system map library and displayed at various levels of resolution. Optionally, the program uses an external file containing digitized map information to produce the corresponding screen image. After an image is displayed, the boundary settings can be modified to create a new customized map, and data added to this map to produce a composite display image. A hard copy of the resulting map can be produced on a Hewlett-Packard[1] 7475A plotter. The plotting module allows the map image to be further modified, or enhanced through control over plot size, line types, colors, labels, and symbols.

---

[1] Reference to trade names does not imply endorsement by the National Marine Fisheries Service, NOAA.

==========

SOFTWARE NAME: HIMP

STATUS: Operational

DESCRIPTIVE NAME: Hawaiian Island Mapping Program

CURRENT VERSION: 1.3
VERSION DATE: 04 11 87

SOFTWARE TYPE: Graphics/plotting development

CLASSIFICATION: Data/map image composition

COMPUTER SYSTEM: IBM PC/XT/AT

LANGUAGE: Turbo Pascal (version 3.1)

DATA BASE USED: Digitized maps and/or user supplied data

DATA SYSTEM: Map image retrieval library

PROCESSING MODE: Interactive

COMPILER: Turbo Pascal Version 3.1

DATA STORAGE: Hard disk or floppy

MONITOR: EGM or CGM

GRAPHICS SUPPORT:

HARDWARE: EGA or CGA screen displays, Hewlett-Packard 7475A plotter (6 pen), Epson FX printers

SOFTWARE: Turbo Graphix Toolbox (Borland), Graphics (IBM,MS) or Pizazz (Application Techniques)

MINIMUM MEMORY: 256K

REQUIRED EXECUTABLE FILES:
himp.com, himp.000. . . himp.005. (7 files total)

REQUIRED SUPPORT FILES:
error.msg, 14x9.fon, 4x6.fon, 8x8.fon

AVAILABLE INSTALLATION FORMATS: 1.2M floppy or (2) 360K floppies. Source code available separately.

==========

Figure 1.--Description of HIMP software.

2 PROGRAM DESCRIPTION

2.1 General Description

As an aid in creating custom-made map images and publication quality plots, the HIMP (Appendix I) consists of two development sections, one screen oriented and one plotter oriented. The screen-oriented section consists of two development phases, one for basic map modification and one for enhancement and customization. In the first phase, a map image based on user's specifications is produced, and in the second, new cartographic features, such as contours or research results, can be added. The plot development section allows for further modifications and printing of the resulting image. This section has several plot enhancement features.

Once initiated, HIMP retrieves and displays any map image in the system library, which is located in the subdirectory HMAP. Alternately, a user-supplied external file can be used to draw and display a given geographic feature. After the map image is displayed, the boundaries can be modified interactively to form a new image. Thus, the user can change the shape and orientation of the map and "zoom" in on or out of a geographic area, relative to the original image. During this process, map image data are written to an output file for plotting or processing in subsequent HIMP sessions.

After the basic map image is formed, contour data from an external file can be used to draw additional map features. The new cartographic information is incorporated into the screen image and the corresponding data appended to the output map file. These appended data are preceded by a flag that allows use of a different line style when contour data are encountered during the map plotting phase. The resulting composite output file may be accessed, via the external file retrieval option, by HIMP in a later session.

The final stages of screen development involve station location and data displays. The HIMP uses data files containing records with three entries each. The first two records, which are not used by the program, may contain any information the user wishes to retain and have transferred to the output file. Subsequent records specify the location, given by the first two entries, and the value of each recorded data element. Three types of data input files can be used with HIMP. The first type is station or catch data; i.e., the records in which a location and a data value are listed for each research station. The second type is trackline data; i.e., a series of location listings that represent a continuous path. The entry following each location listing contains a value that can invoke the display of a symbol. The third type is longline data; i.e., a series of paired records giving the starting and ending locations of each line. In this file, the third entry can represent information associated with each line. For each type of data, the third entry usually can be plotted, subject to user-specified constraints, which can be invoked through a data partitioning option. This option is illustrated in section 4 and further detailed in section 5.

For station locations, HIMP uses only the location information and displays a user-selected symbol at each position. With the data display option, a user-selected symbol is employed to denote the position and actual data value for each element. If desired, the actual data value may be used as the data symbol; data also may be partitioned into groups or bins and assigned symbol sizes by value.

Following the screen-oriented development of a map image, HIMP produces a corresponding map plot for which the user can specify plot size as well as grid lines or tick marks to correspond to intervals of degrees, minutes, and seconds. During the plotting of cartographic features, alternate pens and line styles may be selected, allowing specification of the line type, color, and thickness. For displaying station locations and data, new symbols and sizes may be selected, and for data display, the partitioning of data values changed. The plot may be enhanced with coordinate labels for any or all of the four corners. Finally, repetitive plot labeling is fully supported.

2.2 Program Structure

The HIMP has a somewhat complicated system structure because of the inherent coding limitations of Turbo Pascal. The present version of the Turbo Pascal compiler, version 3.01, uses 64 Kbyte code and data segments. As a result, the individual source code, data allocation, and executable code are limited to 64 Kbytes of memory. These constraints necessitated the development of a highly modular form for the HIMP program structure.

To conform to source code limitations, HIMP is partitioned into a main program (himp.pas) and three support or inclusion modules (Appendix II). These modules, himplot.pas, himputil.pas, and himpsele.pas contain subroutines that are made accessible to the main program through the inclusion statement at the beginning of the program. The himpplot.pas module contains all of the plotting routines and plot support utilities. Himptuil.pas supplies the general utility and maintenance procedures that are used continuously by the main program. Himpsele.pas contains the option-selecting procedures that are accessed at the user's discretion. In addition to the above inclusion files, one communication and four graphics files are designated in the main program inclusion statement. Asynchp.inc supplies the communication utilities for sending data and plot commands to the Hewlett-Packard plotter. Accessing the Turbo Graphix library uses the files typedef.sys, graphix.sys, kernel.sys, and windows.sys. For proper execution, these four graphics files must appear in the inclusion section of the main program source code. Upon compilation, all inclusion files are incorporated into the executable code.

To keep the executable code within the corresponding 64 Kbyte limit, all moderate-to-large subroutines are designated as overlay files. During compilation, memory is allocated within the executable code of the main program or module to accommodate the largest overlay file. Stored on disk in overlay files are all consecutive overlay routines. There are six such files in HIMP: himp.000, himp.001, himp.002, himp.003, himp.004, and

himp.005. When HIMP calls an overlay routine, the executable code is retrieved from the corresponding overlay file and loaded into the designated memory allocation where it is executed. Note that because there are six overlay files, there also are six memory allocation areas within the executable code.

In the basic executable form, HIMP must have four run-time support files--error.msg, 14x9.fon, 4x6.fon, and 8x8.fon--which supply HIMP with execution error message facilities and graphics font support. The entire executable system is listed in the HIMP directory:

himp.com	himp.000	himp.001
himp.002	himp.003	himp.004
himp.005	error.msg	14x9.fon
4x6.fon	8x8.fon	

Additionally, map library files, data files with the .bin extension, and image files with the .map extension are present in the \HMAP subdirectory.

Available upon request are distribution diskettes that also include data files in ASCII form and all sources files. Prior to any modification to the source code, efficiency should be considered because the 64 Kbyte restrictions are quite constraining for the present version of HIMP. Liberal use of inclusion files and overlay techniques insures against exceeding memory limits.

3 HIMP OPERATING INSTRUCTIONS

3.1 Overview

The HIMP is a graphics system that is menu-prompt driven. At each step, the user is prompted for a response after the display of a statement or menu. The program proceeds through three developmental stages. The first stage produces the basic map image and data file incorporating limits defined by the user. The second stage can create input data files for displaying depth contours, station locations, and data on the basic map. Also created at this time are the corresponding output data files. The final stage uses the newly created data files to produce a hard copy of the map on a Hewlett-Packard plotter. During this stage, plot enhancements and labeling can be added. The basic process of map development is illustrated through the operational flowchart in Figure 3-1. A more detailed version of this flowchart is in Appendix I.

3.2 Data-File Format

The HIMP uses two types of data files. The first contains digitized map and contour information and consists of sequential records with two entries each listing the coordinates. The first record gives the minimum and maximum latitudes of the data set, while the second record gives the

```
                 ____________________________
                [        OPENING MENU        ]
                [    (8 Mapping Options)     ]
                [____________________________]
                              ||
                              \/
 ____________________________________________________________
[            BASE MAP CREATION AND DEVELOPMENT               ]
[     1. Retrieve selected map files (.bin and .map).        ]
[     2. Display map image.                                  ]
[     3. Scan image and set new map limits.                  ]
[     4. Draw new map image.                                 ]
[____________________________________________________________]
                              ||
                       (Basic Map Image)
                              \/
 ____________________________________________________________
[                    MAP DEVELOPMENT                         ]
[     1. Add depth contours to map image.                    ]
[     2. Add station locations to map image.                 ]
[     3. Add research data to map image.                     ]
[____________________________________________________________]
                              ||
                     (Developed Map Image)
                              \/
                 ____________________________
                [ CONTINUATION (OUTPUT) MENU  ]
                [ (1) Plot output map         ]
                [ (2) Create output plot file ]
                [ (3) Exit HIMP               ]
                [____________________________]
                     ||          ||        ||
                 (Option 1) (Option 2) (Option 3)
                     ||          ||        |_=========> EXIT
                     ||          ||
                     ||          |_=======> OUTPUT PLOT FILE
                     \/
 ____________________________________________________________
[                    PLOTTER OUTPUT                          ]
[     1. Plot basic map.  (Optional tick marks.)             ]
[     2. Plot station locations and/or data on map.          ]
[     3. Add enhancements.  (Labels.)                        ]<=||
[____________________________________________________________]  ||
                              ||                                ||
                              \/                                ||
                  ____________________      (New Map)         ||
                 [ EXIT or MAKE NEW MAP ]===============_|
                 [____________________]
                              ||
                              ||
                              \/
                            (Exit)
```

Figure 3-1.--The HIMP flowchart.

minimum and maximum longitudes. Subsequent records list coordinates by latitude first and longitude second; this conforms to current NOAA, NMFS data recording protocol. The second type of data file contains station location and research data and consists of records containing three data fields or entries. At present, the first two are dummy records and not used by HIMP, although they may contain useful information, such as minimum and maximum latitudes and longitudes. (If so, the third entry then is arbitrary, and for convenience, zeros can be used for the third entry.) Subsequent records would consist of the latitude, longitude, and value for each record. The order of coordinate components conforms to the current NOAA, NMFS protocol.

Normally with both map-contour files and station data files, HIMP uses input data and produces output data that are in Turbo binary form. These files have the .bin file extension. However, if an input file is from an external source and not the system library, a standard ASCII file can be used. Although the extension for this type of file is arbitrary, the .dat extension is recommended to help denote the ASCII form. The program produces a working binary version of the ASCII file, and subsequent output files also are in binary form. In addition, HIMP generates files containing the graphics image corresponding to the map-contour files. The files generated by the HIMP are in binary form and use the .map extension to identify their contents. For a more detailed description of the data-file formats used by HIMP, refer to section 5.

3.3 Program File Requirements

The HIMP normally is installed in its own directory, \HIMP, with the system map library in the subdirectory \HMAP. In addition to the com file himp.com, the six overlay files himp.000, himp.001, himp.002, himp.003, himp.004, and himp.005 must be present in \HIMP for the program to execute properly. The subdirectory \HMAP normally contains all the binary data files (.bin) and graphics image files (.map) for each map option made available through the HIMP selection menus. User-supplied files do not have to be present in either the \HIMP directory or the \HMAP subdirectory and may exist on any device that is accessible through DOS. For convenience, such files should be placed in the \HIMP directory before use.

For modifications, the source code for the main HIMP program and supporting inclusion modules is available. However, compilation of the modified source code must be supported by several Turbo Pascal Graphix inclusion modules and by an asynchronous communications inclusion module. In addition, several Turbo Pascal accessory files must be present. For information on modifying the source code refer to Appendix III.

3.4 Program Initiation

From \HIMP or an appropriate alternate directory, where himp.com is located, the user may initiate the mapping program by typing "HIMP" and then pressing the carriage return (<CR>) key. After the HIMP banner

appears, a window is displayed in the upper right corner of the screen with the message "HIT RETURN TO COMMENCE." This window is known as the HIMP communication window, henceforth referred to as the comm window, which is normally used by HIMP to prompt the user for responses. The HIMP communicates with the user either via the comm window or through a main screen display. The appropriate response for any option presented is indicated in brackets; any other response causes the program to continue to the next development step. At this time, press <CR> to start HIMP and bring up the opening menu.

3.5 The Opening Menu

The opening menu (Fig. 3-2) displays the eight options for map development made available by HIMP. "Windward" in the HIMP system refers to the main Hawaiian Islands; "leeward" refers to the Northwestern Hawaiian Islands. The desired option may be selected by typing the corresponding number. As with multiple option selections in general, no <CR> is required. If a key other than 1-8 is typed, the message "AVAILABLE OPTIONS ARE (1)-(8)" appears above the menu. Control is transferred immediately to the appropriate module, where the basic map image is developed. Upon its completion, control returns to the main HIMP program for further modification and development.

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\                                                 \
\  Select Hawaii Mapping Option:                  \
\                                                 \
\    (1) Complete archipelago (Low Res)           \
\    (2) Windward group (Med Res)                 \
\    (3) Leeward group (Med Res)                  \
\    (4) Windward group (Hi Res)                  \
\    (5) Leeward group (Hi Res)                   \
\    (6) Individual Windward Isle (Hi Res)        \
\    (7) Individual Leeward Isle (Hi Res)         \
\    (8) Freeform entry                           \
\                                                 \
\ SELECT BY NUMBER                                \
\                                                 \
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

Figure 3-2.--The HIMP opening menu.

The first seven options allow the user to construct charts, with various levels of resolution, based on digitized map files from the system map library. The first option begins with a low resolution map image of the entire Hawaiian Archipelago. Options 2 and 3 begin with a medium resolution of the map images of the Windward and the Leeward Islands, and options 4 and 5 represent a high resolution version of the corresponding

groups. This is done by accessing the individual high resolution files required for each island. Options 6 and 7 transfer control to secondary menus where the user can select an individual island from the corresponding group, and HIMP opens the corresponding individual high resolution files.

When selecting an option, the user should consider the total chart area and resolution required. To construct a chart of an island group covering a large geographic area, a medium or low resolution option should be used. Selecting a high resolution option results in a substantial increase in relative processing time, without an obvious gain in information or visual clarity for both the screen display and plotter output.

Option 8 allows the use of digitized map data from a source other than the system map library. These data can be in Turbo binary form or in standard ASCII representation. If a map image does not exist, this option produces the corresponding image file. Once created, the map image serves as the base map on which new images may be produced. Input files can even be output files from previous HIMP sessions. This option also allows the user to install improved versions of library maps, both .bin and .map files, by using the appropriate output name.

3.6 Opening Menu Options

Depending on which option is selected, control branches and takes a particular map development path. After basic development is completed, all paths return to main HIMP control for further modification and development. The following subsections describe the eight different option paths. Each option incorporates a scanning or limit-setting procedure to aid in defining the basic output map image. This procedure is described in detail in section 3.7.

3.6.1 Island Group Options (Options 1-5)

Options 1-5 are initiated by typing the corresponding number key indicated in the opening menu. Option 1 enables the user to work with the library files containing the low resolution digitized data for the complete Hawaiian Archipelago. Options 2 and 3 use the medium resolution files for the Windward and Leeward groups, respectively. Options 4 and 5 also use the medium resolution files for the Windward and Leeward groups, but only in the initial phase in which the user determines the map boundary limits. Once the limits are determined, the individual island high resolution files are used to produce a new map image.

The HIMP asks for an output filename, without extension, for creating the appropriate .bin and .map files for the base map image. Typing <CR> accepts the default and produces the files MapOut.bin and MapOut.map. The program displays the names of the output map files, then prompts the user, via the comm window, for a <CR> to continue.

The HIMP first displays the image frame and lists its limits in the upper left corner of the screen. A subsequent <CR> displays the map image corresponding to the selected option. Regardless of the resolution level selected, the actual screen image at the given geographic scale has a relatively high apparent resolution with respect to digitizing density. The user may view the map at this point, then continue with development by pressing <CR>. A message in the comm window asks whether new map limits are desired, indicating that a "Y" invokes this option. Typing any other key causes the program to continue without any modification of the map limits. If modification of the map limits is selected, HIMP initiates a scanning process in which each limit is set sequentially; i.e., the right limit (minimum longitude), left limit (maximum longitude), lower limit (minimum latitude), and then upper limit (maximum latitude). The scanning or limit-setting procedure is common among each map development option (opening menu options 1-8) and is described in detail in section 3-7.

Whether or not the map limits are modified, a new map image is drawn on the screen and simultaneously recorded into the corresponding map output (.bin and .map) files. Note that in contrast to the medium resolution maps (options 2 and 3), the high resolution maps (options 4 and 5) are based on the individual map files of high resolution for the Windward and Leeward groups. Data are drawn from each island file that falls within the limits of the desired map image. The comm window then asks whether to continue development or redo the base map. As indicated, entering a "Y" continues development, whereas an "N" returns the program to the original map for a new start. Once the desired base map is complete, control returns to the main HIMP processing sequence for further development and modification.

3.6.2 Individual Isles (Hi Res) (Options 6-7)

These options are initiated by typing "6" or "7" at the opening menu. For option 6, a secondary menu appears listing the 8 Windward Islands that may be selected (Fig. 3-3), and for option 7, listing the 10 Leeward Islands (Fig. 3-4). Typing the corresponding number selects the desired island. As with the opening menu, no <CR> is required. Using a key other than the indicated keys is responded to with the message "AVAILABLE OPTIONS ARE (1)-(8) (or (0))" being displayed above the menu.

The Leeward Island menu contains an additional notation in the listings. When depths are indicated by the "nnf" notation, the depth contour line is included in the map, otherwise very little coastline is present to produce a representative map or the reef structure is the dominate geographical feature present.

A valid response opens the corresponding island (.bin and .map) input files of high resolution. The user is prompted for an output filename for creating the appropriate .bin and .map files for the base map image. Typing <CR> accepts the default and produces the files MapOut.bin and MapOut.map. The program displays the names of the output map files and then prompts the user, via the comm window, for a <CR> to continue.

```
\\\
```

development or redo the base map. As indicated, entering a "Y" continues development, whereas an "N" returns the program to the original island map for a new start. Once the desired base map is complete, control returns to the main HIMP processing sequence for further development and modification.

3.6.3 Freeform Entry (Option 8)

Typing "8" at the opening menu selects option 8. With this option, the user is asked for the input map filename. Note that the full name, including extension, must be used. This file can be in either ASCII or Turbo binary; therefore, a prompt is issued inquiring about the type. An "A" indicates ASCII and causes the program to produce a binary file from the ASCII file. The message "The binary input file 'filename' has been created and will be used for input by HIMP" is issued after completion of the binary file. A "B" indicates that the file is binary and no new file will be created. (It is recommended that the user adopt the convention of using .dat for ASCII map files and .bin for Turbo binary map files.)

A new prompt asks whether the map image (.map) file already exists. A "Y" tells the program that the image exists and can be displayed on the screen directly. An "N" (or other key) tells the program that the input map must be drawn first and then stored in the .map file. Following this, HIMP asks for an output filename, without extension, which is used to create the appropriate .bin and .map files for the base map image. Typing <CR> accepts the default and produces the files MapOut.bin and MapOut.map. The program displays the names of the output map files in either case and then prompts the user, via the comm window, for a <CR> to continue.

The HIMP first displays the image frame and lists the frame limits in the upper left of the screen. A subsequent <CR> results in the map image being drawn or displayed on the screen. A message appearing in the comm window asks whether the map should be modified. A "Y" causes the program to ask whether to set new map limits and indicates that a "Y" or <CR> invokes this option. Selecting this option enables the user to set each of the four map limits by the limit-setting procedure described in section 3.7. An "N" bypasses the limit-setting process. For either option, the user is queried: "Y" indicates the program should continue further development, whereas typing "N" indicates the program should remodify the original map. Typing any other key causes continuation to further map development, similar to the "Y" response.

3.7 Setting Map Limits (Scanning)

With each option, after the original library or user-supplied map has been displayed, the program issues the message "SET NEW MAP LIMITS [Y or CR]?" in the comm window. Selecting this option allows the user to modify the map by designating new limits on the screen. Typing "Y" or <CR> initiates the process. Typing any other key indicates that the present map image is acceptable and ready for further development. The following outline describes the limit-setting procedure used to modify the boundaries of the original map.

The limit-setting process involves four distinct steps; i.e., moving and setting the right (minimum longitude), left (maximum longitude), lower (minimum latitude), and upper (maximum latitude) limits. In each step, messages in the comm window guide the user. Initially, the comm window displays the following instructions:

```
****************************************
* SET MAP LIMITS                       *
* USE CURSOR CONTROLS TO MOVE LIMITS   *
* WHERE SHIFT-<ARROW> ==> 10X<ARROW>   *
* AND SPACE BAR SETS THE LIMITS.       *
* HIT RETURN TO CONTINUE.              *
****************************************
```

These instructions indicate that pressing the appropriate cursor (arrow) key moves a given limit line laterally or vertically. Pressing a cursor key causes the line to move in single-unit steps, whereas holding the shift key down while pressing a cursor key causes movement in 10-unit steps. Once a limit line is placed at the desired location, its position can be fixed and recorded by pressing the space bar.

Each limit can be moved anywhere within the map image bounds. In addition, limits can be set beyond the original map image boundaries by continuing to use the cursor keys after the line has reached the original map limit. However, once the line passes the original boundary, it no longer is displayed. Its position, even off the screen, can be estimated from the number of cursor steps entered. Caution should be used when setting limits outside the original boundaries, because support of this feature is dependent on the location values of the original map files and the finite integer limits of the Pascal compiler. Normally the range beyond the original boundaries is at least equal to about half the corresponding image dimension. Thus, if an original map has a lateral dimension of 20 cm, one can expect that new limit lines could be set 10 cm or more beyond either the left or right map image boundary.

Once the user has read the opening instructions, a <CR> starts the limit-setting procedure. The comm window displays the message

```
SET LONGMIN    (RIGHT LIMIT)
HIT RETURN TO COMMENCE
```

to indicate that the first limit to be set is the right or minimum longitude boundary. After a <CR> is entered, a vertical line appears in the middle of the map display. Using the cursor keys moves the line left or right to the desired position, and pressing the space bar fixes its location on the screen and displays its position in degrees in the upper left corner. At the same time, a prompt for continuation is issued in the comm window. Entering a <CR> allows the program to proceed to the next step.

The message in the comm window

SET LONGMAX (LEFT LIMIT)
HIT RETURN TO COMMENCE

indicates that the second limit line to be set is the left or maximum longitude boundary. A <CR> causes a second vertical line to appear in the middle of the display. By following the same procedure, this line can be set and fixed on the screen, with its position displayed in the upper left corner and a continuation prompt issued in the comm window. A <CR> advances the user to the next step.

The new comm window message

SET LATMIN (LOWER LIMIT)
HIT RETURN TO COMMENCE

indicates that the lower or minimum latitude boundary is to be set. A <CR> produces a horizontal line in the middle of the map display. Using the cursor keys moves this line vertically to the desired location, where it can be fixed by pressing the space bar. The new limit position is displayed in degrees in the upper left corner, and a prompt for continuation is issued.

After a <CR>, the comm window message

SET LATMAX (UPPER LIMIT)
HIT RETURN TO COMMENCE

indicates that the upper or latitude maximum boundary can now be set. After entering a another <CR>, a second horizontal bar appears in the middle of the display. Once it is positioned and set, its position value is displayed in the upper left corner, and the user is prompted to continue.

At this point, the limit-setting process is complete, and the user is prompted for a response to continue. A <CR> causes the new image frame to be displayed with the new limits listed in the upper left corner. Another continuation prompt is issued, and after a <CR> has been entered, the system issues the message

READY FOR NEW MAP
HIT RETURN TO COMMENCE

to indicate that the map is ready to be drawn. Another <CR> results in a display of the new map.

3.8 Adding Depth Contours

Following construction of the base map image, HIMP allows the user to add contours to the image. The comm window displays the message "DISPLAY CONTOURS, FROM EXTERNAL FILES, ON MAP [Y]?" A "Y" response indicates that HIMP is to add depth contours to the map image, and any other key bypasses this option.

The HIMP displays the current (or default) line-style number, which is line-style number 4: *** *** *** ***. The user is then asked to enter the appropriate input filename, complete with extension. The filename also can include the DOS path and device name, such as A:mydir\myfile.dat. Depth contour files should have the same format as map files, with two-entry records in either ASCII or Turbo binary. If HIMP cannot find the file, a message is issued to that effect, and the user is requested to enter a new filename. Once the file is found, the program asks whether it is an ASCII or a binary file. Entering an "A" for ASCII causes HIMP to produce a binary input file from the ASCII file before continuing. Upon completion of this operation, a message is displayed, indicating that the binary file has been created for input. Entering a "B" or any other key causes this step to be bypassed.

At this point, the program determines whether the user wishes to change the line style, by displaying the message "Select new line style [Y]?". A "Y" indicates that a new line style is desired, otherwise the current style is kept. If the user selects this option the following menu is displayed:

```
*********************************************
*   SELECT LINE STYLE:                      *
*   (0)   *************************         *
*   (1)   *   *   *   *   *   *   *         *
*   (2)   *****   *****   *****             *
*   (3)   ***  *  ***  *  ***  *             *
*   (4)   *** *** *** *** *** ***            *
*********************************************
```

The desired line style may be selected by pressing the corresponding number key. After the style has been selected, a continuation prompt is displayed.

By entering a <CR>, the program continues by displaying the base map image and then drawing the contour line(s) onto the image. The corresponding digitized data are flagged and appended to the output map files, both the .bin and .map files. A new message asking the user whether more contours should be added is issued. A "Y" repeats the above process and allows the user to add additional depth contours. Typing any other key

causes HIMP to leave this option and continue on to further map development.

3.9 Adding Station Locations

After the user is given the opportunity to display depth contours, HIMP issues the prompt "DISPLAY STATION LOCATIONS ON MAP [Y]?" via the comm window. A "Y" initiates this option, allowing the user to display results from a specified data file. This file should have two leading records containing three arbitrary entries each. After these leading records should be a series of three-element records listing latitude, longitude, and data values for each station.

The program prompts the user for this file with the message "Enter Input Filename (with extension)." The filename must include the extension and can contain the DOS path and device designation. If the given filename does not correspond to an existing available file, the message "Cannot find data file (filename)" is displayed, and the user is asked to supply a new name. When the file is found, the system issues the message "Is input data file in ASCII [A] or Pascal binary (.bin) [B]?" An "A" causes the corresponding binary file to be created from the ASCII file, and the message "Input Station File: (filename)" is issued. Next, the program asks for the output filename by issuing the message "Enter Output Station Filename (no extension) [OutStn]: (Binary File with Station Locations)." A <CR> causes the program to use the default name OutStn. The HIMP responds with the message "Output Station File: (filename.bin)."

The system issues a continuation prompt to which a <CR> enables the selection of a symbol for the given stations. The following menu is displayed:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\  SELECT DISPLAY SYMBOL TO DENOTE STATIONS    \
\     (1)     +                                \
\     (2)     x                                \
\     (3)     box                              \
\     (4)     fbox                             \
\     (5)     diamond                          \
\     (6)     Y                                \
\     (7)     *                                \
\     (8)     o                                \
\     (9)     Station Number                   \
\   other     User-defined (keyboard) symbol   \
\  SELECT BY NUMBER OR BY KEYBOARD CHARACTER   \
\                    FOR USER-DEFINED SYMBOL   \
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

In this menu, option 4 is a symbol consisting of a diamond within a box; option 9 uses sequential numbers to denote the stations. Selecting option

9 causes the program to request the first station number from the user. Subsequent stations are numbered in sequence from this number. Any of the numbered options can be selected by pressing the corresponding number key. Pressing any other key results in the display of the message "ENTER KEYBOARD CHARACTER TO BE USED AS DISPLAY SYMBOL." The user can then respond by pressing the desired key.

After the desired symbol is selected, the comm window displays the following message asking the user to confirm the symbol selection:

```
*********************************************************
* SELECTED SYMBOL IS                                    *
*                        (SYMBOL)                       *
*                                                       *
* Accept this selection [Y] or make new selection [N]? *
*********************************************************
```

The "(SYMBOL)" represents the symbol selected by the user. If the option selected is 9, the character "N" is displayed to indicate that station numbers are used as symbols. Entering an "N" returns the user to the symbol menu for a new selection. A "Y" (or other response) causes the comm window to display the following message soliciting the scale size for the symbol:

```
*********************************************************
* SELECTED SYMBOL IS                                    *
* WITH SCALE SIZE 1       (SYMBOL)                      *
* Select new scale size (integer multiple)              *
* or accept current scale (hit non-integer key)         *
*********************************************************
```

"(SYMBOL)" represents the symbol selected and is initially at scale size 1. Pressing any numeric key from 1 to 9 selects the corresponding scale size and displays the symbol with that scale. A 0-entry displays a point, representing the symbol with a zero dimension. The symbol may be viewed at different scale sizes by repeatedly pressing the desired numeric keys. Pressing any nonnumeric key causes the program to accept the current selection.

Currently, the scale size selection option is limited by the current Turbo Pascal compiler and graphics support. Scales are available only in multiples of the primary (1) scale. In practice, scales above 3 have limited utility because of the somewhat inhibitingly large size of the symbols relative to the overall size of the map. Thus, only scale 1 is accurately represented in the later plotting facilities. Larger scales are represented in proportion to scale 1 and not as multiples of it.

After the symbol and corresponding scale size have been selected, the map image reappears and the station locations are drawn on top. Only the stations within the image boundary are displayed. Although the station locations are displayed on top of the map, the location data do not actually become a part of the map image file. Instead, the locations are stored in the output file, which can be accessed by the plotting routines later.

3.10 Adding Data

After a short pause, the comm window appears with the message "DISPLAY DATA ON MAP [Y]?" A "Y" response enables the user to display, on the map image, data from an appropriate file, whereas any other response bypasses this option. After a "Y" is entered, the program issues the following inquiry:

WHAT IS THE DATA TYPE?
(1) Station/catch
(2) Trackline
(3) Longline

The reply may be issued by entering the corresponding key. Option 1 refers to data collected at individual stations or locations. Trackline data, option 2, are in the form of a continuous, single curved line or path, with symbols being displayed along the path depending on the the third (data) entry of each record. Longline data, option 3, are represented by a series of straight lines between sets of data points. The program then indicates the number entered and the data type selected. For example, pressing "1" results in the following display:

STATION/CATCH DATA

For more information on the input data formats, refer to section 5.

The user is now queried for the name of the input data file, with extension. As before, the filename specification may include the DOS path and any device designation. The query is repeated if the file corresponding to the filename cannot be located. The program then asks for the file type to be supplied, with "A" indicating ASCII and "B" binary. An "A" results in the creation of a binary file from the corresponding ASCII file. Finally, the system asks for an output data filename. Entering a <CR> accepts the default OutData.bin.

After a continuation prompt, the user is asked to select a data symbol:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\  SELECT DISPLAY SYMBOL TO DENOTE DATA VALUES \
\    (1)     +                                  \
\    (2)     x                                  \
\    (3)     box                                \
\    (4)     fbox                               \
\    (5)     diamond                            \
\    (6)     Y                                  \
\    (7)     *                                  \
\    (8)     o                                  \
\    (9)     Data Value                         \
\   other    User-defined (keyboard) symbol     \
\  SELECT BY NUMBER OR BY KEYBOARD CHARACTER    \
\                    FOR USER-DEFINED SYMBOL    \
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

Option 4 is a symbol consisting of a diamond within a box, and option 9 uses the actual data values as display symbols. Selecting option 9 causes the program to request the number of significant figures to be used. Entering 0 indicates that only the integer part should be used as the symbol. Any of the numbered options can be selected by pressing the corresponding number key. Pressing any other key results in the display of the message "ENTER KEYBOARD CHARACTER TO BE USED AS DISPLAY SYMBOL." The user can then respond by pressing the desired key.

Once the desired symbol is selected, the comm window display asks the user to confirm the symbol selection:

```
**********************************************************
* SELECTED SYMBOL IS                                     *
*                          (SYMBOL)                      *
*                                                        *
* Accept this selection, or make new selection [N]?     *
**********************************************************
```

A "Y" confirms the current symbol selection, whereas an "N" returns the user to the symbol option menu.

After the desired symbol is selected, the program displays "Select Symbol Scale Proportional to Data Values [Y]?" This option may be invoked by pressing "Y." The program responds by asking for the number of data bins desired: "Select number of data bins [1,2,3,4, or 5]." The number and a <CR> should be entered. The system asks for the minimum and maximum values of the first bin. Scale selection for this bin is initiated through the comm window:

```
*********************************************************
* SELECTED SYMBOL IS                                    *
* WITH SCALE SIZE 1        (SYMBOL)                     *
* Select new scale size (integer multiple)              *
* or accept current scale (hit non-integer key)         *
*********************************************************
```

Any scale (0-9) can be selected, and the currently selected symbol, denoted above by "(SYMBOL)," displayed to that scale by pressing the corresponding numeric key. Viewing the scales in this way may be done repeatedly until a nonnumeric key is pressed to accept the current scale size for the given bin.

Depending on the number of bins selected, the program continues to query the user for the maximum value of each bin and, simultaneously, the corresponding scale size for that bin. If the proportional symbol option was not initially invoked, no queries for bin number or data range are made, and only one scale size is requested.

After the data symbol is selected, with or without the proportional scaling option, the map image is displayed and the data drawn on top. Note that station locations do not appear. This is to avoid image congestion with the data symbols. Only data whose positions are within the map image boundary appear. After a brief pause, the comm window appears with a continuation prompt. A <CR> allows the user to leave the screen development section of HIMP and continue to the continuation or output menu. Here, the user may leave HIMP or select one of two (at present only one) plotting options.

3.11 Continuation or Output Menu

After viewing the screen image of the composite map, the user can activate the continuation menu by entering a <CR>.

```
//////////////////////////////////////////////////////////
/ SELECT PLOT OPTIONS:                                  /
/   (1) Plot Output Map--Plotter must be physically     /
/                        attached and ready.            /
/   (2) Create Output Plot File--For transport to       /
/                                plotter (Pascal Driver). /
/   (3) Exit HIMP                                       /
//////////////////////////////////////////////////////////
```

Selecting option 1 puts the user into the real-time, plot development section of HIMP, allowing plots to be developed on a Hewlett-Packard plotter. (This option is currently in operation and fully supported.) Selecting option 2 creates a postprocessing file for developing plots on a Hewlett-Packard plotter at a later time. (Currently this option is not

supported). Option 3 allows the user to exit HIMP directly. Although no postprocessing file is created, all output files are saved and accessible for future development.

3.12 Real-Time Plot Initiation

The plot routine sequence is initiated with HIMP opening a communication port to a Hewlett-Packard plotter. If the port is successfully opened, a message to that effect is displayed briefly. Note that the port can be opened and a "true" response indicated even when the plotter is not on. Turbo Pascal and, thus, HIMP cannot check to determine whether an external device is operational, so make sure the plotter is ready before proceeding.

Before actual plotting commences, HIMP displays a message indicating the current setting of the plot factor. This factor, initially set at 0.9, indicates the fraction of the maximum possible plot area, for standard-sized (8 1/2" X 11") paper, that will be used for the map. The default factor produces a plot that is 90% of the maximum plot area. The user is asked whether the plot should be made with the default plot factor or a new plot factor. To select a new factor, "Y" should be pressed. The system then requests the user to enter a new value between 0.0 and 1.0, then press <CR>.

The plotter first draws four small scaling points, one at each corner, to indicate the maximum physical plot area. The map border is then drawn with the current plot factor. The system proceeds to ask whether tick marks or grid lines should be drawn for longitude and latitude on degree, minute, and second divisions. Requests are only issued for a division if the scale and location of the map can accommodate it. For example, a map of a small island or ocean feature may not contain a longitude degree division within its lateral limits. If so, the user would not be asked if longitude degree tick marks are desired.

The tick-mark option is selected by entering a <CR> in response to the appropriate prompt, then a menu listing four plotting options appears. These options allow the tick marks to be placed at either or both borders, or to have grids plotted. The longitude degree tick-mark option initiates the following menu:

```
//////////////////////////////////////////////////////////
/    LONGITUDE TICK-MARK OPTIONS:                        /
/      (1) Degree tick marks along bottom border.        /
/      (2) Degree tick marks along top border.           /
/      (3) Degree tick marks along top and bottom borders./
/      (4) Degree grid lines.                            /
/      or  No tick marks or grid lines.                  /
//////////////////////////////////////////////////////////
```

The desired option is selected by pressing the corresponding numbered key. Pressing any key other than 1-4 causes no degree ticks or grids to be drawn. After the option is selected, the system inquires about the interval, "Desired tick-mark interval (No. degrees per tick)." The interval should be entered and followed by a <CR>. All other tick-mark options--longitude and latitude in degrees, minutes, and seconds--follow this format. The corresponding tick marks decrease in size from degrees to seconds.

3.13 Base Map Plotting

Once the map boundary and supporting tick marks or grid lines are plotted, the program is ready to draw the base map. First, the user is asked whether a new pen (color) or line type or both are to be used. This allows map features to be drawn in colors (or thicknesses) other than those used to draw the border and ticks. For pen selection, the following prompt appears:

```
SELECT NEW PEN NUMBER?
(No. 1-6. Or keep current pen.)
```

Pressing any 1-6 key selects that pen number; pressing any other key keeps the current selection. Pen selection is followed by the line selection prompt "SELECT NEW LINE TYPE [Y]?" A "Y" brings up the line-type selection menu:

```
//////////////////////////////////////////////////////////
/  SELECT NEW LINE TYPE:                                 /
/  (1)  .      .      .      .                          /
/  (2)  ____   ____   ____                              /
/  (3)  _____  _____  _____                             /
/  (4)  _______._______._______.                        /
/  (5)  _______ _ _______ _ _______ _                   /
/  (6)  ____ _ _ ____ _ _ ____ _ _                      /
/  or_____________________________                      /
//////////////////////////////////////////////////////////
```

Pressing any 1-6 key selects the corresponding line type, whereas entering any other key selects the solid line type.

As soon as the pen and line type are determined (or left unchanged), the plotter draws any coastlines present in the output map file created during the HIMP screen development session. When the program comes to any data representing depth contours, plotting stops and the user is asked whether a new line type should be used. A "Y" brings up the line-type menu from which a new type may be selected. For each set of depth contour data encountered, the line-type option is made available to the user. Thus,

several different line types may be displayed on the plot to help distinguish various depths.

3.14 Station Location Plotting

Once the base map is plotted, HIMP determines whether a station location file was created during the screen development phase. If such a file had been created, the system will ask the user whether the locations should be plotted onto the base map. A "Y" response causes the system to invoke this option and ask whether a new pen selection is desired. Another "Y" response allows a new pen to be selected by pressing the corresponding numeric key. Next, the station plot symbol, which corresponds to the selected station screen symbol, is displayed in the comm window, and the user asked if a new symbol should be selected. The symbol may be changed through the symbol selection menu by using any response other than "Y" or <CR>. The comm window then indicates the current default plot symbol size and asks whether to keep it or select a new size. To keep the current size, enter "Y"; to select a new size, enter "N." If a new size is to be selected, the user is asked for the symbol width and height in centimeters. Note that because of the protocol of the Hewlett-Packard, the height should be 1.44 times the width to produce a symbol with an aspect ratio of 1. A symbol specification of 1.0 cm X 1.44 cm actually produces a symbol with a physical size of 1.0 X 1.0 cm. The user should refer to the Hewlett-Packard HP 7475A Graphics Plotter Interfacing and Programming Manual for details.

The comm window displays the message "READY TO PLOT STATION LOCATIONS--HIT RETURN TO COMMENCE." A <CR> causes the plotter to plot the station locations, with the selected symbol and the specified size, onto the base map. Each symbol is centered at the plot position corresponding to the given geographic map location.

3.15 Data Plotting

If a data output file had been created during the screen development phase of HIMP, the user is asked whether the data should be plotted onto the map. A "Y" response invokes this option and causes the system to ask whether a new pen should be selected for the data. Responding with the corresponding numeric key selects a new pen. Next, the system displays the current data symbol and asks whether to use this symbol for the plot or select a new one. A "Y" response or <CR> accepts the current data symbol, whereas any other key brings up the symbol menu for a new selection.

If the data have been partitioned previously, the following message appears:

```
DATA ARE CURRENTLY PARTITIONED INTO n BINS.
KEEP CURRENT PARTITIONING OR SELECT NEW PARTITIONS [Y]?
```

The number of previously selected bins is represented by n. A response of "Y" allows the user to select a different number of data bins and define their ranges. The program selection of the size of the symbols representing each bin by assigning the width and height to the smallest symbol and an incremental size increase for the remaining symbols. This procedure starts with the statement

```
CURRENT DEFAULT PLOT SYMBOL SIZE IS  w X h
+ 0.04W X 0.058H CM INCREMENT INCREASES.
```

(Note: The 1.44 h/w ratio produces a plot aspect ratio of 1.)

```
KEEP DEFAULT OR SELECT NEW SIZE PARAMETERS [Y]?
```

The w X h denotes the smallest bin character size in the character format of the Hewlett-Packard plotter. If a "Y" is entered, the system asks for the symbol width and then height. To determine the incremental increase, the system issues the statement:

```
ASPECT RATIO FOR SYMBOLS (H/W) IS h/w
INCREMENTAL INCREASE WILL BE dW IN WIDTH
dH = (H/W)*dW IN HEIGHT.
ENTER INCREMENT dW (CM) + RETURN:
```

Here, h/w denotes the actual aspect ratio. The increment increase can be entered at this time.

If no partitioning was performed in the screen display section, the user is asked whether partitioning is desired: "PARTITION DATA BEFORE PLOTTING." A "Y" response allows the user to select the number of partitions, their range, and plot size via the procedure given above.

Before the data are plotted, the user is asked whether a one-character offset is desired. This is useful if station locations have already been plotted and the data values are to be used as the data symbols, thus preventing the location symbol from interfering with the data value. One representation using this option might look like this: "* 123." If the data are of the trackline or longline type, the user is asked whether a new line type is desired. If a "Y" response is made, a new line type may be selected from the line-type menu. The program will then draw the data on top of the map. All new trackline data solicit a system query to determine whether a new line type is desired. If it is, the line-type menu is used for the selection. After the data are plotted, HIMP offers two other optional plot enhancements.

3.16 Plot Enhancements

The first plot enhancement option is labeling the corner coordinates. A "Y" reply to the corresponding prompt brings up the following menu:

(1) LABEL UPPER LEFT CORNER
(2) LABEL LOWER LEFT CORNER
(3) LABEL LOWER RIGHT CORNER
(4) LABEL UPPER RIGHT CORNER
OR CONTINUE

Pressing a given numeric key invokes the corresponding label option. Successive options may be invoked by using these keys. All corner labeling is performed with the same pen that produced the original border lines and any tick marks or grid lines. Once the desired corners are selected, pressing any nonoption designated key lets the program proceed to the second enhancement option.

The second plot enhancement option is the labeling of the plot itself. This is a repeatable option that starts with the simple prompt "Label Plot [Y]?" A "Y" invokes this option. The system first asks whether a new pen is to be chosen, for which a "Y" response will allow selection via the pen menu. The next prompt asks for the number of lines to be used; this number should be entered and followed with a <CR>. The default character size of 0.187 w X 0.269 h is displayed, and the user asked whether to accept the default or select new size parameters. A "Y" invokes new parameter selection, whereas any other key accepts the default size. For new size parameters, the program asks for the character (symbol) width and then height in centimeters. Remember that the ratio of h/w should be 1.44 for a physical aspect ratio of 1.

To determine the placement of the label, the following message is issued:

Center of first label line located midway between left and
right paper boundaries, and 1/10 down from top boundary.
Accept current location or select new location [Y]?

These boundaries refer to the maximum physical area that can be plotted, as denoted by the four small points created prior to the drawing of the map border. If a position other than the default position is desired, a "Y" should be entered. The system then asks for the fractional (lateral) distance from left to right, and for the fractional (vertical) distance from bottom to top, to place the center of the first label line. When entered, the position is displayed. If more than one line is drawn, the system displays the default drop distance to the next (and successive lines) and asks whether to accept the default or select a new distance. A "Y" response allows the user to specify a new drop distance.

The label is plotted, then the system asks whether additional labels should be made. If more labels are desired, a "Y" response repeats the process. After labeling is finished, the user is asked whether a new plot is desired. A "Y" returns the user to the initial plotting position, from which a completely new plot may be created from the same map, station, and data files. Entering any other key exits HIMP.

4 MAPPING EXAMPLES

4.1 Introduction

The following mapping examples illustrate the features supplied by HIMP to facilitate map development. The first example presents the basic procedure for producing a simple island map of Oahu. Successive examples produce maps of increasing complexity and use more advanced developmental techniques. In all examples, the HIMP prompts and displays are listed on the left side of the page, and user responses appear on the right. The comm window messages are denoted by a (*) marker preceding the statement. Comments are enclosed in double parentheses [e.g. ((Map being drawn on display screen.))].

4.2 Example 1--Basic Oahu Map

Example 1 outlines the procedure for making a "plain vanilla" map of Oahu. The final plotted map will reflect the data and image of the high resolution Oahu files contained in the HIMP system library, which is found in the \HMAP subdirectory.

```
=============================================================
HIMP                (*) Hit Return to Commence   |   <CR>
.............................................|...........
Select Hawaii Mapping Option:                    |     6
  (1) Complete archipelago (Low Res)             |
  (2) Windward group (Med Res)                   |
  (3) Leeward group (Med Res)                    |
  (4) Windward group (Hi Res)                    |
  (5) Leeward group (Hi Res)                     |
  (6) Individual Windward Isle (Hi Res)          |
  (7) Individual Leeward Isle (Hi Res)           |
  (8) Freeform entry                             |
SELECT BY NUMBER                                 |
.............................................|...........
Select Island:                                   |     6
  (1) Hawaii                                     |
  (2) Maui                                       |
  (3) Kahoolawe                                  |
  (4) Lanai                                      |
  (5) Molokai                                    |
```

```
  (6) Oahu
  (7) Kauai
  (8) Niihau
SELECT BY NUMBER
```

```
Enter Output Data Filename (No Extension):          <CR>
[Default --> "MapOut"]
```

```
Output Data File: MapOut.bin
Output Map File:  MapOut.map
(*) HIT RETURN TO CONTINUE                          <CR>
```

```
PRESENT MAP LIMITS
  Longmin:  1.5752000000E+02
  Longmax:  1.5838300000E+02  ((MAP FRAME))
  Latmin:   2.1238812500E+01
  Latmax:   2.1778187500E+01
(*) HIT RETURN TO CONTINUE                          <CR>
```

```
(*) HIT RETURN TO CONTINUE                          <CR>
```

```
(*) SET NEW MAP LIMITS [Y]                          <CR>
```

```
PRESENT MAP LIMITS
  Longmin:  1.5752000000E+02
  Longmax:  1.5838300000E+02  ((MAP FRAME))
  Latmin:   2.1238812500E+01
  Latmax:   2.1778187500E+01
(*) READY FOR NEW MAP
(*) HIT RETURN TO COMMENCE                          <CR>
```

```
          ((Map being drawn on display screen.))
```

```
(*) CONTINUE, OR REDO MAP [N]?                      <CR>
```

```
(*) DISPLAY CONTOURS, FROM
    EXTERNAL FILES, ON MAP [Y]?                     <CR>
```

```
(*) DISPLAY STATION LOCATIONS ON MAP [Y]?           <CR>
```

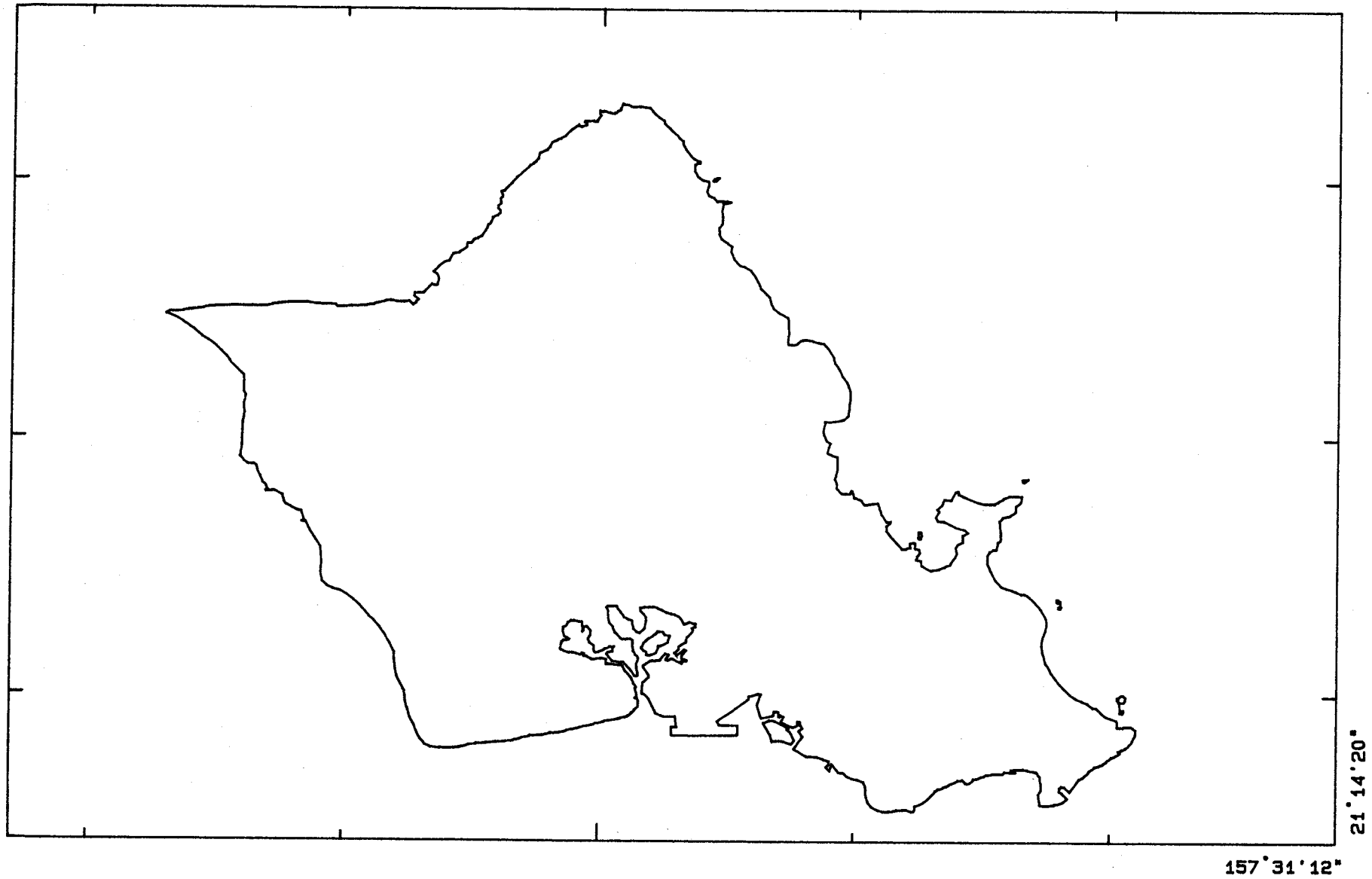
```
(*) DISPLAY DATA ON MAP [Y]?                        <CR>
```

(*) HIT RETURN TO CONTINUE	<CR>
SELECT PLOT OPTION: (1) Plot Output Map--Plotter must be physically attached and ready (2) Create Output Plot File--For transport to plotter (Pascal Driver) (3) Exit HIMP	1
Port Status = TRUE Plot factor is set at 0.9. Proceed with this value or select new value [Y]?	<CR>
((Plotter draws map frame.))	
Draw Longitude Degree Tick Marks [Y]?	Y
(1) Degree tick marks along bottom border (2) Degree tick marks along top border (3) Degree tick marks along top & bottom (4) Degree grid lines OR No tick marks or grid lines	3
Desired tick mark interval (No. degrees/mark)	1 <CR>
Draw Longitude Minute Tick Marks [Y]?	Y
(1) Minute tick marks along bottom border (2) Minute tick marks along top border (3) Minute tick marks along top & bottom (4) Minute grid lines OR No tick marks or grid lines	3
Desired tick mark interval (No. minutes/mark)	10 <CR>
Draw Latitude Minute Tick Marks [Y]?	Y
(1) Minute tick marks along left border (2) Minute tick marks along right border (3) Minute tick marks along left & right (4) Minute grid lines OR No tick marks or grid lines	3
Desired tick mark interval (No. minutes/mark)	10 <CR>
Draw Longitude Second Tick Marks [Y]?	<CR>
Draw Latitude Second Tick Marks [Y]?	<CR>
SELECT NEW PEN NUMBER? (No.: 1-6. Or keep current pen.)	<CR>

```
..................................................|...........
SELECT NEW LINE TYPE [Y]?                         |   <CR>
..................................................|...........
              ((Plotting of coastline(s).))
..........................................................
Label corner coordinates [Y]?                     |     Y
..................................................|...........
(1) LABEL UPPER LEFT CORNER                       |     3
(2) LABEL LOWER LEFT CORNER                       |   <CR>
(3) LABEL LOWER RIGHT CORNER                      |
(4) LABEL UPPER RIGHT CORNER                      |
OR  CONTINUE                                      |
..................................................|...........
Label Plot [Y]?                                   |     Y
..................................................|...........
FOR LABEL: SELECT NEW PEN NUMBER?                 |   <CR>
  (No.: 1-6. Or keep current pen.)                |
..................................................|...........
Number of label lines:                            | 1 <CR>
..................................................|...........
Default character size is 0.187 w X 0.269 h (cm)  |   <CR>
Accept default, or select new size parameters     |
                                        [Y]?      |
..................................................|...........
Center of first label line located midway          |   <CR>
between left and right paper boundaries and       |
1/10 from top boundary.  Accept current           |
location, or select new location [Y]?             |
..................................................|...........
Enter text for line 1:                            | OAHU <CR>
..................................................|...........
WRITE ADDITIONAL LABEL [Y]?                       |   <CR>
..................................................|...........
Make a new plot [Y] (or exit)?                    |   <CR>
..................................................|...........
FINIS                                             |
==========================================================
```

Note that because no depth contours, station locations, or data are displayed during the screen development phase, the queries for corresponding plot routines are not issued. No query for plotting the latitude degree tick marks is issued because the map area does not contain a degree latitude line. In labeling corner coordinates, typing a "3" selects the lower right corner. Entering a <CR> terminates the corner label selection and continues the plot development.

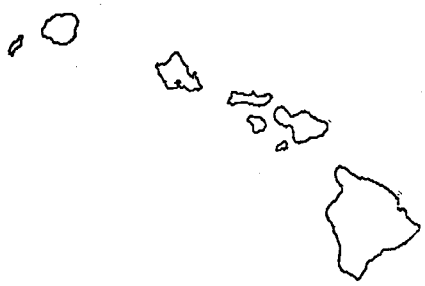
OAHU



4.3 Example 2--Modified Group Map

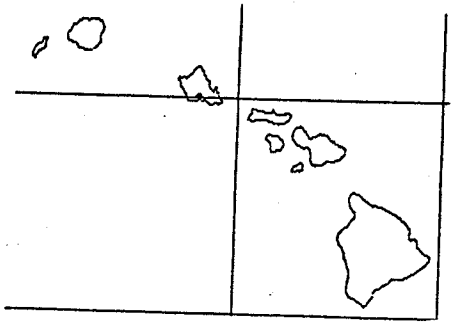
Example 2 illustrates how to use a library map and modify the limits to produce a new submap.

```
=============================================================
HIMP                (*) Hit Return to Commence   |   <CR>
.............................................|...........
Select Hawaii Mapping Option:                    |     2
  (1) Complete archipelago (Low Res)             |
  (2) Windward group (Med Res)                   |
  (3) Leeward group (Med Res)                    |
  (4) Windward group (Hi Res)                    |
  (5) Leeward group (Hi Res)                     |
  (6) Individual Windward Isle (Hi Res)          |
  (7) Individual Leeward Isle (Hi Res)           |
  (8) Freeform entry                             |
SELECT BY NUMBER                                 |
.............................................|...........
Enter Output Data Filename (No Extension):       | MHC <CR>
[Default --> "MapOut"]                           |
.............................................|...........
Output Data File:  MHC.bin                       |
Output Map File:   MHC.map                       |
                   (*) HIT RETURN TO CONTINUE    |   <CR>
.............................................|...........
PRESENT MAP LIMITS                               |
  Longmin:  1.545083000E+02                      |
  Longmax:  1.607964000E+02  ((MAP FRAME))       |
  Latmin:   1.860260875E+01                      |
  Latmax:   2.253267125E+01                      |
                  (*) HIT RETURN TO CONTINUE     |   <CR>
.............................................|...........
                  (*) HIT RETURN TO CONTINUE     |   <CR>
```



```
.............................................|...........
       (*) SET NEW MAP LIMITS [Y]                |    Y
.............................................|...........
       (*) SET MAP LIMITS                        |
           USE CURSOR CONTROLS TO MOVE LIMITS    |
           WHERE SHIFT-<ARROW> ==> 10X<ARROW>    |
           AND SPACE BAR SETS THE LIMITS.        |
           HIT RETURN TO CONTINUE                |   <CR>
```

```
.............................................|...........
      (*) SET LONGMIN    (RIGHT LIMIT)         |
          HIT RETURN TO COMMENCE               |   <CR>
.............................................|...........
        ((Limit line now appears in middle of map   ))
        ((and is moved to the right using shift-rt. ))
        ((Longmin set at this position with <space>.))
.............................................................
New longmin:  1.5470480313E+02                |
      (*) SET LONGMAX    (LEFT LIMIT)          |
          HIT RETURN TO COMMENCE               |   <CR>
.............................................|...........
        ((Longmin line remains near the right map))
        ((frame boundary while a second vertical ))
        ((line appears in the center.  This line ))
        ((is moved slightly right with one shift-))
        ((rt. entry. LONGMAX set at this position))
        ((with <space>.                          ))
.............................................................
New longmax:  1.5745584668E+02                |
      (*) SET LATMIN     (LOWER LIMIT)         |
          HIT RETURN TO COMMENCE               |   <CR>
.............................................|...........
       ((Both longitude limits remain on the right ))
       ((side of the screen while a horizontal line))
       ((appears in the middle.  This line is moved))
       ((vertically to a position just south of    ))
       ((Hawaii and set with a <space>.            ))
.............................................................
New latmin:  1.8799111875E+01                 |
      (*) SET LATMAX     (UPPER LIMIT)         |
          HIT RETURN TO COMMENCE               |   <CR>
.............................................|...........
       ((Two vertical longitude limit lines and one       ))
       ((horizontal latitude limit line are in place      ))
       ((while a second latitude limit line appears in    ))
       ((the middle of the screen. This line is moved to a))
       ((position just above Maui and set with a <space>. ))
```



```
.............................................................
New latmax:  2.1353652500E+01                 |
```

```
          HIT RETURN TO CONTINUE                  |   <CR>
.............................................................
PRESENT MAP LIMITS                                |
  Longmin:  1.547048033E+02                       |
  Longmax:  1.574558468E+02 ((NEW MAP FRAME))     |
  Latmin:   1.879911187E+01                       |
  Latmax:   2.135365250E+01                       |
 (*) READY FOR NEW MAP                            |
 (*) HIT RETURN TO COMMENCE                       |   <CR>
.............................................................
          ((Map being drawn on display screen.))
```

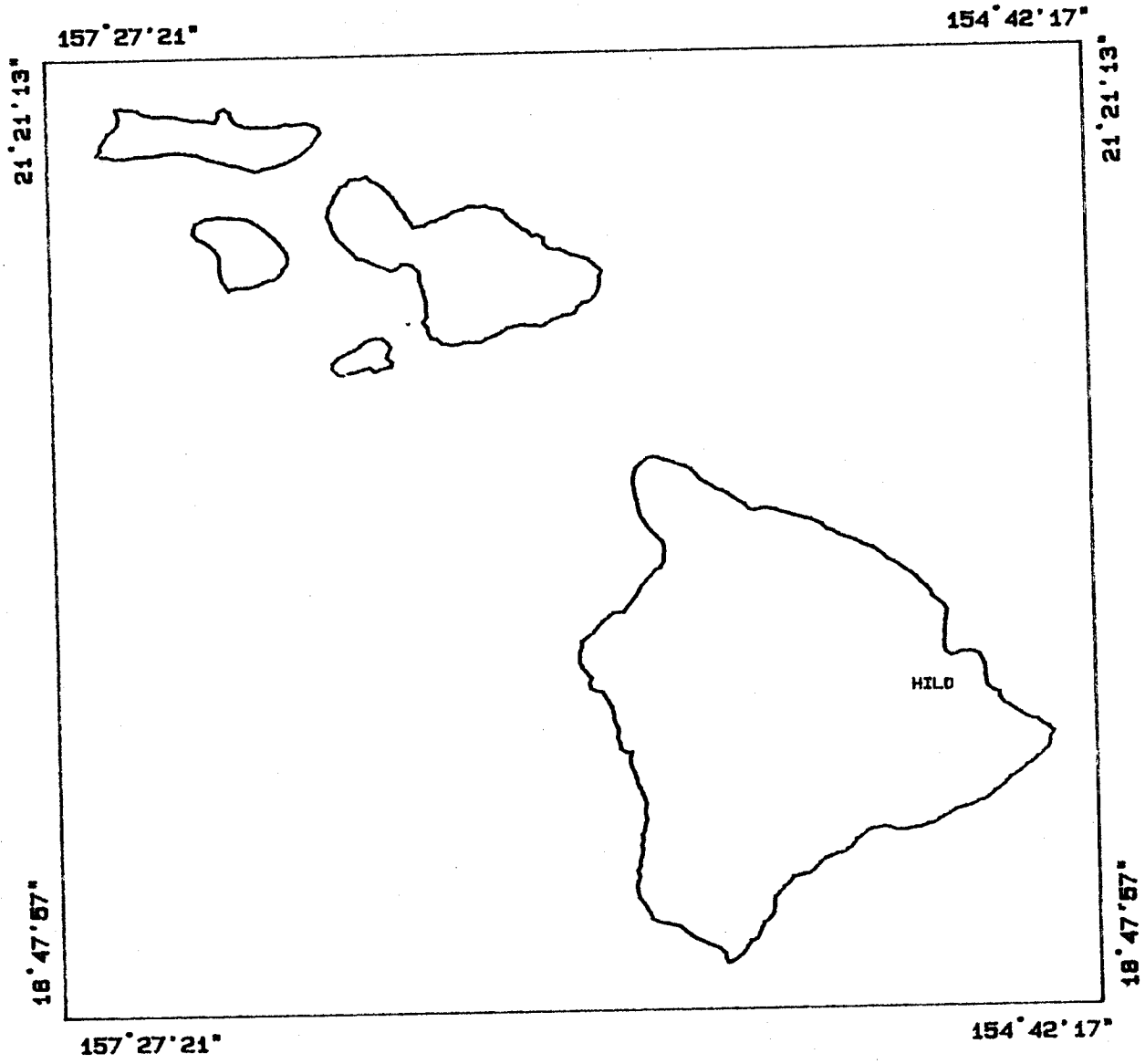
```
.............................................................
(*) CONTINUE, OR REDO MAP [N]?                    |   <CR>
.............................................................
(*) DISPLAY CONTOURS, FROM                        |
    EXTERNAL FILES, ON MAP [Y]?                   |   <CR>
.............................................................
(*) DISPLAY STATION LOCATIONS ON MAP [Y]?         |   <CR>
.............................................................
(*) DISPLAY DATA ON MAP [Y]?                      |   <CR>
.............................................................
(*) HIT RETURN TO CONTINUE                        |   <CR>
.............................................................
SELECT PLOT OPTION :                              |     1
  (1) Plot Output Map--Plotter must be            |
          physically attached and ready           |
  (2) Create Output Plot File--For transport      |
          to plotter (Pascal Driver)              |
  (3) Exit HIMP                                   |
.............................................................
Port Status = TRUE                                |
Plot factor is set at 0.9.  Proceed with this     |
value or select new value [Y]?                    |     Y
.............................................................
Enter new plot factor (0.0< <=1.0):               | 0.7 <CR>
.............................................................
              ((Plotter draws map frame.))
.............................................................
Draw Longitude Degree Tick Marks [Y]?             |   <CR>
.............................................................
```

Draw Longitude Minute Tick Marks [Y]?	<CR>
Draw Latitude Minute Tick Marks [Y]?	<CR>
Draw Longitude Second Tick Marks [Y]?	<CR>
Draw Latitude Second Tick Marks [Y]?	<CR>
SELECT NEW PEN NUMBER? (No.: 1-6. Or keep current pen.)	<CR>
SELECT NEW LINE TYPE [Y]?	<CR>
((Plotting of coastline(s).))	
Label corner coordinates [Y]?	Y
(1) LABEL UPPER LEFT CORNER (2) LABEL LOWER LEFT CORNER (3) LABEL LOWER RIGHT CORNER (4) LABEL UPPER RIGHT CORNER OR CONTINUE	1 2 3 4 <CR>
Label Plot [Y]?	Y
FOR LABEL: SELECT NEW PEN NUMBER? (No.: 1-6. Or keep current pen.)	<CR>
Number of label lines:	1 <CR>
Default character size is 0.187 w X 0.269 h (cm) Accept default, or select new size parameters [Y]?	<CR>
Center of first label line located midway between left and right paper boundaries, and 1/10 from top boundary. Accept current location, or select new location [Y]?	<CR>
Enter text for line 1:	MAUI AND HAWAII COUNTIES <CR>
WRITE ADDITIONAL LABEL [Y/N]?	Y
FOR LABEL: SELECT NEW PEN NUMBER? (No.: 1-6. Or keep current pen.)	2
Number of label lines:	1 <CR>
Default character size is 0.187 w X 0.269 h (cm)	Y

```
Accept default, or select new size parameters   |
                                        [Y]?    |
.................................................|...........
Enter new character width in cm + Return        | 0.1 <CR>
.................................................|...........
Enter new character height in cm + Return       | 0.14 <CR>
.................................................|...........
Center of first label line located midway       |     Y
between left and right paper boundaries and     |
1/10 from top boundary.  Accept current         |
location, or select new location [Y]?           |
.................................................|...........
Enter fractional distance from left to right    | 0.7 <CR>
boundaries (0.0-1.0) [Return to keep default]   |
.................................................|...........
Enter fractional distance from bottom to top    | 0.38 <CR>
boundaries (0.0-1.0) [Return to keep default]   |
.................................................|...........
    ((New lateral location: 0.7 from left to right   ))
    ((boundaries                                    ))
    ((New vertical location: 0.38 from bottom to top ))
    ((boundaries                                    ))
.................................................................
Enter text for line 1:                          | HILO <CR>
.................................................|...........
WRITE ADDITIONAL LABEL [Y]?                     |   <CR>
.................................................|...........
Make a new plot [Y] (or exit)?                  |   <CR>
.................................................|...........
FINIS                                           |
=================================================================
```

Despite the complicated appearance of the limit-setting procedure, it is really only a repetition of a three-step sequence. For each limit, the user 1) presses <CR> to initiate the sequence and display a limit line, 2) moves the line to the desired position by using a combination of cursor and shift keys, and 3) sets the line as a limit by pressing the space bar. There are four repetitions of this sequence, one for each limit that is set. In the final plot, the secondary labeling procedure produces the label "HILO" by using an alternate pen. This illustrates how the pen select option, which is available at different points during plot development, can be used to produce different colors or line thicknesses.

MAUI AND HAWAII COUNTIES



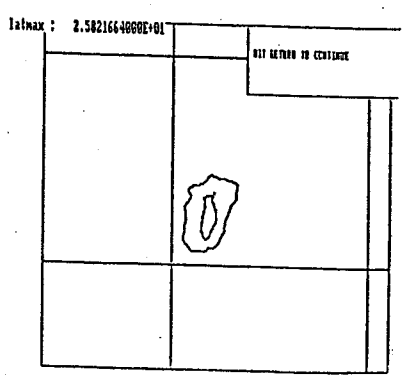
4.4 Example 3--High Resolution Island and Data Display

Example 3 illustrates the techniques producing a composite image of research data on a high resolution island map. The basic island map is produced from the HIMP library files, whereas the data are supplied from an external file specification.

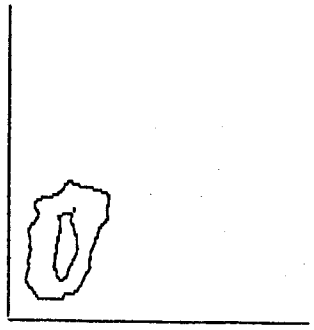
```
=============================================================
HIMP                (*) Hit Return to Commence    |   <CR>
.............................................................
Select Hawaii Mapping Option:                     |     7
  (1) Complete archipelago (Low Res)              |
  (2) Windward group (Med Res)                    |
  (3) Leeward group (Med Res)                     |
  (4) Windward group (Hi Res)                     |
  (5) Leeward group (Hi Res)                      |
  (6) Individual Windward Isle (Hi Res)           |
  (7) Individual Leeward Isle (Hi Res)            |
  (8) Freeform entry                              |
SELECT BY NUMBER                                  |
.................................................|...........
Select Island:                                    |     5
                                                  |
  (1) Nihoa                                       |
  (2) Necker                                      |
  (3) French Frigate Shoals 10f                   |
  (4) Gardner Pinnacles 20f                       |
  (5) Laysan                                      |
  (6) Lisianski                                   |
  (7) Midway NA                                   |
  (8) Kure 18f                                    |
  (9) Maro Reef 10f                               |
  (0) Pearl/Hermes Reef 10f NA                    |
                                                  |
 SELECT BY NUMBER                                 |
.................................................|...........
Enter Output Data Filename (No Extension):        |   <CR>
[Default --> "MapOut"]                            |
.................................................|...........
Output Data File:  MapOut.bin                     |
Output Map File:   MapOut.map                     |
                  (*) HIT RETURN TO CONTINUE      |   <CR>
.................................................|...........
PRESENT MAP LIMITS                                |
  Longmin:  1.716667000E+02                       |
  Longmax:  1.718000000E+02  ((MAP FRAME))        |
  Latmin:   2.571667000E+01                       |
  Latmax:   2.583333000E+01                       |
                  (*) HIT RETURN TO CONTINUE      |   <CR>
.................................................|...........
                  (*) HIT RETURN TO CONTINUE      |   <CR>
                                                  |
```

```
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
.................................................|...........
      (*) SET NEW MAP LIMITS [Y]                  |    Y
.................................................|...........
      (*) SET MAP LIMITS                          |
          USE CURSOR CONTROLS TO MOVE LIMITS      |
          WHERE SHIFT-<ARROW> ==> 10X<ARROW>      |
          AND SPACE BAR SETS THE LIMITS.          |
          HIT RETURN TO CONTINUE                  |   <CR>
.................................................|...........
      (*) SET LONGMIN    (RIGHT LIMIT)            |
          HIT RETURN TO COMMENCE                  |   <CR>
.................................................|...........
    ((Limit line now appears in middle of map and is   ))
    ((moved to the right by using the shift-rt. cursor.))
    ((Longmin set at this position with <space>.       ))
.................................................................
New longmin:  1.7167503125E+02                    |
      (*) SET LONGMAX    (LEFT LIMIT)             |
          HIT RETURN TO COMMENCE                  |   <CR>
.................................................|...........
        ((Longmin line remains near the right map ))
        ((frame boundary while a second vertical  ))
        ((line appears in the center.  This line  ))
        ((is moved slightly left using the shift- ))
        ((lt. entry.  Longmax set at this position))
        ((with <space>.                           ))
.................................................................
New longmax:  1.717500125E+02                     |
      (*) SET LATMIN     (LOWER LIMIT)            |
          HIT RETURN TO COMMENCE                  |   <CR>
.................................................|...........
      ((Both longitude limits remain, one left of  ))
      ((center and one on the right side of the    ))
      ((screen, while a horizontal line appears    ))
      ((in the middle.  This line is moved to just ))
      ((below Laysan and set with a <space> entry. ))
.................................................................
New latmin:  2.5751668000E+01                     |
      (*) SET LATMAX     (UPPER LIMIT)            |
          HIT RETURN TO COMMENCE                  |   <CR>
.................................................|...........
     ((Two vertical longitude limit lines and one  ))
     ((horizontal latitude limit line are in place ))
```

```
((while a second latitude limit line appears  ))
((in the middle of the screen.  This line is  ))
((moved upward to a position near the top of  ))
((the screen and set with a <space>.          ))
```



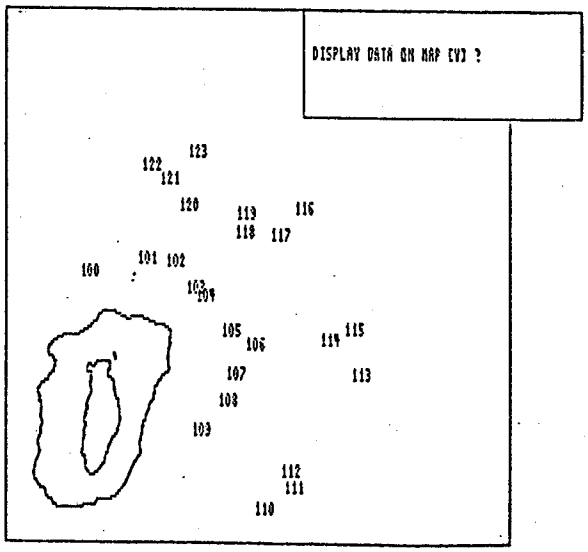

```
.............................................................
New latmax:  2.5821664000E+01                    |
      (*) HIT RETURN TO CONTINUE                  |   <CR>
.............................................................
PRESENT MAP LIMITS                                |
  Longmin:  1.7167503125E+02                      |
  Longmax:  1.7175001250E+02 ((NEW MAP FRAME))    |
  Latmin:   2.5751668000E+01                      |
  Latmax:   2.5821664000E+01                      |
 (*) Ready for new map                            |
 (*) HIT RETURN TO COMMENCE                       |   <CR>
.............................................................
          ((Map being drawn on display screen.))
```



```
.............................................................
 (*) CONTINUE OR REDO MAP [N]?                    |   <CR>
.............................................................
 (*) DISPLAY CONTOURS, FROM EXTERNAL              |
             FILES, ON MAP [Y]?                   |   <CR>
.............................................................
 (*) DISPLAY STATION LOCATIONS ON MAP [Y]?        |    Y
```

```
.............................................................
Enter Input Station Filename (with extension):  | laysdata.
                                                | dat <CR>
.............................................................
Is file in ASCII [A] or                         |    A
                 Pascal binary (.bin) [B]?      |
.............................................................
((Assume that laysdata.dat is an ASCII data file with
  records containing three entries each (latitude,
  longitude, data value), of which only the first two
  will be used for station location.  HIMP will now
  produce the corresponding binary data file laysdata.bin,
  which will enhance computational speed.  In future uses
  of HIMP with this data set, the user can respond with a
  "B" to the above query.))
.............................................................
The binary input file laysdata.bin has been     |
created and will be used for input by HIMP.     |
Input Station File: laysdata.bin                |
                                                |
Enter Output Station Filename                   |
                (no extension)  [OutStn]:       |
(Binary File of Station Locations)              |   <CR>
.............................................................
Output Station File: OutStn.bin                 |
                    (*) HIT RETURN TO CONTINUE  |   <CR>
.............................................................
SELECT DISPLAY SYMBOL TO DENOTE STATIONS        |
  (1)    +                                      |
  (2)    x                                      |
  (3)    Box                                    |
  (4)    Filled box                             |
  (5)    Diamond                                |
  (6)    Y                                      |
  (7)    *                                      |
  (8)    o                                      |
  (9)    Station number                         |
 other   User-defined (keyboard) symbol         |
                                                |
SELECT BY NUMBER OR BY KEYBOARD                 |
          CHARACTER FOR USER-DEFINED SYMBOL     |    9
.............................................................
ENTER FIRST STATION NUMBER:                     | 100 <CR>
.............................................................
   (*) Selected symbol is          N            |
       Accept this selection [Y] or             |
                    make a new selection [N].   |    Y
.............................................................
   (*) SELECTED SYMBOL IS          N            |
       WITH SCALE SIZE  1                       |
       Select new scale size (integer multiple) |
       or accept current scale (hit non-integer)|    <CR>
```

((In the statements, above "N" represents the station number, starting from 100. The user is constrained under this option to numbers increasing by 1 (i.e., 100, 101. . . 109, for a set of 10 stations. Scaling is restricted to multiple factors of the 1-scale because of limitations in the Turbo Pascal graphics library. The user may select a scale size by pressing the corresponding numeric key. The symbol will be displayed in the comm window in the selected size. Scale selection may be repeated continually until the desired size is obtained, at which time it can be accepted by pressing any non-integer key. Selecting "0" will result in one pixel being activated, to represent a zero scale size. Once the scale is selected and accepted, the base map is displayed and the station locations drawn on top.))



(*) DISPLAY DATA ON MAP [Y]?	Y
WHAT IS THE DATA TYPE?	1
(1) Station/catch	
(2) Trackline	
(3) Longline	

((The following confirmation message is "flashed" for 1 sec to verify the response of type 1.))

STATION/CATCH

Enter Input Data Filename (with extension):	laysdata. bin <CR>

```
.............................................................
Is file in ASCII [A] or                          |    B
                 Pascal binary (.bin) [B]?       |
.............................................................
Input Data File: laysdata.bin                    |
                                                 |
Enter Output Data Filename                       |
            (no extension) [OutData]             |   <CR>
(Binary File Containing Coordinates and Data)    |
.............................................................
Output Data File: OutData.bin                    |
           (*) HIT RETURN TO CONTINUE            |   <CR>
.............................................................
SELECT DISPLAY SYMBOL TO DENOTE DATA VALUES      |
  (1)    +                                       |
  (2)    x                                       |
  (3)    Box                                     |
  (4)    Filled box                              |
  (5)    Diamond                                 |
  (6)    Y                                       |
  (7)    *                                       |
  (8)    o                                       |
  (9)    Data value                              |
 other   User-defined (keyboard) symbol          |
                                                 |
SELECT BY NUMBER OR BY KEYBOARD                  |
           CHARACTER FOR USER-DEFINED SYMBOL     |     9
.............................................................
How many significant figures (0 for integer)?    |  2 <CR>
.............................................................
Symbol will be the corresponding data value.     |
(Represented by the letter N)                    |
   (*) SELECTED SYMBOL IS  N                     |
       ACCEPT THIS SELECTION [Y OR <CR>]?        |     Y
.............................................................
Select Symbol Scale Proportional to              |
                          Data Value [Y]?        |     Y
.............................................................
 ((This allows the data to be represented by symbols of  ))
 ((different size, based on value.  HIMP will now query  ))
 ((the user for information on how the data should be    ))
 ((partitioned (data bin range definition).              ))
.............................................................
Select number of data bins [1,2,3,4, or 5]:      |  2 <CR>
.............................................................
Enter minimum data value for bin 1 + Return      |  0 <CR>
.............................................................
Enter maximum data value for bin 1 + Return      |  1 <CR>
.............................................................
Select scale for bin 1 symbol ((1 sec flash))    |
  (*) SELECTED SYMBOL IS           N             |
      WITH SCALE SIZE 1                          |
```

```
      Select new scale size (integer multiple)  |
      or accept current scale (hit non-integer) |    <CR>
.......................................................
    ((This assigns a scale size of 1 to data in bin 1.))
.......................................................
Enter maximum data value for bin 2 + Return     | 2.5 <CR>
.......................................................
Select scale for bin 2 symbol  ((1 sec flash))  |
  (*) SELECTED SYMBOL IS  N                     |
      WITH SCALE SIZE 1                         |
      Select new scale size (integer multiple)  |
      or accept current scale (hit non-integer) |  2 <CR>
.......................................................
    ((This assigns a scale size of 2 to data in bin 2.))
    ((HIMP now draws the data values onto the base map))
.......................................................
                    (*) HIT RETURN TO CONTINUE  |   <CR>
                                                |
                                                |
                                                |
                                                |
                                                |
                                                |
                                                |
                                                |
                                                |
                                                |
                                                |
                                                |
                                                |
                                                |
.......................................................
SELECT PLOT OPTION:                             |
  (1) Plot Output Map--Plotter must be          |     1
                 physically attached and ready  |
  (2) Create OutPut Plot File--For transport    |
                    to plotter (Pascal Driver)  |
  (3) Exit HIMP                                 |
.......................................................
              (( MAKE SURE PLOTTER IS READY ))
.......................................................
Port Status = TRUE                              |
        ((Indicates comm port is open))         |
Plot factor is set at 0.90                      |
Proceed with this value,                        |
                 or select new value [Y]?       |     Y
.......................................................
Enter new plot factor (0.0 < <= 1.0)            |  0.7 <CR>
.......................................................
               ((Plotter draws map frame))
.......................................................
Draw Longitude Minute Tick Marks [Y]            |     Y
```

```
.......................................................
(1) Minute tick marks along bottom border        |     4
(2) Minute tick marks along top border           |
(3) Minute tick marks along top & bottom borders|
(4) Minute grid lines                            |
OR  No tick marks or grid lines                  |
.......................................................
Desired tick mark interval (No. minutes/tick):   |   1 <CR>
.......................................................
Select new line type [Y]?                        |     Y
.......................................................
Select new line type:                            |
  (1) .       .       .       .                  |     6
  (2) ____    ____    ____                       |
  (3) _____   _____   _____                      |
  (4) _______._______._______.                   |
  (5) ______ _ ______ _ ______ _                 |
  (6) ____ _ _ ____ _ _ ____ _ _                 |
  OR  ____________________                       |
.......................................................
New Line Type:  6     (( 1 sec-flash ))
          ((Plotter draws vertical grid lines))
.......................................................
Draw Latitude Minute Tick Marks [Y]?             |     Y
.......................................................
(1) Minute tick marks along left border          |     4
(2) Minute tick marks along right border         |
(3) Minute tick marks along left & right borders|
(4) Minute grid lines                            |
OR  No tick marks or grid lines                  |
.......................................................
Desired tick mark interval (No. minutes/tick):   |   1 <CR>
.......................................................
Select new line type [Y]?                        |     Y
.......................................................
Select new line type:                            |
  (1) .       .       .       .                  |     6
  (2) ____    ____    ____                       |
  (3) _____   _____   _____                      |
  (4) _______._______._______.                   |
  (5) ______ _ ______ _ ______ _                 |
  (6) ____ _ _ ____ _ _ ____ _ _                 |
  OR  ____________________                       |
.......................................................
New Line Type:  6     (( 1-sec flash ))
          ((Plotter draws lateral grid lines))
.......................................................
Draw Longitude Second Tick Marks [Y]?            |     Y
.......................................................
(1) Second tick marks along bottom border        |     3
(2) Second tick marks along top border           |
(3) Second tick marks along top & bottom borders|
```

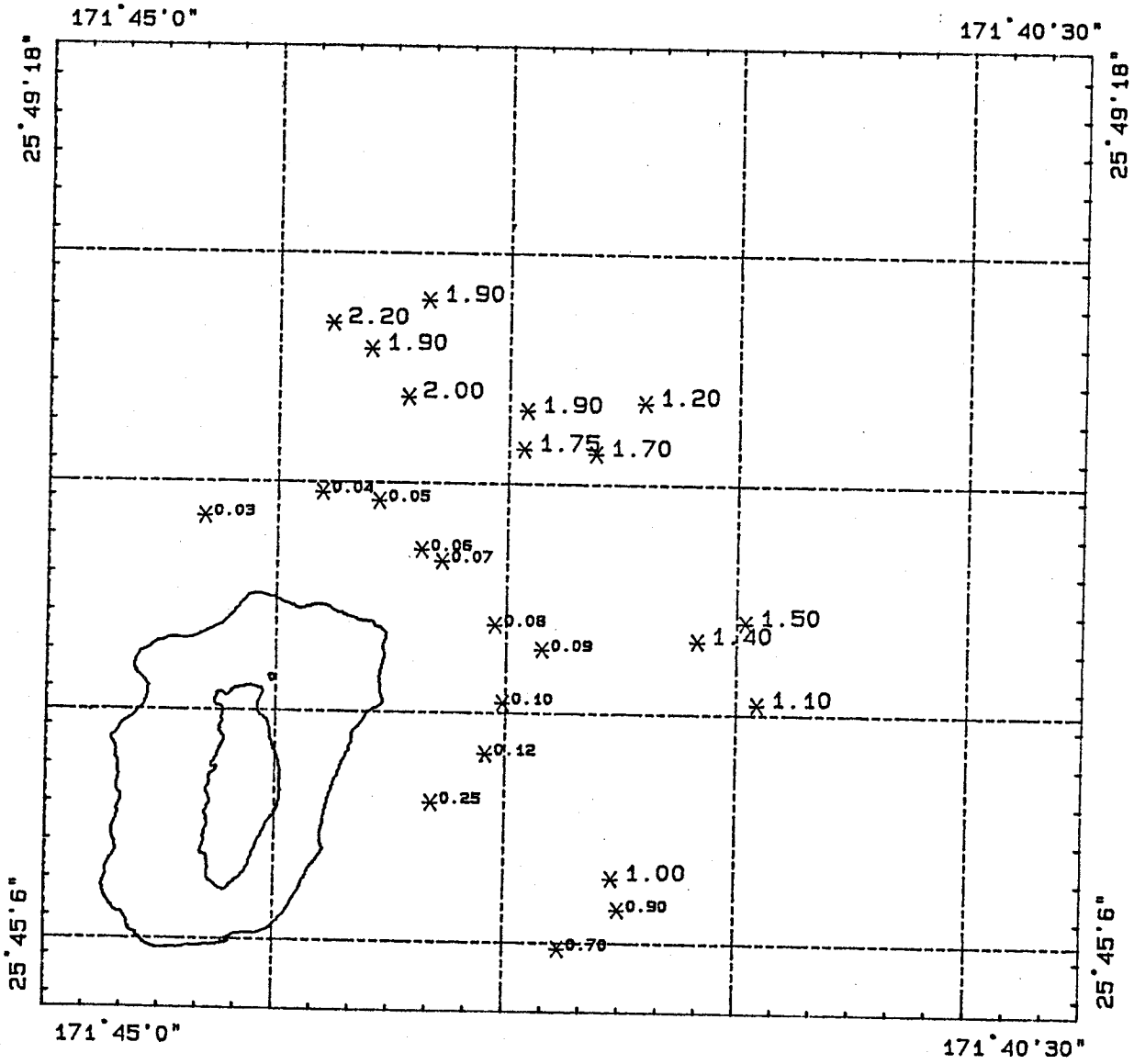
```
(4) Second grid lines                                  |
OR  No tick marks or grid lines                        |
.......................................................
Desired tick mark interval (No. seconds/tick):         |  10 <CR>
.......................................................
   ((Plotter draws tick marks at 10 sec-intervals.))
.......................................................
Draw Latitude Second Tick Marks [Y]?                   |     Y
.......................................................
(1) Second tick marks along left border                |     3
(2) Second tick marks along right border               |
(3) Second tick marks along left & right borders       |
(4) Second grid lines                                  |
OR  No tick marks or grid lines                        |
.......................................................
Desired tickmark interval (No. seconds/tick):          |  10 <CR>
.......................................................
    ((Plotter draws tick marks at 10-sec intervals.))
.......................................................
Select New Pen Number?                                 |   <CR>
   No.: 1-6.  Or keep current pen.                     |
.......................................................
Select new line type [Y]?                              |   <CR>
.......................................................
        ((Plotter draws Laysan Island coastline.))
.......................................................
Plot Station Locations [Y]?                            |     Y
.......................................................
Select New Pen Number?                                 |   <CR>
   (No.: 1-6. Or keep current pen.)                    |
.......................................................
(*) SELECTED SYMBOL IS          N                      |
    ACCEPT THIS PLOT SYMBOL [Y or CR]?                 |     N
.......................................................
  (( The "N" response allows the user to change the plot ))
  (( symbol from that used during screen development.    ))
.......................................................
SELECT PLOT SYMBOL TO DENOTE STATIONS:                 |
  (1)    +                                             |
  (2)    x                                             |
  (3)    Box                                           |
  (4)    Filled box                                    |
  (5)    Diamond                                       |
  (6)    Y                                             |
  (7)    *                                             |
  (8)    o                                             |
  (9)    Station number                                |
 other   User-defined (keyboard) symbol                |
                                                       |
SELECT BY NUMBER OR BY KEYBOARD                        |
           CHARACTER FOR USER-DEFINED SYMBOL           |     1
```

```
.............................................................
(*) SELECTED SYMBOL IS          +                  |
    Accept this selection [Y] or                   |
                    make a new selection [N]?      |    <CR>
.............................................................

(*) PLOT SYMBOL SIZE IS 0.200 X 0.200 CM.          |
    KEEP THIS SIZE [Y] OR SELECT NEW SIZE [N]?     |     Y
.............................................................
(*) READY TO PLOT STATION LOCATIONS.               |
    HIT RETURN TO COMMENCE                         |    <CR>
.............................................................
      ((Plotter now draws station locations onto map.))
.............................................................
       Plot data values [Y]?                       |     Y
.............................................................
Select New Pen Number?                             |   <CR>
   (No.: 1-6.  Or keep current pen.)               |
.............................................................
(*) SELECTED SYMBOL IS          N                  |
    ACCEPT THIS PLOT SYMBOL [Y or CR]?             |   <CR>
.............................................................
DATA IS CURRENTLY PARTITIONED INTO 2 BINS          |
KEEP CURRENT PARTITIONING OR SELECT NEW            |
PARTITIONS [Y]?                                    |   <CR>
.............................................................
  (( A <CR> is used to keep the two-partition structure.  ))
  (( If new partitioning is desired, a "Y" should be      ))
  (( entered.  Any other response is equivalent to <CR>.  ))
.............................................................
CURRENT DEFAULT SYMBOL SIZE IS 0.400 W X 0.576 H|
+ 0.04W X 0.058H CM INCREMENT INCREASE.            |
                                                   |
(Note:  The 1.44 h/w ratio produces a plot         |
        aspect ratio of 1)                         |
KEEP DEFAULT OR SELECT NEW SIZE PARAMETERS [Y]?    |     Y
.............................................................
      (( The "Y" response indicates that a change in ))
      (( symbol size is desired.                     ))
.............................................................
ENTER NEW SYMBOL WIDTH (CM) + RETURN              | 0.2 <CR>
.............................................................
ENTER NEW SYMBOL HEIGHT (CM) + RETURN             | 0.28 <CR>
.............................................................
ASPECT RATIO FOR SYMBOLS (H/W) IS 1.40000000E+00|
INCREMENTAL INCREASE WILL BE dW IN WIDTH           |
AND dH = (H/W) * dW IN HEIGHT.                     |
ENTER INCREMENT dW (CM) + RETURN                   | 0.05 <CR>
.............................................................
Is a one-character offset desired [Y]?             |     Y
```

```
.............................................................
((This allows the data values to be drawn next to the     ))
((station location and avoids confusing overwrites.       ))
((The plotter now draws data values next to each station  ))
((location symbol.  The characters are drawn in two sizes,))
((depending on which bin the corresponding values fall in.))
.............................................................
Label corner coordinates [Y]?                   |     Y
.............................................................
  (1) Label upper left corner                   |     1
  (2) Label lower left corner                   |     2
  (3) Label lower right corner                  |     3
  (4) Label upper right corner                  |     4
  OR  Continue                                  |    <CR>
.............................................................
 (( Plotter draws latitude and longitude at each corner.))
.............................................................
Label Plot?                                     |     Y
.............................................................
Select New Pen Number?                          |   <CR>
   (No.: 1-6  Or keep current pen.)             |
.............................................................
Number of line in label?                        |   1 <CR>
.............................................................
Default character size is 0.187 w X 0.269 h (cm)|
Accept default or                               |
              select new size parameters [Y]?   |   <CR>
.............................................................
Center of first label line is midway between the|
left and right boundaries and 1/10 down from the|
top boundary.                                   |
Accept current location, or select new location |
                                       [Y]?     |   <CR>
.............................................................
Enter text for line 1:                          | LAYSAN
                                                | CATCH DATA
                                                |   <CR>
.............................................................
WRITE ADDITIONAL LABELS (Y/N)?                  |   <CR>
.............................................................
Make a new plot [Y] (or exit)?                  |   <CR>
.............................................................
FINIS                                           |
=============================================================
```

The preceding example illustrates how the user can control the various features of a map image during the plotting of development. Some features used during the screen-oriented phase can be changed or modified while plotting. By electing to make a new map, the user may produce additional plots and experiment with various modifications.

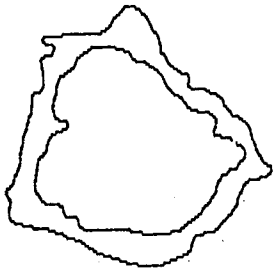
LAYSAN CATCH DATA



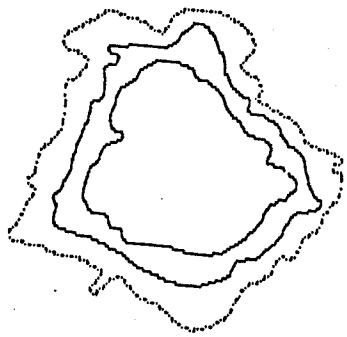
4.5 Example 4--Freeform Map Development

Example 4 illustrates how a basic map image is derived from a non-library external file, with digitized data representing two depth contours for Hancock Seamount. An additional depth contour is added to the map by using another external digitized data file. Both data files are located in the \ex subdirectory. In general, HIMP recognizes all standard DOS paths, such as ex\my.dat or A:\mydir\myfile.dat.

```
=============================================================
HIMP                (*) Hit Return to Commence     |   <CR>
.............................................................
Select Hawaii Mapping Option:                      |     8
  (1) Complete archipelago (Low Res)               |
  (2) Windward group (Med Res)                     |
  (3) Leeward group (Med Res)                      |
  (4) Windward group (Hi Res)                      |
  (5) Leeward group (Hi Res)                       |
  (6) Individual Windward Isle (Hi Res)            |
  (7) Individual Leeward Isle (Hi Res)             |
  (8) Freeform entry                               |
SELECT BY NUMBER                                   |
.................................................|...........
Enter Input Filename (with extension):             | ex\hanc0.
                                                   | dat <CR>
.................................................|...........
Is input file in ASCII [A] or                      |
                   Pascal binary (.bin) [B]?       |     A
.................................................|...........
The binary input file hanc0.bin has been created|
and will be used for input by HIMP.                |
Input Data File:  ex\hanc0.bin                     |
Input Map File:   ex\hanc0.map                     |
  (*) Does map file already exist (Y/N)?           |     N
.................................................|...........
Enter Output Data Filename (No Extension):         |   <CR>
[Default --> "MapOut"]                             |
.................................................|...........
Output Data File:  MapOut.bin                      |
Output Map File:   MapOut.map                      |
                  (*) HIT RETURN TO CONTINUE       |   <CR>
.................................................|...........
PRESENT MAP LIMITS                                 |
  Longmin:  1.8090830000E+02                       |
  Longmax:  1.8096670000E+02  ((MAP FRAME))        |
  Latmin:   2.9766670000E+01                       |
  Latmax:   2.9833330000E+01                       |
                  (*) HIT RETURN TO CONTINUE       |   <CR>
.................................................|...........
HIMP now draws the contour lines and produces the base
map image.  (If the map file already exists, the map image
is loaded onto the screen and not drawn.)
```



```
.............................................................
     (*) Modify Map [Y]?                          |   <CR>
.............................................................
     (*) CONTINUE OR REMODIFY [N]?                |   <CR>
.............................................................
     (*) DISPLAY CONTOURS,                        |
           FROM EXTERNAL FILES, ON MAP [Y]?       |     Y
.............................................................
Current line style is 4                           | ex\hanc1.
Enter Input Filename (with extension):            | dat <CR>
.............................................................
Is input file in ASCII [A] or                     |
                    Pascal binary (.bin) [B]?     |     A
.............................................................
The binary input file hanc1.bin has been created|
and will be used for input by HIMP                |
Input Data File:  ex\hanc1.bin                    |
Select new line style [Y]?                        |     Y
.............................................|...........
SELECT LINE STYLE:                                |
  (0) ***************                             |
  (1) *   *   *   *                               |
  (2) *****   *****                               |
  (3) ***  *  ***  *                              |
  (4) *** *** *** ***                             |  3 <CR>
.............................................|...........
       (( HIMP displays the base map image and  ))
       (( draws the new depth contour on top.   ))
.............................................................
            (*) ADD ADDITIONAL CONTOURS [Y]?      |   <CR>
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
                                                  |
```



```
                                                  |
.................................................|...........
    (*) DISPLAY STATION LOCATIONS ON MAP [Y]?    |   <CR>
.................................................|...........
                (*) DISPLAY DATA ON MAP [Y]?     |   <CR>
.................................................|...........
                   (*) HIT RETURN TO CONTINUE    |   <CR>
.................................................................
SELECT PLOT OPTION:                               |
  (1) Plot Output Map--Plotter must be            |     1
                 physically attached and ready    |
  (2) Create OutPut Plot File--For transport      |
                     to plotter (Pascal Driver)   |
  (3) Exit HIMP                                   |
.................................................................
             (( MAKE SURE PLOTTER IS READY ))
.................................................................
Port Status = TRUE                                |
        ((Indicates comm port is open))           |
Plot factor is set at 0.90                        |
Proceed with this value,                          |
                 or select new value [Y]?         |     Y
.................................................................
Enter new plot factor (0.0 < <= 1.0)              |  0.7 <CR>
.................................................................
               ((Plotter draws map frame))
.................................................................
Draw Longitude Minute Tick Marks [Y]?             |     Y
.................................................................
(1) Minute tick marks along bottom border         |     3
(2) Minute tick marks along top border            |
(3) Minute tick marks along top & bottom borders|
(4) Minute grid lines                             |
OR  No tick marks or grid lines                   |
.................................................................
Desired tick mark interval (No. minutes/tick):    |   1 <CR>
.................................................................
     ((Plotter draws minute tick marks along top and ))
     ((bottom frame boundaries at 1-min intervals.   ))
.................................................................
Draw Latitude Minute Tick Marks [Y]?              |     Y
.................................................................
(1) Minute tick marks along left border           |     3
(2) Minute tick marks along right border          |
(3) Minute tick marks along left & right borders|
(4) Minute grid lines                             |
OR  No tick marks or grid lines                   |
.................................................................
Desired tick mark interval (No. minutes/tick):    |   1 <CR>
.................................................................
     ((Plotter draws minute tick marks along left and ))
     ((right frame boundaries at 1-min intervals.     ))
```

```
.............................................................
Draw Longitude Second Tick Marks [Y]?                  |   <CR>
.............................................................
Draw Latitude Second Tick Marks [Y]?                   |   <CR>
.............................................................
SELECT NEW PEN NUMBER?                                 |   <CR>
  (No.: 1-6.  Or keep current pen.)                    |
.............................................................
SELECT NEW LINE TYPE [Y]?                              |     Y
.............................................................
SELECT NEW LINE TYPE:                                  |
  (1) .       .       .       .                        |     4
  (2) ____    ____    ____                             |
  (3) _____   _____   _____                            |
  (4) _______._______._______.                         |
  (5) ______ _ ______ _ ______ _                       |
  (6) ____ _ _ ____ _ _ ____ _ _                       |
  OR  ____________________                             |
.............................................................
New Line Type:  4
        ((Plotter draws original depth contours.))
.............................................................
SELECT NEW LINE TYPE FOR CONTOUR [Y]?                  |     Y
.............................................................
Select new line type:                                  |
  (1) .       .       .       .                        |     5
  (2) ____    ____    ____                             |
  (3) _____   _____   _____                            |
  (4) _______._______._______.                         |
  (5) ______ _ ______ _ ______ _                       |
  (6) ____ _ _ ____ _ _ ____ _ _                       |
  OR  ____________________                             |
.............................................................
New Line Type:  5
            ((Plotter draws new depth contour.))
.............................................................
Label corner coordinates [Y]?                          |     Y
.............................................................
  (1) Label upper left corner                          |     1
  (2) Label lower left corner                          |     2
  (3) Label lower right corner                         |     3
  (4) Label upper right corner                         |     4
  OR  Continue                                         |    <CR>
.............................................................
    ((Plotter draws latitude and longitude at each corner.))
.............................................................
Label Plot?                                            |     Y
.............................................................
SELECT NEW PEN NUMBER?                                 |   <CR>
  (No.: 1-6.  Or keep current pen.)                    |
.............................................................
Number of lines in label?                              |   1 <CR>
```

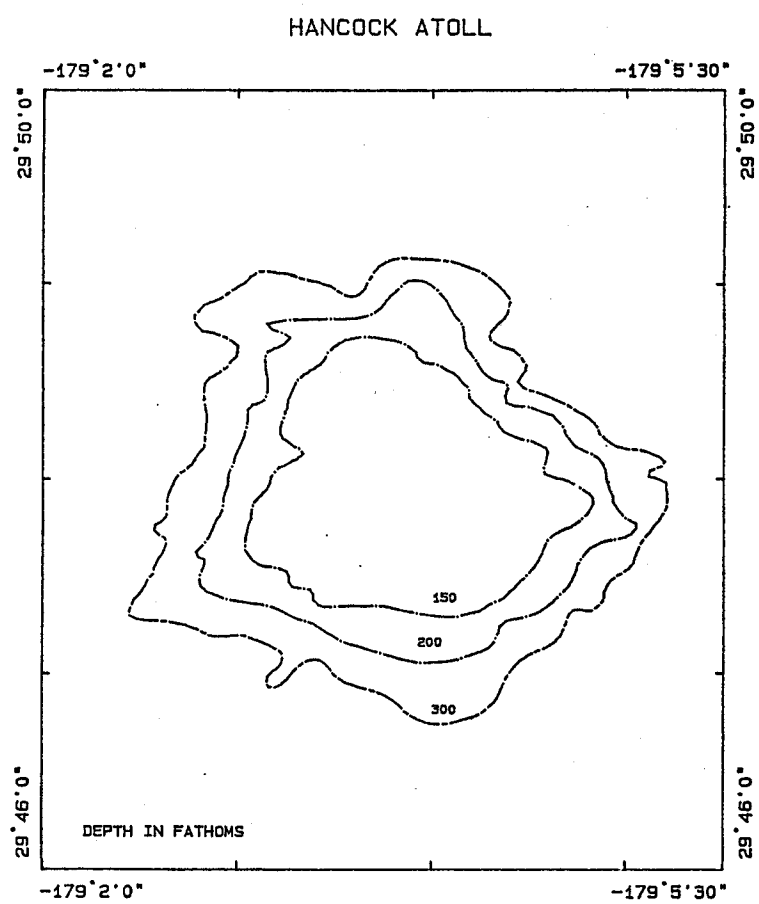
```
.......................................................
Default character size is 0.187 w X 0.269 h (cm)|
Accept default or                               |
             select new size parameters [Y]?    |   <CR>
.......................................................
Center of first label line is midway between the|
left and right boundaries and 1/10 down from the|
top boundary.                                   |
Accept current location, or select new location |
                                     [Y]?       |   <CR>
.......................................................
Enter text for line 1:                          | HANCOCK
                                                | SEAMOUNT
                                                |  <CR>
.......................................................
WRITE ADDITIONAL LABEL (Y/N)?                   |    Y
.......................................................
SELECT NEW PEN NUMBER?                          |    2
  (No.: 1-6.  Or keep current pen.)             |
.......................................................
((A new pen with a finer tip is selected for small labels.))
.......................................................
Number of lines in label?                       |  1 <CR>
.......................................................
Default character size is 0.187 w X 0.269 h (cm)|
Accept default or                               |
             select new size parameters [Y]?    |    Y
.......................................................
Enter new character width in cm + Return        | 0.1 <CR>
.................................................|..........
Enter new character height in cm + Return       | 0.14 <CR>
.................................................|..........
Center of first label line located midway        |    Y
between left and right paper boundaries and     |
1/10 from top boundary.  Accept current         |
location, or select new location [Y]?           |
.................................................|..........
Enter fractional distance from left to right    | 0.54 <CR>
boundaries (0.0-1.0) [Return to keep default]   |
.................................................|..........
Enter fractional distance from bottom to top    | 0.39 <CR>
boundaries (0.0-1.0) [Return to keep default]   |
.................................................|..........
    ((New lateral location:  0.54 from left to right  ))
    ((boundaries                                      ))
    ((New vertical location:  0.39 from bottom to top ))
    ((boundaries                                      ))
.......................................................
Enter text for line 1:                          | 150 <CR>
.................................................|..........
WRITE ADDITIONAL LABEL [Y/N]?                   |    Y
.......................................................
```

```
SELECT NEW PEN NUMBER?                                  |   <CR>
   (No.: 1-6.  Or keep current pen.)                    |
.............................................................
Number of lines in label?                               |   1 <CR>
.............................................................
Default character size is 0.187 w X 0.269 h (cm)|
Accept default or                                       |
              select new size parameters [Y]?           |     Y
.............................................................
Enter new character width in cm + Return                | 0.1 <CR>
.............................................................|...........
Enter new character height in cm + Return               | 0.14 <CR>
.............................................................|...........
Center of first label line located midway                |     Y
between left and right paper boundaries and             |
1/10 from top boundary.  Accept current                 |
location, or select new location [Y]?                   |
.............................................................|...........
Enter fractional distance from left to right            | 0.53 <CR>
boundaries (0.0-1.0) [Return to keep default]           |
.............................................................|...........
Enter fractional distance from bottom to top            | 0.35 <CR>
boundaries (0.0-1.0) [Return to keep default]           |
.............................................................|...........
    ((New lateral location:  0.53 from left to right  ))
    ((boundaries                                      ))
    ((New vertical location:  0.35 from bottom to top ))
    ((boundaries                                      ))
.............................................................
Enter text for line 1:                                  |  200 <CR>
.............................................................|...........
WRITE ADDITIONAL LABEL [Y/N]?                           |    Y
.............................................................
SELECT NEW PEN NUMBER?                                  |   <CR>
   (No.: 1-6.  Or keep current pen.)                    |
.............................................................
Number of lines in label?                               |   1 <CR>
.............................................................
Default character size is 0.187 w X 0.269 h (cm)|
Accept default or                                       |
              select new size parameters [Y]?           |     Y
.............................................................
Enter new character width in cm + Return                | 0.1 <CR>
.............................................................|...........
Enter new character height in cm + Return               | 0.14 <CR>
.............................................................|...........
Center of first label line located midway                |     Y
between left and right paper boundaries and             |
1/10 from top boundary.  Accept current                 |
location, or select new location [Y]?                   |
.............................................................|...........
Enter fractional distance from left to right            | 0.54 <CR>
boundaries (0.0-1.0) [Return to keep default]           |
```

```
..............................................|...........
Enter fractional distance from bottom to top  | 0.29 <CR>
boundaries (0.0-1.0) [Return to keep default] |
..............................................|...........
    ((New lateral location:  0.54 from left to right  ))
    ((boundaries                                      ))
    ((New vertical location:  0.29 from bottom to top ))
    ((boundaries                                      ))
.......................................................
Enter text for line 1:                        |  300 <CR>
..............................................|...........
WRITE ADDITIONAL LABEL [Y/N]?                 |    Y
.......................................................
SELECT NEW PEN NUMBER?                        |   <CR>
(No.: 1-6.  Or keep current pen.)             |
.......................................................
Number of lines in label?                     |   1 <CR>
.......................................................
Default character size is 0.187 w X 0.269 h (cm)|
Accept default or                             |
             select new size parameters [Y]?  |     Y
.......................................................
Enter new character width in cm + Return      | 0.12 <CR>
..............................................|...........
Enter new character height in cm + Return     | 0.17 <CR>
..............................................|...........
Center of first label line located midway      |     Y
between left and right paper boundaries and   |
1/10 from top boundary.  Accept current       |
location, or select new location [Y]?         |
..............................................|...........
Enter fractional distance from left to right  | 0.35 <CR>
boundaries (0.0-1.0) [Return to keep default] |
..............................................|...........
Enter fractional distance from bottom to top  | 0.18 <CR>
boundaries (0.0-1.0) [Return to keep default] |
..............................................|...........
    ((New lateral location:  0.35 from left to right  ))
    ((boundaries                                      ))
    ((New vertical location:  0.18 from bottom to top ))
    ((boundaries                                      ))
.......................................................
Enter text for line 1:                        | DEPTH IN
                                              |FATHOMS<CR>
..............................................|...........
WRITE ADDITIONAL LABEL [Y/N]?                 |   <CR>
.......................................................
Make new plot [Y] (or exit)?                  |   <CR>
.......................................................
FINIS                                         |
=======================================================
```

The preceding example illustrates, to a large degree, the control that the user has over the map image during the plotting stage of production. In addition to enhancing the image with several plotting features, some screen-developed items such as data symbols, line types, and data partitioning can be changed or altered during this phase. Control over pen selection allows the use of different colors and thicknesses of pens to add detail and clarity to the plot. Because the HIMP produces a plot in a stepwise sequence, the user has extensive control over plot development. It is advisable to take a few basic notes as a draft is developed and then use the "make new plot" option to produce an improved draft or final plot. This is advantageous when placing secondary labels on the plot.

During screen development, the user could have specified an output file by name (e.g., HANCOO.bin) and added the new depth contour to create a current version of the atoll map. A subsequent run of HIMP could specify this new file as input. The resulting map image would show all three depth contours in the same line type. As new depth contours become available, they can be added in the same manner to produce new versions of the atoll map. Any map files produced by HIMP may be installed into the map library either by adding the "HMAP\" prefix to the output file specifications during the HIMP session or by copying the files into the HMAP directory while in DOS.



4.6 Example 5--Trackline Map Development

Example 5 uses a trackline data file to display a track recording off the west coast of Oahu. Trackline records consist of latitude and longitude values followed by a data entry. A data entry can represent some measured value or a numeric indicator of an event, such as an hourly position fix. The data file uses the third entry of each record to indicate when an hourly position fix was made. The user can use the partitioning option to constrain the symbol-displaying routine so that only certain numeric values result in a symbol appearing along the track. All third entries will either be "0" or "1"; the latter indicates when a fix was taken. With the trackline option, HIMP considers all data to represent a continuous line. (This is in contrast to the catch-station option, where the data represent individual points and the longline option represents a series of individual straight lines.)

```
=============================================================
HIMP                (*) Hit Return to Commence   |   <CR>
.............................................|...........
Select Hawaii Mapping Option:                    |     6
  (1) Complete archipelago (Low Res)             |
  (2) Windward group (Med Res)                   |
  (3) Leeward group (Med Res)                    |
  (4) Windward group (Hi Res)                    |
  (5) Leeward group (Hi Res)                     |
  (6) Individual Windward Isle (Hi Res)          |
  (7) Individual Leeward Isle (Hi Res)           |
  (8) Freeform entry                             |
SELECT BY NUMBER                                 |
.............................................|...........
Select Island:                                   |     6
  (1) Hawaii                                     |
  (2) Maui                                       |
  (3) Kahoolawe                                  |
  (4) Lanai                                      |
  (5) Molokai                                    |
  (6) Oahu                                       |
  (7) Kauai                                      |
  (8) Niihau                                     |
SELECT BY NUMBER                                 |
.............................................|...........
Enter Output Data Filename (No Extension):       |   <CR>
[Default --> "MapOut"]                           |
.............................................|...........
Output Data File:  MapOut.bin                    |
Output Map File:   MapOut.map                    |
                  (*) HIT RETURN TO CONTINUE     |   <CR>
.............................................|...........
PRESENT MAP LIMITS                               |
  Longmin:  1.5752000000E+02                     |
  Longmax:  1.5838300000E+02  ((MAP FRAME))      |
```

```
  Latmin:   2.1238812500E+01                        |
  Latmax:   2.1778187500E+01                        |
                 (*) HIT RETURN TO CONTINUE         |   <CR>
.............................................................
     (( The high resolution map of Oahu is displayed. ))
.............................................................
                 (*) HIT RETURN TO CONTINUE         |   <CR>
.............................................................
       (*) SET NEW MAP LIMITS [Y]                   |    Y
.............................................................
       (*) SET MAP LIMITS                           |
           USE CURSOR CONTROLS TO MOVE LIMITS       |
           WHERE SHIFT-<ARROW> ==> 10X<ARROW>       |
           AND SPACE BAR SETS THE LIMITS.           |
           (HIT RETURN TO CONTINUE)                 |   <CR>
.............................................................
       (*) SET LONGMIN    (RIGHT LIMIT)             |
           HIT RETURN TO COMMENCE                   |   <CR>
.............................................................
  (( Limit line now appears in middle of map and then is))
  (( moved to the left using the shift-lt. cursor.      ))
  (( Longmin is set at this position with a <space>.    ))
.............................................................
New longmin:  1.5822118750E+02                      |
       (*) SET LONGMAX    (LEFT LIMIT)              |
           HIT RETURN TO COMMENCE                   |   <CR>
.............................................................
  (( Longmin line remains near the left side of Oahu,   ))
  (( while a second vertical line appears in the center.))
  (( This line is moved left to the left frame boundary ))
  (( using the shift-lt. entry.  Longmax set at this    ))
  (( position with a <space>.                           ))
.............................................................
New longmax:  1.5838300000E+02                      |
       (*) SET LATMIN     (LOWER LIMIT)             |
           HIT RETURN TO COMMENCE                   |   <CR>
.............................................................
  (( Both longitude limits remain, one on the left side ))
  (( of the island and one near the left side of the    ))
  (( image frame.  A horizontal line appears in the     ))
  (( middle.  This line is moved to a position roughly  ))
  (( parallel to the location of Makaha and set with a  ))
  (( <space>.                                           ))
.............................................................
New latmin:  2.1400625000E+01                       |
       (*) SET LATMAX     (UPPER LIMIT)             |
           HIT RETURN TO COMMENCE                   |   <CR>
.............................................................
  (( Two vertical longitude limit lines and one         ))
  (( horizontal latitude limit line are in place, while ))
  (( a second latitude limit line appears in the middle ))
  (( of the screen.  This line is moved upward to a     ))
```

```
  (( position just above and parallel to Kaena Point    ))
  (( and set with a <space>.                            ))
.............................................................
New latmax:  2.1616375000E+01                      |
       (*) HIT RETURN TO CONTINUE                  |   <CR>
.............................................................
PRESENT MAP LIMITS                                 |
  Longmin:  1.5822118750E+02                       |
  Longmax:  1.5838300000E+02 ((NEW MAP FRAME))     |
  Latmin:   2.1400625000E+01                       |
  Latmax:   2.1616375000E+01                       |
 (*) READY FOR NEW MAP                             |
 (*) HIT RETURN TO COMMENCE                        |   <CR>
.............................................................
  (( A map image of a section of west Oahu is drawn on  ))
  (( the display screen.                                ))
.............................................................
 (*) CONTINUE OR REDO MAP [N]?                     |   <CR>
.............................................................
 (*) DISPLAY CONTOURS, FROM EXTERNAL               |
             FILES, ON MAP [Y]?                    |   <CR>
.............................................................
 (*) DISPLAY STATION LOCATIONS ON MAP [Y]?         |   <CR>
.............................................................
 (*) DISPLAY DATA ON MAP [Y]?                      |     Y
.............................................................
 WHAT IS THE DATA TYPE?                            |     2
      (1)  Station/catch                           |
      (2)  Trackline                               |
      (3)  Longline                                |
.............................................................
 TRACKLINE DATA    (( 1-sec flash ))
.............................................................
Enter Input Data Filename (with extension):        |ex\rb1trak.
  (( Assume this binary file already exists  )) | bin <CR>
  (( in the ex-directory.                    )) |
.............................................................
Is file in ASCII [A] or                            |    B
                 Pascal binary (.bin) [B]?         |
.............................................................
Input Data File: ex\rb1trak.bin                    |
                                                   |
Enter Output Data Filename                         |
            (no extension) [OutData]               |   <CR>
(Binary File Containing Coordinates and Data)      |
.............................................................
Output Data File: OutData.bin                      |
           (*) HIT RETURN TO CONTINUE              |   <CR>
```

```
.............................................................
SELECT DISPLAY SYMBOL TO DENOTE DATA VALUES      |
  (1)    +                                       |
  (2)    x                                       |
  (3)    Box                                     |
  (4)    Filled box                              |
  (5)    Diamond                                 |
  (6)    Y                                       |
  (7)    *                                       |
  (8)    o                                       |
  (9)    Data value                              |
 other   User-defined (keyboard) symbol          |
                                                 |
SELECT BY NUMBER OR BY KEYBOARD                  |
          CHARACTER FOR USER-DEFINED SYMBOL      |     1
.............................................................
   (*) SELECTED SYMBOL IS       +                |
       Accept this selection [Y or <CR>]?        |     Y
.............................................................
  (( The file rb1trak.bin contains a series of records  ))
  (( giving latitude, longitude, and a integer value    ))
  (( that is used to denote hourly fixes.  In this case,))
  (( "1" is selected to indicate an hourly fix and      ))
  (( "0" otherwise.  To facilitate symbol marking for   ))
  (( these fixes, a partitioning of the data is made    ))
  (( so that a "0" causes a zero-dimensioned symbol     ))
  (( (no symbol) to be displayed, while a "1" results   ))
  (( in a "+" of scale size 2 being displayed.  The     ))
  (( corresponding data bins will be [-0.1, 0.1] and    ))
  (( [0.1, 2.0].                                        ))
.............................................................
Select Symbol Scale Proportional to              |
                          Data Value [Y]?        |     Y
.............................................................
Select number of data bins [1,2,3,4, or 5]:      |  2 <CR>
.............................................................
Enter minimum data value for bin 1 + Return      | -0.1<CR>
.............................................................
Enter maximum data value for bin 1 + Return      |  0.1<CR>
.............................................................
Select scale for bin 1 symbol  ((1 sec flash))   |
  (*) SELECTED SYMBOL IS          +              |
      WITH SCALE SIZE 1                          |
      Select new scale size (integer multiple)   |
      or accept current scale (hit non-integer)  |  0 <CR>
.............................................................
  (( This assigns a scale size of 0 to data in bin 1    ))
  (( [-0.1, 0.1].  When the "0" is pressed the "+" will ))
  (( be replaced by "." to indicate size 0.  The <CR>   ))
  (( will cause HIMP to accept this size for bin 1.     ))
```

```
.............................................................
Enter maximum data value for bin 2 + Return      |  2 <CR>
.............................................................
Select scale for bin 2 symbol  ((1 sec-flash))   |
  (*) SELECTED SYMBOL IS          +              |
      WITH SCALE SIZE 1                          |
      Select new scale size (integer multiple),  |
      or accept current scale (hit non-integer)  |  2 <CR>
.............................................................
  (( This assigns a scale size of 2 to data in bin 2,   ))
  (( [ 0.1, 2.0]. When the "2" is pressed, the "+" will))
  (( increase in size by a factor of two. The <CR> will))
  (( cause HIMP to accept this size for bin 2.          ))
  (( HIMP now draws the trackline and then fixes the    ))
  (( symbols onto the base map.                         ))
.............................................................
                   (*) HIT RETURN TO CONTINUE    |   <CR>
                                                 |
                                                 |
                                                 |
                                                 |
                                                 |
                                                 |
                                                 |
                                                 |
                                                 |
                                                 |
                                                 |
.............................................................
SELECT PLOT OPTION:                              |
  (1) Plot Output Map--Plotter must be           |     1
                 physically attached and ready   |
  (2) Create OutPut Plot File--For transport     |
                   to plotter (Pascal Driver)    |
  (3) Exit HIMP                                  |
.............................................................
             (( MAKE SURE PLOTTER IS READY ))
.............................................................
Port Status = TRUE                               |
        ((Indicates comm port is open))          |
Plot factor is set at 0.90                       |
Proceed with this value,                         |
                 or select new value [Y]?        |     Y
.............................................................
Enter new plot factor (0.0 < <= 1.0)             |  0.7 <CR>
.............................................................
               ((Plotter draws map frame))
.............................................................
Draw Longitude Minute Tick Marks [Y]?            |     Y
```

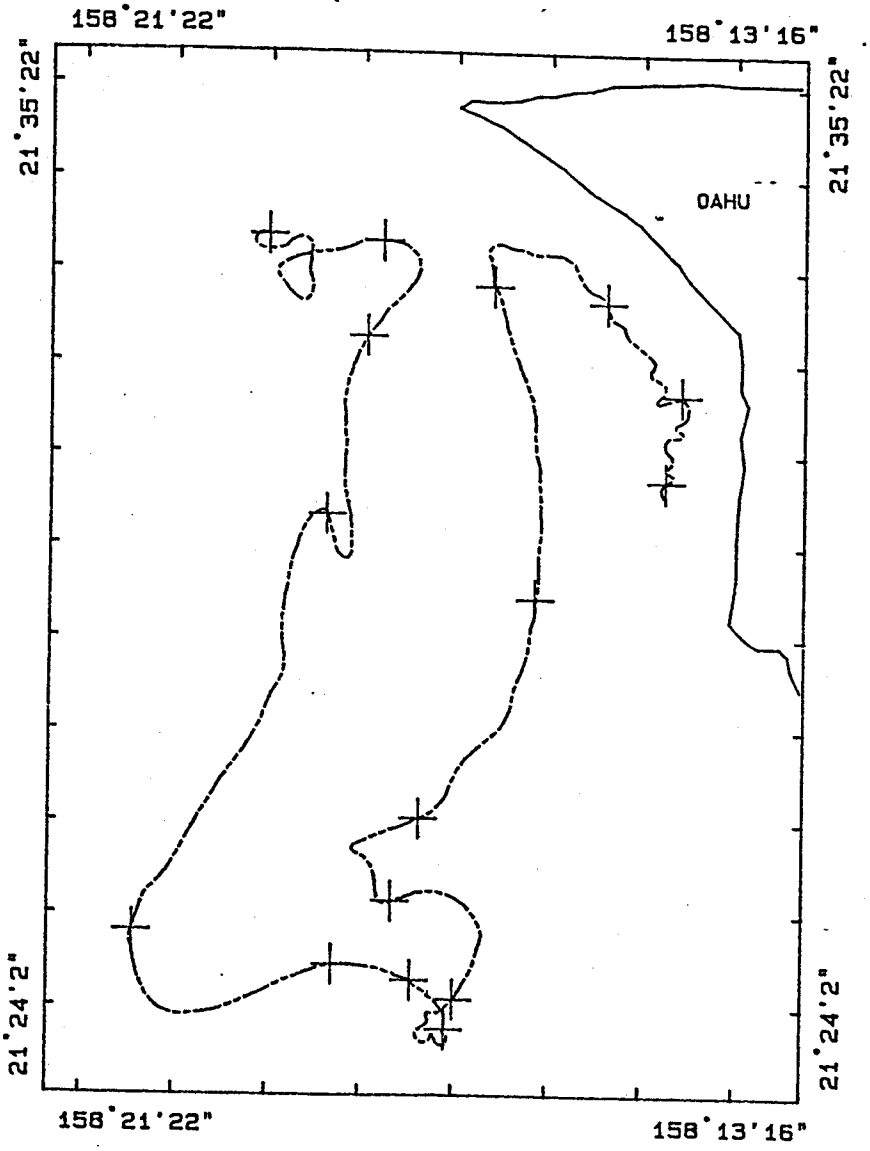
```
.............................................................
(1) Minute tick marks along bottom border        |     3
(2) Minute tick marks along top border           |
(3) Minute tick marks along top & bottom borders|
(4) Minute grid lines                            |
OR  No tick marks or grid lines                  |
.............................................................
Desired tick mark interval (No. minutes/tick):   |  1 <CR>
.............................................................
Draw Latitude Minute Tick Marks [Y]?             |     Y
.................................................|...........
(1) Minute tick marks along left border          |     3
(2) Minute tick marks along right border         |
(3) Minute tick marks along left & right         |
(4) Minute grid lines                            |
OR  No tick marks or grid lines                  |
                                                 |
Desired tick mark interval (No. minutes/mark):   |  1 <CR>
.................................................|...........
Draw Longitude Second Tick Marks [Y]?            |   <CR>
.................................................|...........
Draw Latitude Second Tick Marks [Y]?             |   <CR>
.................................................|...........
SELECT NEW PEN NUMBER?                           |   <CR>
  (No.: 1-6.  Or keep current pen.)              |
.................................................|...........
SELECT NEW LINE TYPE [Y]?                        |   <CR>
.................................................|...........
            (( Plotting of Oahu coastline. ))
.............................................................
       Plot data values [Y]?                     |     Y
.............................................................
SELECT NEW PEN NUMBER?                           |   <CR>
  (No.: 1-6.  Or keep current pen.)              |
.............................................................
(*) SELECTED SYMBOL IS          +                |
    ACCEPT THIS PLOT SYMBOL [Y or CR]?           |   <CR>
.............................................................
DATA ARE CURRENTLY PARTITIONED INTO TWO BINS     |
KEEP CURRENT PARTITIONING OR SELECT NEW          |
PARTITIONS [Y]?                                  |   <CR>
.............................................................
  (( A <CR> is used to keep the two-partition structure.))
  (( If a new partitioning is desired, a "Y" should be  ))
  (( entered.  Any other response is equivalent to <CR>.))
.............................................................
CURRENT DEFAULT SYMBOL SIZE IS 0.400 W X 0.576 H|
+ 0.04 W X 0.058 H CM INCREMENT INCREASE.        |
                                                 |
(Note:  The 1.44 h/w ratio produces a plot       |
        aspect ratio of 1.)                      |
KEEP DEFAULT OR SELECT NEW SIZE PARAMETERS [Y]?  |   <CR>
```

```
.............................................................
SELECT NEW LINE TYPE FOR CONTOUR [Y]?                |     Y

.............................................................
Select new line type:                                |
  (1) .       .       .       .                      |     6
  (2) ____    ____    ____                           |
  (3) _____   _____   _____                          |
  (4) _______._______._______.                       |
  (5) _____ _ _____ _ _____ _                        |
  (6) ____ _ _ ____ _ _ ____ _ _                     |
  Or _____________________                           |
.............................................................
New Line Type:  6                                    |

....................................................|...........
SELECT NEW PEN NUMBER?                               |   <CR>
  (No.: 1-6.  Or keep current pen.)                  |
.............................................................
    (( Plotter draws trackline on first pass and then ))
    (( draws symbols on second pass.                  ))
.............................................................
Label corner coordinates [Y]?                        |     Y
.............................................................
  (1) Label upper left corner                        |     1
  (2) Label lower left corner                        |     2
  (3) Label lower right corner                       |     3
  (4) Label upper right corner                       |     4
  OR  Continue                                       |    <CR>
.............................................................
 (( Plotter draws latitude and longitude at each corner.))
.............................................................
Label Plot?                                          |     Y
.............................................................
SELECT NEW PEN NUMBER?                               |   <CR>
  (No.: 1-6.  Or keep current pen.)                  |
.............................................................
Number of lines in label?                            |   1 <CR>
.............................................................
Default character size is 0.187 w X 0.269 h (cm)|
Accept default, or                                   |
              select new size parameters [Y]?        |   <CR>
.............................................................
Center of first label line is midway between the|
left and right boundaries and 1/10 down from the|
top boundary.                                        |
Accept current location, or select new location |
                                        [Y]?         |   <CR>
.............................................................
Enter text for line 1:                               | RB1 TRACK
                                                     | LINE <CR>
.............................................................
```

```
WRITE ADDITIONAL LABEL (Y/N)?                          |     Y
.......................................................................
SELECT NEW PEN NUMBER?                                 |   <CR>
  (No.: 1-6.  Or keep current pen.)                    |
.......................................................................
Number of lines in label?                              |   1 <CR>
.......................................................................
Default character size is 0.187 w X 0.269 h (cm)|
Accept default, or                                     |
              select new size parameters [Y]?          |     Y
.......................................................................
Enter new character width in cm + Return               | 0.12<CR>
.......................................................|...........
Enter new character height in cm + Return              | 0.17<CR>
.......................................................|...........
Center of first label line located midway               |     Y
between left and right paper boundaries and            |
1/10 from top boundary.  Accept current                |
location, or select new location [Y]?                  |
.......................................................|...........
Enter fractional distance from left to right           | 0.65 <CR>
boundaries (0.0-1.0) [Return to keep default]          |
.......................................................|...........
Enter fractional distance from bottom to top           | 0.75 <CR>
boundaries (0.0-1.0) [Return to keep default]          |
.......................................................|...........
    ((New lateral location:  0.65 from left to right  ))
    ((boundaries                                      ))
    ((New vertical location:  0.75 from bottom to top ))
    ((boundaries                                      ))
.......................................................................
Enter text for line 1:                                 | OAHU <CR>
.......................................................|...........
WRITE ADDITIONAL LABEL [Y/N]?                          |   <CR>
.......................................................................
Make new plot [Y] (else exit)?                         |   <CR>
.......................................................................
FINIS                                                  |
=======================================================================
```

If the data file contains more than one set of trackline data, each set must be separated by a record with zero entries. Such a record alerts HIMP that there is more than one line present and prevents connection between the terminating and initial points of successive tracks. In addition, during the plotting phase, the user also is asked whether a new line style is desired for the next track. When the default symbol size is accepted, bin 1 data (0) does not produce symbols during the data plot. This is because of the zero scale size selection for bin 1 during screen development. Once a zero scale is selected, no symbol will appear in either screen mode or plot mode for the corresponding bin data.

RB1 TRACKLINE



5 I/O DATA FILES

5.1 Introduction

The mapping program works with three basic types of data files. The first type is the cartographic data file containing the digitized coordinates of some cartographic feature or features, such as an island coastline or geographic contours. Composed of records containing two entries each, the cartographic or map-contour data file can be in either ASCII or Turbo Pascal binary. A second type, which is associated with the cartographic data file, is the map image file. This file is a binary screen dump of the map image constructed from the corresponding map data file. The third file type is the research data file, which contains three entries per record and lists the coordinates and value of a recorded data element. Like the cartographic data file, the research data file can be in either ASCII or binary form.

In the following discussion, each file type is described in detail. In addition to outlining the characteristic features of each file type, a brief presentation is given on how HIMP implements these files, followed by a brief discussion on the use of the partitioning option to allow data entries to control display features on the map image.

5.2 Cartographic Data Files

Cartographic data files contain a complete listing of the digitized data needed to display a given geographic feature. Such a feature might be the coastline of an island or group of islands, the depth contour, political boundaries, or research zones. The most common use of this file type is for the library maps found in the HMAP subdirectory. Each island or group in the library is represented by a corresponding binary data file, which can be identified by its bin extension.

External ASCII data files also may be used, but HIMP will produce a corresponding binary version of the files before the map image is created. This is done because graphics computations are much faster when binary files, rather than ASCII, are used.. Regardless of whether the original files are in ASCII or binary, all resulting output data files are in binary.

A cartographic data file begins with a record that gives the minimum and maximum latitude for that file. The second record lists the minimum and maximum longitude. Subsequent records list the digitized data points required to reproduce the given feature. Each record lists latitude and longitude for each point (see Fig. 5-1). HIMP uses the first two records to reconstruct the proper map frame setting. Depending on the selected option, the program uses the data file to either draw the map image or load the image from the corresponding image (.map) file. The latter option is used if a map image previously was created and stored in the image file.

Record No.	Entry 1	Entry 2	Comment
1	min latitude	max latitude	lat limits
2	min longitude	max longitude	long limits
3	latitude1	longitude1	first point coordinates
4	latitude2	longitude2	second point coordinates
.	.	.	.
.	.	.	.
.	.	.	.
N+2	latitudeN	longitudeN	last point coordinates

Figure 5-1.--Example of a cartographic (map-contour) file.

During screen development, an output cartographic data file is created with the same format as the input file. Once the user modifies the image to his or her specifications, the new latitude and longitude limits are written to the two leading records of the output file. The remaining data bounded by these limits are sequentially written to the output file.

New cartographic features may be added to the image by accessing the corresponding external data files. An example of this procedure is the addition of a 100-m depth contour to the existing map of Lanai. HIMP will first add a flag record, consisting of two -1 entries, to the output map data file. Next are records listing the digitized points that make up the contour line. The flag record is used to indicate that new or different data follow. HIMP displays these data by using a dashed line style. When the output file is used for plotting, the same flag record invokes a query for line style to be used to plot these data. (If a differentiation of the data is not desired, an editor on the ASCII version of the file may be used to remove the flag record. Subsequent use of this file will result in the contour line being displayed and plotted with the default solid line style.) A hypothetical example file in which two contours are added to a map file is illustrated in Figure 5-2. Normally the first record for each contour is a set of zeros to allow a "break" in the lines so that the contours are not connected to the island coastline or themselves.

5.3 Map Image Files

Once the desired basic map image is developed, HIMP stores the image in a binary output file. This "screen dump" is an exact copy of the screen image, including all cartographic features added from external files. The image file has the same name specification as the output data file, except

Record No.	Entry 1	Entry 2	Comment
1	min latitude	max latitude	lat limits
2	min longitude	max longitude	long limits
3	latitude1	longitude1	first point coordinates
4	latitude2	longitude2	second point coordinates
.	.	.	.
.	.	.	.
N+2	latitudeN	longitudeN	last point coordinates
N+3	-1	-1	new data flags
N+4	latitudeA1	longitudeA1	first point contour A
N+5	latitudeA2	longitudeA2	second point contour A
.	.	.	.
.	.	.	.
N+3+a	latitudeAa	longitudeAa	last point contour A
N+4+a	-1	-1	new data flags
N+5+a	latitudeB1	longitudeB1	first point contour B
N+6+a	latitudeB2	longitudeB2	second point contour B
.	.	.	.
.	.	.	.
N+4+a+b	latitudeBb	longitudeBb	last point contour B

Figure 5-2.--Example of an output file for island image with two contours added. The island image is from N points, contour A from a points, and contour B from b points.

with a map extension. For example, the image file corresponding to the default data file MapOut.bin is MapOut.map. Once the image file is created, subsequent invocations of HIMP retrieve and display the image almost instantaneously, in contrast to the more time consuming screen-drawing procedure.

As a reminder, image files should contain only cartographic features and not research data. Also note that drive and directory destinations can be used in the file specification, enabling the user to place both data and image files on diskettes or in other directories. The user can even access

a library map, modify it, and replace the old version with the modified version, but to avoid accidental overwrite or deletion of a good file, extreme caution should be used.

5.4 Research Data Files

Research data files contain a complete listing of locations and values for a given set of recorded data. All research data files contain records with three entries each. The entries for the first and second records are arbitrary and are not used in computations by HIMP. Because these records are just transferred to the output files, their contents are left to the discretion of the user. Normally information that might be useful to the user, such as geographic limits and data range, is supplied here. Subsequent records list the location by latitude and longitude, and the value of each data element. An example of a typical research data file is in Figure 5-3. In this example, the term "dummies" refers to data that are transferred to the output file but are not used in any computations. Bracketed quantities are examples of possible entries. As with cartographic data files, research data files can be either in ASCII or Turbo Pascal binary. HIMP will produce a binary version of any ASCII file before it is used on the screen display.

Record No.	Entry 1	Entry 2	Entry 3	Comment
1	arbitrary [min lat]	arbitrary [max lat]	arbitrary [min value]	dummies
2	arbitrary [min long]	arbitrary [max long]	arbitrary [max value]	dummies
3	latitude1	longitude1	value1	data 1
4	latitude2	longitude2	value2	data 2
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
N+2	latitudeN	longitudeN	valueN	data N

Figure 5-3.--An example of a research data file.

5.5 Data Types for Research Data Files

There are three general classifications for data that can be represented by research data files. These are station or catch data, trackline data, and longline data. Station (or catch) data represent individual or discrete data elements with an associated location. An example of this might be the number of lobsters caught at each of a series of traps over a period of time. Trackline data describe a path taken during a study. The data thus represent a continuous curve. An example of

these type of data might be the course a monitoring vessel took while tracking a tuna tagged with a signaling device. Longline data are a series of paired records representing the end points of line segments. This can be thought of as a listing of the terminal positions for a set of longlines.

Each type of data is handled differently by HIMP. For station data, points and values are displayed individually on the base map. For trackline data, points are connected to produce a continuous curve, except when a natural break is indicated by zeros in the first two entries of a record. For longline data, HIMP draws a straight line between successive pairs of points. Thus, six longline records, following the initial two arbitrary records, produce three lines on a map image. Twelve records produce six lines.

For station data, symbols or data values are placed at each data element location within the basic map image limits. For trackline and longline data, the line segment or segments are drawn first, then symbols or data values are placed at each data element location. For trackline data, placing symbols at each point along the curve may not be desired, especially if the data files are large or have a high density. Symbols may be desired only at points where hourly positional fixes were taken. One solution to this potential dilemma is having data value entries of "1" for all records corresponding to hourly fixes, and entries of "0" for all other records. The user can then invoke the two-bin partitioning option and assign an interval of [-0.1, 0.1] to the first bin and [0.1, 2] to the second. A symbol scale of 0 can be assigned to bin 1, and a scale of 1 to bin 2. As a result, only hourly fix locations are marked with a symbol. Similarly during plot development, a plot size of 0 can be assigned to bin 1 and a non-zero size increment assigned to bin 2, to produce the a hard copy of the results.

If a trackline file contains two or more actual tracklines, separation between successive lines is indicated by a flag record containing zero latitude and longitude entries. In the same manner, a composite trackline consisting of day and night segments may be differentiated. When the program encounters the zero entries during the screen development mode, a break will be generated, enabling discrete tracks to be drawn. Zeros in the plotting mode allow the user to select different line styles for each track. (To insure that lines start with the initial data point and remain continuous, coordinates should be repeated initially and after each flag.)

5.6 Station Location Option

The HIMP allows the user to display the station positions for a given data set, independent of the data display. Under this option, the program only uses the coordinates of each data record and displays a user-selected symbol for each location. Because this option is designed for individual or station-catch data, it should not be used with trackline or longline data. Results with such data are unpredictable and probably of little value.

During screen development the station location and data display options are distinct and intentionally kept separate. If both options are exercised, station symbols and data symbols will not appear simultaneously on the screen. In contrast, both types of symbols may be plotted during plot development. If station locations have been plotted, the user is given the option of selecting a one-character offset during data plotting, allowing a positioning (station) symbol to be used for location, then a data value to be placed next to it. For example, composite display elements might appear as *2.0 , +300 , or x1. Note that the size of the shift corresponds to the size of the symbol or character being used for that data element. For partitioned data sets, where more than one symbol size is possible, the shift distance will vary. Thus, a 0.1-cm symbol invokes a 0.1-cm shift, while a 0.28-cm symbol invokes a 0.28-cm shift.

5.7 File Construction and Coordinate System Constraints

Normally cartographic data and research data files may be constructed by using an editor, a digitizing table, or a combination of both. For simple files of small size, any editor (e.g., Wordstar, Turbo, or Edlin) may be used to enter individual records. This is appropriate for simple boundary or cellular contours, or small data sets. For geographic maps and contours and large cartographic data sets, the digitizing table and appropriate supporting software are desired. For trackline data, the digitizing table also is a good choice of input mechanisms. Data values can be entered simultaneously with the digitized data, or added with an editor at a later time. Time of entry depends on the type of software driver creating the file. For an outline of the available BASIC digitizing software, see Appendix IV.

All files should use coordinate values in terms of degrees and fractions of degrees. The digitizing software performs all conversions and supports calculations in units of degrees (no minutes or seconds). All editing of files should be done with this in mind. When data are located west of the dateline, absolute longitude should be used. Absolute longitude covers the range 0-360 degrees, with no reference to sign (±) or an east or west direction. (In this system, a longitude of 175.5°E equals 184.5°.) This is to conform to an absolute coordinate system and allow uniform geographic display computations in the HIMP utility algorithms. During plotting, HIMP converts from the absolute longitude system by using degrees only, then converts back to the conventional longitude system by using degrees, minutes, and seconds. (Latitudes also are converted back to degrees, minutes, and seconds.)

MELVIN:

Pages 72 and 74 are BLANK UNNUMBERED PAGES.

APPENDIXES

APPENDIX I

HIMP DETAILED FLOWCHART

```
                                  WITH OPTIONS:
 ____________________________________  (1) Archipelago    (Lo Res)
 |                                  |  (2) Windward group (Med Res)
 | OPENING MENU (OPTIONS 1-8)       |  (3) Leeward group  (Med Res)
 |                                  |  (4) Windward group (Hi Res)
 ------------------------------------  (5) Leeward group  (Hi Res)
                                 ||     (6) Windward Isle  (Hi Res)
                                 ||     (7) Leeward Isle   (Hi Res)
                                 \/     (8) Free form entry
 ____________________________________________________________________
[  |    |    |        |             |        |    |    |    ]
[ (1)  (2)  (3)      (4)           (5)      (6)  (7)  (8)  ]
[  |    |    |        |             |        |    |    |    ]
[  |    |    |  [~~~~~~~~~~] [~~~~~~~~~~]    |    |    |    ]
[  |    |    |  [Windward  ] [Leeward   ]    |    |    |    ]
[  |    |    |  [Individual] [Individual]    |    |    |    ]
[  |    |    |  [Isle Menu ] [Isle Menu ]    |    |    |    ]
[  |    |    |  [   (* 1)  ] [   (* 2)  ]    |    |    |    ]
[  |    |    |  ~~~~~~~~~~~~ ~~~~~~~~~~~~    |    |    |    ]
[  |    |    |        |             |        |    |    |    ]
[  v    v    v        v             v        v    v    v    ]
[  ..................................................      ]
[  :  1. Retrieve selected map files (.bin and .map). :    ]
[  :  2. Display map image.                           :    ]
[  :  3. Scan image and set new map limits.           :    ]
[  :  4. Draw new map image.                          :    ]
[  :..................................................:    ]
[____________________________________________________________]
                                 ||
(*1) WINDWARD INDIVIDUAL         ||      (*2) LEEWARD INDIVIDUAL
     ISLE MENU (Hi Res):         ||           ISLE MENU (Hi Res):
(1)  Hawaii                      ||      (1)  Nihoa
(2)  Maui                        ||      (2)  Necker
(3)  Kahoolawe                   ||      (3)  French Frigate S.
(4)  Lanai                       ||      (4)  Gardner Pinnacles
(5)  Molokai                     ||      (5)  Laysan
(6)  Oahu                        ||      (6)  Lisianski
(7)  Kauai                       ||      (7)  Midway
(8)  Niihau                      ||      (8)  Kure
                                 ||      (9)  Maro Reef
                                 ||      (0)  Pearl/Hermes Reef
                                 ||
                        (Basic Map Image)
                                 ||
                                 \/
```

```
 ____________________________________
[        Add Depth Contours          ]
[    (with optional line type)       ]
[____________________________________]
                 ||
                 ||
                 \/
 ____________________________________
[      Add Station Locations         ]
[     (with symbol selection)        ]
[____________________________________]
                 ||
                 ||
                 \/
 ____________________________________
[          Add Research Data         ]
[ (with symbol and/or line type)     ]
[____________________________________]
                 ||
                 ||
                 ||
      (Developed Map Image)
                 ||
                 ||
                 ||
                 \/
     TO CONTINUATION OPTIONS
                 ||
                 ||
                 ||
                 \/
 ____________________________________
| CONTINUATION (OUTPUT) MENU         |
| (1) Plot Output Map                |
| (2) Create Output Plot File        |
| (3) Exit HIMP                      |
|____________________________________|
     ||          ||          ||
 (Option 1) (Option 2) (Option 3)
     ||          ||          ||
     ||          ||          |_=========> EXIT
     ||          ||
     ||          |_=========> OUTPUT PLOT FILE
     ||
     \/
```

```
 ____________________________
| Plot Basic Map Image       |
|  * Chart Border            |      Control over pen
|  * Tick Marks/Gridlines    |      (color), line type,
|        (deg, min, sec)     |<==|| and tick or grid
|  * Coastlines and/or       |   || intervals
|          Depth Contours    |   ||
|____________________________|   ||
            ||                 ||
            \/                 ||
 ____________________________   || (Optional)
| Plot Station Locations     |   || Symbol control
|____________________________|   || (color, size, type)
            ||                 ||
            \/                 ||
 ____________________________   || (Optional)
| Plot Research Data         |   || Symbol control
|____________________________|   || (color, size, type)
            ||                 ||
            \/                 ||
 ____________________________   ||
| Add Corner Geographic      |   || (Optional)
|  Locations (long/lat)      |   || Any or all corners
|____________________________|   || in deg, min, sec
            ||                 ||
            \/                 ||
 ____________________________   || (Optional)
| Add Plot Labels            |   || Control over color,
|____________________________|   || size, location
            ||                 ||
            \/                 ||
 ____________________________   ||
| MAKE A NEW PLOT (YES/NO)?  | =(YES)
|____________________________| ===(NO)===>  EXIT
```

APPENDIX II

HIMP SUBROUTINE LIST

The following list is a breakdown of the supporting subroutines in the main HIMP program and HIMP inclusion files.

I. HIMP (Main Program)

Banner	HitRetToCon	MapWindow
ScaleWindow	Latlong	Latlong1
Scan	ScanI	ScanII
SetUp	ReSetUp	Select
Contour	DataPoint	StationPoint
InitCALR	InitWGMR	InitLGMR
InitWGHR	InitLGHR	InitIWIHR
InitILIHR	InitFF	

II. Himpplot

StationPlot	DataPlot	DrawXTick
DrawYTick	LongDegTick	LatDegTick
LongMinTick	LatMinTick	LongSecTick
LatSecTick	Cornerlabel	LabelPlot
HPPlot	CreateASCIIFile	CreatePlot

III. Himputil

Rep	Initiate	MessageW
TextBi2	TextBi3	PenSelect
LineSelect	LineStyle	Coordlabel

IV. Himpselect

SymbolSelect	DataSymbolSelect	PtScaleSelect
PtValueSelect		

APPENDIX III

PROGRAM LISTING

```
program  HIMP;

type
  sarray      = array[1..2] of real;
  tarray      = array[1..3] of real;
  WorkString  = String[30];

const
  sxm       = 640;       { max pixel screen size }
  sym       = 200;
  psxm      = 25.0;      { max physical screen image size, in cm. }
  psym      = 16.3;
  psar      = 0.652;     { physical screen aspect ratio, monitor-dependent }
  plus      = 'SM+;';    { HP code for "+" }
  cross     = 'SMx;';    { HP code for "x" }
  box       = 'UC99,6,0,0,8,-6,0,0,-8;';
     {HP  code to create a box symbol.}
  fbox      = 'UC99,6,0,0,8,-6,0,0,-8,3,4,3,0,-3,4,-3,-4,3,-4,3,4;';
     {HP code to create a filled box symbol.}
  dia       = 'UC3,0,99,3,4,-3,4,-3,-4,3,-4;';
     {HP code to create a diamond symbol.}
  yyy       = 'SMY;';    { HP code for "y" }
  star      = 'SM*;';    { HP code for "*" }
  oh        = 'SMo;';    { HP code for "O" }

var
  t                                          :tarray;
  s                                          :sarray;
  InFile,OutFile                             :file of sarray;
  InFile1,OutFile1                           :file of tarray;
  TextInDataName,TextInStatName              :WorkString;
  TextInFileName                             :WorkString;
  InFileName,OutFileName                     :string[30];
  InMapName,OutMapName                       :string[30];
  InStatName,OutStatName                     :string[30];
  InDataName,OutDataName                     :string[30];
     {String variables for plot symbols.}
  symbolstg,stnsymbolstg,datasymbolstg       :string[50];
     {Screen display symbols.}
  symbol,stnsymbol,datasymbol                :string[2];
  Stg                                        :String[255];
  sx1,sx2,sy1,sy2,wx1,wx2,wy1,wy2,wStg,hStg  :String[30];
  stglength                                  :String[3];
     {Screen display limits.}
  XGlbmin,XGlbmax,YGlbmin,YGlbmax            :integer;
  FileType                                   :char;
  ch,pen                                     :char;
```

```
  e,o,n                                          :char;
  aflag,Iflag,cflag                              :integer;
  i,stncflag,datacflag,bins                      :integer;
  scale,scalestn                                 :integer;
  scale0,scale1,scale2,scale3,scale4,scale5      :integer;
  nn,mm,err                                      :integer;
  p1x,p2x,p1y,p2y,dpx,dpy                        :integer;
  xi,yi,ai,bi,dxi,dyi,interval                   :integer;
  StnNo,FirstStnNo,LS                            :integer;
  latmin,latmax,longmin,longmax                  :real;
  latmin2,latmax2,longmin2,longmax2              :real;
  lattmp,longtmp                                 :real;
  bin0,bin1,bin2,bin3,bin4,bin5                  :real;
  dx,dy,x1,y1,x2,y2,a,b,c                        :real;
  d1z,d10z,z                                     :real;
  datamin,datamax                                :real;
  wsz1,wsz2,wsz3,wsz4,wsz5,wszhi                 :real;
  hsz1,hsz2,hsz3,hsz4,hsz5,hszhi                 :real;
  hwratio                                        :real;
  gar,gxm,gym,step                               :real;
  OK,Stationflag,Dataflag                        :Boolean;

{$I typedef.sys}
{$I graphix.sys}
{$I kernel.sys}
{$I windows.sys}
{$I Asynchp.Inc}
{$I himputil.pas}
{$I himpsele.pas}
{$I himpplot.pas}

procedure Banner;
     {Displays "HIMP" banner for 3 sec.}
  begin
    ClearScreen;
    SetForegroundColor(2);
    SetBackgroundColor(0);
    DefineWindow(3,5,0,75,150);
    DefineWorld(3,0.,100.,100.,0);
    SelectWorld(3);
    SelectWindow(3);
    SetBackGround(0);
    DrawBorder;
    Delay(1000);
    DrawTextW(15.,20.,10,'H I M P');
    DrawTextW(15.,80.,1,'HAWAIIAN ISLAND MAPPING PROGRAM');
    DrawTextW(15.,90.,1,'VERSION 1.3          4-16-87');
    Delay(1000);
    StoreWindow(3);
   for i := 1 to 2 do
   begin
     {Alternate display colors every 0.5 sec.}
```

```
    SetForegroundColor(14);
    ReStoreWindow(3,0,0);
    Delay(500);
    SetForegroundColor(2);
    ReStoreWindow(3,0,0);
    Delay(500);
   end;
  end;
{*************************************************************************}
procedure HitRetToCon;
     {Causes program to wait for a <CR> before continuing.}
  begin
    MessageW;
    DrawTextW(2.0,20.0,1,'HIT RETURN TO CONTINUE');
    Rep;
  end;
{*************************************************************************}
procedure MapWindow;
     {Clears screen and displays map development window.}
     {Restores a previously stored map image.}
  begin
    SelectWorld(1);
    SelectWindow(1);
    ClearScreen;
    RestoreWindow(1,0,0);
  end;
{*************************************************************************}
procedure ScaleWindow;
     {Determines screen display dimensions from geographic limits, displays
      the resulting window, and displays geographic limits.}
  begin
     {Determines differences in latitudes and longitudes.}
    dx:=longmax-longmin;
    dy:=latmax-latmin;
    DefineWorld(1,longmax,latmin,longmin,latmax);
    ClearScreen;
     {If geographic aspect ratio is less than screen aspect ratio, then let
      lateral limits fill maximum screen width.}
    if dy/dx < psar
       then begin
         YGlbmin:=Round(YMaxGlb*(1.0-(dy/dx)/psar)/2.0);
         YGlbmax:=Round(YMaxGlb*(1.0+(dy/dx)/psar)/2.0);
         DefineWindow(1,0,YGlbmin,XMaxGlb,YGlbmax);
       end;
     {Or let vertical limits fill maximum screen height.}
    if dy/dx >= psar
       then begin
         XGlbmin:=Round(XMaxGlb*(1.0-(dx/dy)*psar)/2.0);
         XGlbmax:=Round(XMaxGlb*(1.0+(dx/dy)*psar)/2.0);
         DefineWindow(1,XGlbmin,0,XGlbmax,YMaxGlb);
       end;
```

```
    SelectWorld(1);
    SelectWindow(1);
    DrawBorder;
    StoreWindow(1);
    gotoxy(1,1);
    WriteLn('PRESENT MAP LIMIT LINES');
    WriteLn('   Longmin :',longmin);
    WriteLn('   Longmax :',longmax);
    WriteLn('   Latmin  :',latmin);
    WriteLn('   Latmax  :',latmax);
  end;
{*************************************************************************}
Overlay Procedure Latlong;
     {Read in two-element records, containing latitudes and longitudes, and
      connect points to produce contour lines on screen.  This procedure is
      used with single file maps (options (1), (2), (3), (6), (7), (8)).}
  begin
     {Sends map limits to output file.  Map limits are determined by first
      two-file records or by scanning procedure.}
    s[1]:=latmin; s[2]:=latmax; Write(OutFile,s);
    s[1]:=longmin; s[2]:=longmax; Write(OutFile,s);
    x1:=0; y1:=0;
     {Start reading records.}
    Repeat
      Read(InFile,s);
     {Plot line change flag.  If -1, just pass to output file and read next
      record.}
        if ((s[1]=-1) and (s[2]=-1)) then begin
          write(OutFile,s); Read(InFile,s); end;
      a:=s[2]; b:=s[1];
     {If data point falls within map limits, draw connecting line.  If not,
      ignore point and read next record.}
      if (a>=longmin) and (a<=longmax) and
         (b>=latmin) and (b<=latmax) then begin
        if x1=0 then begin;
     {If this is first point, within limits, start with it.}
          x1:=a; y1:=b;
          s[1]:=0; s[2]:=0; Write(OutFile,s);
          s[1]:=b; s[2]:=a; Write(OutFile,s);
        end
     {Draw line between successive points, then make the terminology point
      the new initial point.}
        else begin
          x2:=a; y2:=b; Write(OutFile,s);
          DrawLine(x1,y1,x2,y2);
          x1:=x2; y1:=y2;
        end;
      end
      else begin
     {If point outside limits, then break line and start at next point that
      is inside limits.}
      x1:=0; y1:=0;
```

```
      end;
    until EOF(InFile);
end;
{*************************************************************************}
Overlay Procedure Latlong1;
     {Read in two-element record, containing latitudes and longitudes, and
      connect points to produce contour lines on screen.  This procedure is
      used with multi-file maps (options (4) and (5)).}
  begin
     {Since limits are previously set, ignore the first two records.}
    Read(InFile,s); Read(InFile,s);
    x1:=0; y1:=0;
     {Transfer of data from input to output files and drawing screen
display lines are the same as "latlong."}
    Repeat
      Read(InFile,s);
      a:=s[2]; b:=s[1];
      if (a>=longmin) and (a<=longmax) and
         (b>=latmin) and (b<=latmax) then begin
        if x1=0 then begin;
          x1:=a; y1:=b;
          s[1]:=0; s[2]:=0; Write(OutFile,s);
          s[1]:=b; s[2]:=a; Write(OutFile,s);
        end
        else begin
          x2:=a; y2:=b; Write(OutFile,s);
          DrawLine(x1,y1,x2,y2);
          x1:=x2;y1:=y2;
        end;
      end
      else begin
      x1:=0;y1:=0;
      end;
    until EOF(InFile);
end;
{*************************************************************************}
Overlay Procedure ScanI;
     {ScanI sets minimum and maximum longitudes for a given map.}
  begin
     {Start in middle, draw a vertical line from bottom to top of map
      frame.}
    z:=(longmax+longmin)/2.0;
    DrawLine(z,latmin,z,latmax);
    repeat
     {Read keyboard input to determine new location of vertical line.
      Remove old line, draw new line.}
      read(Kbd,ch);
      case ord(ch) of
     {Go left one step and draw line.}
          75 : begin
                 z:=z+d1z;
                 RestoreWindow(1,0,0);
```

```
                DrawLine(z,latmin,z,latmax);
              end;
     {Go right one step and draw line.}
          77 : begin
                z:=z-d1z;
                RestoreWindow(1,0,0);
                DrawLine(z,latmin,z,latmax);
              end;
     {Go left 10 steps and draw line.}
          52 : begin
                z:=z+d10z;
                RestoreWindow(1,0,0);
                DrawLine(z,latmin,z,latmax);
              end;
     {Go right 10 steps and draw line.}
          54 : begin
                z:=z-d10z;
                RestoreWindow(1,0,0);
                DrawLine(z,latmin,z,latmax);
              end;
       end;
     {Exit when space bar pressed.  Store map image with final vertical
      line.}
     until ch=' ';
   StoreWindow(1);
end;
{*************************************************************************}
Overlay Procedure ScanII;
     {ScanII sets minimum and maximum latitudes for a given map.}
  begin
     {Start in middle, draw a horizontal line from left to right map
      borders.}
    z:=(latmax+latmin)/2.0;
    DrawLine(longmin,z,longmax,z);
    repeat
     {Read keyboard input to determine new location at horizontal line.
      Remove old line, draw new line.}
      read(Kbd,ch);
      case ord(ch) of
     {Go up one step and draw line.}
          72 : begin
                z:=z+d1z;
                RestoreWindow(1,0,0);
                DrawLine(longmin,z,longmax,z);
              end;
     {Go down one step and draw line.}
          80 : begin
                z:=z-d1z;
                RestoreWindow(1,0,0);
                DrawLine(longmin,z,longmax,z);
              end;
```

```
          56 : begin
     {Go up 10 steps and draw line.}
                z:=z+d10z;
                RestoreWindow(1,0,0);
                DrawLine(longmin,z,longmax,z);
              end;
     {Go down 10 steps and draw line.}
          50 : begin
                z:=z-d10z;
                RestoreWindow(1,0,0);
                DrawLine(longmin,z,longmax,z);
              end;
      end;
     {Exit when space bar pressed.  Store map image with final horizontal
      line.}
    until ch=' ';
  StoreWindow(1);
end;
{*************************************************************************}
procedure SetUp;
     {SetUp queries user for output filename and allocates files for map
      binary data and image data.}
  begin
    ClearScreen;
    GotoXY(1,1);
    SetForegroundColor(2);   {GREEN}
    SetBackgroundColor(0);   {BLACK}
    WriteLn('Enter Output Data Filename (No Extension) : ');
    WriteLn('[Default --> "MapOut"] '); WriteLn;
    ReadLn(OutFileName);
     {If response is a null, then "MapOut.***" is the name of the output
      files.}
    if OutFileName='' then
      begin
        OutFileName:='MapOut';
      end;
    OutMapName:=OutFileName+'.map';
    OutFileName:=OutFileName+'.bin';
    WriteLn('Output Data File : ',OutFileName);
    WriteLn('Output Map File  : ',OutMapName);
     {Alternate binary input and output files.  Open said files.}
    Assign(InFile,InFileName); Reset(InFile);
    Assign(OutFile,OutFileName); ReWrite(OutFile);
    HitRetToCon;
     {Read first two records of input file for latitude and longitude
      limits.}
    Read(InFile,s);
    latmin:=s[1]; latmax:=s[2];
    Read(InFile,s);
    longmin:=s[1]; longmax:=s[2];
  end;
```

```
{*************************************************************************}
procedure ReSetUp;
     {Reopen input and output files, then read first two records of input
      file for longitude and latitude limits.  ReSetUp is used when a
      second pass of the data is required--not used for map files.}
  begin
    ReSet(InFile); ReWrite(OutFile);
    Read(InFile,s); latmin:=s[1]; latmax:=s[2];
    Read(InFile,s); longmin:=s[1]; longmax:=s[2];
  end;
{*************************************************************************}
procedure Scan;
     {Scanning procedure to determine new user-defined map limits.}
label label0;
  begin
     {Retrieve map image.}
    ReStoreWindow(1,0,0);
    MessageW;
    DrawTextW(2.,5.,1,'SET NEW MAP LIMITS [Y] ?');
    Read(Kbd,ch);
     {If no modification of limits is desired, exit scan and continue on
      with original limits.}
    if not (ch in ['Y','y']) then goto label0;
     {Determine (reciprocal) number of pixels per lateral step increment
      for scanning procedure.}
    d1z:=2.0*dx/sxm; d10z:=10.0*d1z;
    MessageW;
    DrawTextW(2.,5.,1,'SET MAP LIMITS');
    DrawTextW(2.,15.,1,'USE CURSOR CONTROLS TO MOVE LIMITS,');
    DrawTextW(2.,25.,1,'WHERE SHIFT-<ARROW> ==> 10X<ARROW>,');
    DrawTextW(2.,35.,1,'AND SPACE BAR SETS THE LIMITS.');
    DrawTextW(2.,45.,1,'(HIT RETURN TO CONTINUE.)');
    Rep;
     {Restore map image and set new minimum longitude limit.}
    MessageW;
    DrawTextW(2.,10.,1,'SET LONGMIN    (RIGHT LIMIT)');
    DrawTextW(2.,35.,1,'HIT RETURN TO COMMENCE');
    Rep;
     {Restore map image and set new maximum longitude limit.}
    MapWindow;
   ScanI;
    longmin2:=z;
    GoToXY(1,1);
    WriteLn('New longmin : ',longmin2);
    MessageW;
    DrawTextW(2.,10.,1,'SET LONGMAX    (LEFT LIMIT)');
    DrawTextW(2.,35.,1,'HIT RETURN TO COMMENCE');
    Rep;
     {Restore map image and set new minimum latitude limit.}
    MapWindow;
     {Determine (reciprocal) number of pixels per vertical step increment
      for scanning procedure.}
```

```
  ScanI;
    longmax2:=z;
    gotoxy(1,1);
    WriteLn('new longmax : ',longmax2);
    MessageW;
    DrawTextW(2.,10.,1,'SET LATMIN     (LOWER LIMIT)');
    DrawTextW(2.,35.,1,'HIT RETURN TO COMMENCE');
  Rep;
     {Restore map image and set new maximum latitude limit.}
  MapWindow;
    d1z:=dy/sym; d10z:=10.0*d1z;
  ScanII;
    latmin2:=z;
    gotoxy(1,1);
    WriteLn('new latmin : ',latmin2);
  MessageW;
    DrawTextW(2.,10.,1,'SET LATMAX     (UPPER LIMIT)');
    DrawTextW(2.,35.,1,'HIT RETURN TO COMMENCE');
  Rep;
  MapWindow;
  ScanII;
    latmax2:=z;
    gotoxy(1,1);
    WriteLn('new latmax : ',latmax2);
    HitRetToCon;
     {Record new limits.}
    longmin:=longmin2;
    longmax:=longmax2;
    latmin:=latmin2;
    latmax:=latmax2;
label10:  end;
{*************************************************************************}
Overlay Procedure InitCALR;  {Complete Archipelago - Low Res}
label label10;
  begin
     {Point to the low res. archipelago map data file in HMAP library.}
    InFileName:='hmap\archlr.bin';
     {Get output filename(s) from user.}
  Setup;
     {Set up map window, display archipelago map, and scan map for new
      limits.}
label10:  ScaleWindow;
    HitRetToCon;
    SetBackGroundColor(0);   {black}
    Clearscreen;
    LoadWindow(1,-1,-1,'hmap\archlr.map');
    StoreWindow(1);
    HitRetToCon;
  Scan;
     {Set up new map window, display window, and draw new map inside
      window.}
```

```
  ScaleWindow;
  MessageW;
    DrawTextW(2.,5.,1,'READY FOR NEW MAP');
    DrawTextW(2.,35.,1,'HIT RETURN TO COMMENCE');
  Rep;
  MapWindow;
  Latlong;
     {Save new map image and query user if image should be kept or new
      image produced.}
    SaveWindow(1,OutMapName);
  MessageW;
    DrawTextW(2.,5.,1,'CONTINUE, OR REDO MAP [N] ?');
    Read(Kbd,ch);
    if ch in ['N','n'] then begin
        ReSetUp;
        Goto label0
      end;
    Iflag:=0;
    Close(InFile);
    end;
{*************************************************************************}
Overlay Procedure InitWGMR;  {Windward Group - Med Res}
label label0;
  begin
     {Point to medium res. windward group data file in HIMP library.}
    InFileName:='hmap\wgmr.bin';
     {Get output filename(s) from user.}
  Setup;
     {Set up window, display group map, and scan new limits.}
label0:  ScaleWindow;
    HitRetToCon;
    SetBackGroundColor(0);   {black}
    Clearscreen;
    LoadWindow(1,-1,-1,'hmap\wgmr.map');
    StoreWindow(1);
    HitRetToCon;
  Scan;
     {Set up new map window, display window, and draw new map inside
      window.}
  ScaleWindow;
  MessageW;
    DrawTextW(2.,5.,1,'READY FOR NEW MAP');
    DrawTextW(2.,35.,1,'HIT RETURN TO COMMENCE');
  Rep;
  MapWindow;
  Latlong;
     {Save new map image and query user if image should be kept or new
      image produced.}
    SaveWindow(1,OutMapName);
  MessageW;
    DrawTextW(2.,5.,1,'CONTINUE, OR REDO MAP [N] ?');
    Read(Kbd,ch);
```

```
    if ch in ['N','n'] then begin
        ReSetUp;
        Goto label0
      end;
    Iflag:=0;
    Close(InFile);
    end;
{*************************************************************************}
Overlay Procedure InitLGMR;  {Leeward Group - Med Res}
     {Same procedure as in InitWGMR, but for leeward group.}
label label0;
  begin
      InFileName:='hmap\lgmr.bin';
  Setup;
label0:  ScaleWindow;
    HitRetToCon;
    SetBackGroundColor(0);   {black}
    Clearscreen;
    LoadWindow(1,-1,-1,'hmap\lgmr.map');
    StoreWindow(1);
    HitRetToCon;
  Scan;
  ScaleWindow;
  MessageW;
    DrawTextW(2.,5.,1,'READY FOR NEW MAP');
    DrawTextW(2.,35.,1,'HIT RETURN TO COMMENCE');
  Rep;
  MapWindow;
  Latlong;
    SaveWindow(1,OutMapName);
  MessageW;
    DrawTextW(2.,5.,1,'CONTINUE, OR REDO MAP [N] ?');
    Read(Kbd,ch);
    if ch in ['N','n'] then begin
        ReSetUp;
        Goto label0
      end;
    Iflag:=0;
    Close(InFile);
    end;
{*************************************************************************}
Overlay Procedure InitWGHR;  {Windward Group - Hi Res}
label label0;
  begin
     {Point to medium res. windward group data file in HIMP library.}
    InFileName:='hmap\wgmr.bin';
     {Get output filename(s) from user.}
  Setup;
     {Set up window, display group map, and scan map for new limits.}
label0:  ScaleWindow;
    HitRetToCon;
    SetBackGroundColor(0);   {black}
```

```
    Clearscreen;
    LoadWindow(1,-1,-1,'hmap\wgmr.map');
    StoreWindow(1);
    HitRetToCon;
  Scan;
    s[1]:=latmin; s[2]:=latmax; Write(OutFile,s);
    s[1]:=longmin; s[2]:=longmax; Write(OutFile,s);
    Close(InFile);   {Close group file.}
  ScaleWindow;   {Construct window based on new limits.}
  MessageW;
    DrawTextW(2.,5.,1,'READY FOR NEW MAP');
    DrawTextW(2.,35.,1,'HIT RETURN TO COMMENCE');
  Rep;
     {Display new blank map window.}
  MapWindow;
     {Retrieve binary data from each individual hi res. map file that fall
      within new limits and draw new map.  Append data to output file.
      Latlong1 used to ignore first two records of each file.}
{HAWAII inclusion}
    if ((latmax >= 18.8) and (latmin <= 20.4) and (longmax >= 154.3)
       and (longmin <= 156.6)) then begin
         Assign(InFile,'hmap\hawahr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{MAUI inclusion}
    if ((latmax >= 20.5) and (latmin <= 21.1) and (longmax >= 155.9)
       and (longmin <= 156.8)) then begin
         Assign(InFile,'hmap\mauihr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{KAHOOLAWE inclusion}
    if ((latmax >= 20.4) and (latmin <= 20.65) and (longmax >= 156.5)
       and (longmin <= 156.75)) then begin
         Assign(InFile,'hmap\kahohr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{LANAI inclusion}
    if ((latmax >= 20.65) and (latmin <= 20.95) and (longmax >= 156.7)
       and (longmin <= 157.2)) then begin
         Assign(InFile,'hmap\lanahr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{MOLOKAI inclusion}
    if ((latmax >= 21.00) and (latmin <= 21.3) and (longmax >= 156.6)
       and (longmin <= 157.4)) then begin
         Assign(InFile,'hmap\molohr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
```

```
        end;
{OAHU inclusion}
    if ((latmax >= 21.25) and (latmin <= 21.75) and (longmax >= 157.5)
       and (longmin <= 158.4)) then begin
         Assign(InFile,'hmap\oahuhr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{KAUAI inclusion}
    if ((latmax >= 21.75) and (latmin <= 22.3) and (longmax >= 159.1)
       and (longmin <= 160.0)) then begin
         Assign(InFile,'hmap\kauahr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{NIIHAU inclusion}
    if ((latmax >= 21.6) and (latmin <= 22.2) and (longmax >= 159.95)
       and (longmin <= 160.4)) then begin
         Assign(InFile,'hmap\niihhr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
    SaveWindow(1,OutMapName);
  MessageW;
  DrawTextW(2.,5.,1,'CONTINUE, OR REDO MAP [N] ?');
  Read(Kbd,ch);
     {If new map is desired, go back and retrieve group map data file and
      start again.}
  if ch in ['N','n'] then begin
      Assign(InFile,'hmap\wgmr.bin');
      ReSetUp;
      Goto label0
    end;
  Iflag:=0;
 end;
{*************************************************************************}
Overlay Procedure InitLGHR;  {Leeward Group - Hi Res}
     {Same procedures used as in InitWGHR above, but with individual hi
      res. leeward files.}
label label0;
  begin
      InFileName:='hmap\lgmr.bin';
  Setup;
label0:  ScaleWindow;
    HitRetToCon;
    SetBackGroundColor(0);   {black}
    Clearscreen;
    LoadWindow(1,-1,-1,'hmap\lgmr.map');
    StoreWindow(1);
    HitRetToCon;
  Scan;
    s[1]:=latmin; s[2]:=latmax; Write(OutFile,s);
```

```
    s[1]:=longmin; s[2]:=longmax; Write(OutFile,s);
    Close(InFile);
  ScaleWindow;
  MessageW;
    DrawTextW(2.,5.,1,'Ready for new map');
    DrawTextW(2.,35.,1,'Hit Return to commence');
  Rep;
  MapWindow;
{NIHOA inclusion}
    if ((latmax >= 23.0) and (latmin <= 23.1) and (longmax >= 161.8)
       and (longmin <= 162.0)) then begin
         Assign(InFile,'hmap\nihohr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{NECKER inclusion}
    if ((latmax >= 23.5) and (latmin <= 23.6) and (longmax >= 164.5)
       and (longmin <= 165.0)) then begin
         Assign(InFile,'hmap\neckhr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{FRENCH FRIGATE SHOALS inclusion}
    if ((latmax >= 23.5) and (latmin <= 24.0) and (longmax >= 166.0)
       and (longmin <= 166.5)) then begin
         Assign(InFile,'hmap\ffs10fhr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{GARDNER PINNACLES inclusion}
    if ((latmax >= 24.2) and (latmin <= 25.1) and (longmax >= 167.9)
       and (longmin <= 168.4)) then begin
         Assign(InFile,'hmap\gp20fhr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{MARO REEF inclusion}
    if ((latmax >= 25.0) and (latmin <= 26.0) and (longmax >= 170.0)
       and (longmin <= 171.0)) then begin
         Assign(InFile,'hmap\marohr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{LAYSAN inclusion}
     if ((latmax >= 25.5) and (latmin <= 26.0) and (longmax >= 171.5)
       and (longmin <= 172.0)) then begin
         Assign(InFile,'hmap\layshr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
```

```
{LISIANSKI inclusion}
     if ((latmax >= 26.0) and (latmin <= 26.5) and (longmax >= 173.5)
       and (longmin <= 174.0)) then begin
         Assign(InFile,'hmap\lisihr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{MIDWAY inclusion}
     if ((latmax >= 28.0) and (latmin <= 28.3) and (longmax >= 177.0)
       and (longmin <= 177.7)) then begin
         Assign(InFile,'hmap\midwhr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{KURE inclusion}
     if ((latmax >= 28.3) and (latmin <= 28.6) and (longmax >= 178.0)
       and (longmin <= 178.6)) then begin
         Assign(InFile,'hmap\kur18fhr.bin'); ReSet(InFile);
         latlong1;
         Close(InFile);
       end;
{PEARL AND HERMES inclusion}
     if ((latmax >= 27.65) and (latmin <= 28.0) and (longmax >= 175.65)
        and (longmin <= 176.0)) then begin
          Assign(InFile,'hmap\phhr.bin'); ReSet(InFile);
          latlong1;
          Close(InFile);
        end;
    SaveWindow(1,OutMapName);
  MessageW;
    DrawTextW(2.,5.,1,'CONTINUE, OR REDO MAP [N] ');
    Read(Kbd,ch);
    if ch in ['N','n'] then begin
        Assign(InFile,'hmap\lgmr.bin');
        ReSetUp;
        Goto label0
      end;
    Iflag:=0;
    end;
{*************************************************************************}
Overlay Procedure InitIWIHR;  {Individual Windward Isle - Hi Res}
label label0,label1;
  begin
    ClearScreen;
      GotoXY(1,1);
label0:  WriteLn('Select Island :'); WriteLn;
     {Secondary menu to select individual island.}
      WriteLn('  (1) Hawaii ');
      WriteLn('  (2) Maui   ');
      WriteLn('  (3) Kahoolawe');
      WriteLn('  (4) Lanai  ');
      WriteLn('  (5) Molokai');
```

```
      WriteLn('  (6) Oahu   ');
      WriteLn('  (7) Kauai  ');
      WriteLn('  (8) Niihau '); WriteLn;
      WriteLn('SELECT BY NUMBER');
      Read(Kbd,ch);
     {Based on user selection, point to corresponding individual hi res.
      island files.}
      Case ord(ch) of
        49 : begin
               InFileName:='hmap\hawahr.bin';
               InMapName:='hmap\hawahr.map';
             end;
        50 : begin
               InFileName:='hmap\mauihr.bin';
               InMapName:='hmap\mauihr.map';
             end;
        51 : begin
               InFileName:='hmap\kahohr.bin';
               InMapName:='hmap\kahohr.map';
             end;
        52 : begin
               InFileName:='hmap\lanahr.bin';
               InMapName:='hmap\lanahr.map';
             end;
        53 : begin
               InFileName:='hmap\molohr.bin';
               InMapName:='hmap\molohr.map';
             end;
        54 : begin
               InFileName:='hmap\oahuhr.bin';
               InMapName:='hmap\oahuhr.map';
             end;
        55 : begin
               InFileName:='hmap\kauahr.bin';
               InMapName:='hmap\kauahr.map';
             end;
        56 : begin
               InFileName:='hmap\niihhr.bin';
               InMapName:='hmap\niihhr.map';
             end;
           else begin
             ClearScreen; GotoXY(1,1);
             WriteLn('AVAILABLE OPTIONS ARE (1) - (8)');
             Goto label0;
           end;
      end;
  SetUp;
     {Ask user for output filename.}
     {Display hi res. map image and scan for new limits.}
label1:  ScaleWindow;
    HitRetToCon;
    SetBackgroundColor(0);   {black}
```

```
    ClearScreen;
    LoadWindow(1,-1,-1,InMapName);
    StoreWindow(1);
    HitRetToCon;
  Scan;
  ScaleWindow;
     {Display window based on new map limits and draw new map inside
      window.}
  MessageW;
    DrawTextW(2.,5.,1,'READY FOR NEW MAP');
    DrawTextW(2.,35.,1,'HIT RETURN TO COMMENCE');
  Rep;
  MapWindow;
  Latlong;
    SaveWindow(1,OutMapName);
  MessageW;
     {Redraw option.}
    DrawTextW(2.,5.,1,'CONTINUE, OR REDO MAP [N] ?');
    Read(Kbd,ch);
    if ch in ['N','n'] then begin
        ReSetUp;
        Goto label1
      end;
    Iflag:=0;
    Close(InFile);
    end;
{*************************************************************************}
Overlay Procedure InitILIHR;  {Individual Leeward Isle - Hi Res}
     {Same procedure as used in InitIWIHR above, but for leeward islands.}
label label0, label1;
  begin
      ClearScreen;
      GotoXY(1,1);
label0:  WriteLn('Select Island :'); WriteLn;
      WriteLn('  (1) Nihoa  ');
      WriteLn('  (2) Necker ');
      WriteLn('  (3) French Frigate Shoals 10f ');
      WriteLn('  (4) Gardner Pinnacles 20f ');
      WriteLn('  (5) Laysan ');
      WriteLn('  (6) Lisianski ');
      WriteLn('  (7) Midway and Reef ');
      WriteLn('  (8) Kure 18f ');
      WriteLn('  (9) Maro Reef 10f');
      WriteLn('  (0) Pearl/Hermes Reef 10f '); WriteLn;
      WriteLn('NOTE:  Fathom Contour Inicated By "nnf"'); WriteLn;
      WriteLn('SELECT BY NUMBER');
      Read(Kbd,ch);
      Case ord(ch) of
        49 : begin
               InFileName:='hmap\nihohr.bin';
               InMapName:='hmap\nihohr.map';
             end;
```

```
        50 : begin
               InFileName:='hmap\neckhr.bin';
               InMapName:='hmap\neckhr.map';
             end;
        51 : begin
               InFileName:='hmap\ffs10fhr.bin';
               InMapName:='hmap\ffs10fhr.map';
             end;
        52 : begin
               InFileName:='hmap\gp20fhr.bin';
               InMapName:='hmap\gp20fhr.map';
             end;
        53 : begin
               InFileName:='hmap\layshr.bin';
               InMapName:='hmap\layshr.map';
             end;
        54 : begin
               InFileName:='hmap\lisihr.bin';
               InMapName:='hmap\lisihr.map';
             end;
        55 : begin
               InFileName:='hmap\midwhr.bin';
               InMapName:='hmap\midwhr.map';
             end;
        56 : begin
               InFileName:='hmap\kur18fhr.bin';
               InMapName:='hmap\kur18fhr.map';
             end;
        57 : begin
               InFileName:='hmap\marohr.bin';
               InMapName:='hmap\marohr.map';
             end;
        48 : begin
               InFileName:='hmap\phhr.bin';
               InMapName:='hmap\phhr.map';
             end;
           else begin
             ClearScreen; GotoXY(1,1);
             WriteLn('AVAILABLE OPTIONS ARE (1) -(0)');
             Goto label0;
           end;
      end;
  SetUp;
label1:  ScaleWindow;
    HitRetToCon;
    SetBackgroundColor(0);   {black}
    ClearScreen;
    LoadWindow(1,-1,-1,InMapName);
    StoreWindow(1);
    HitRetToCon;
  Scan;
  ScaleWindow;
```

```
  MessageW;
    DrawTextW(2.,5.,1,'Ready for new map');
    DrawTextW(2.,35.,1,'HIT RETURN TO COMMENCE');
  Rep;
  MapWindow;
  Latlong;
    SaveWindow(1,OutMapName);
  MessageW;
    DrawTextW(2.,5.,1,'CONTINUE, OR REDO MAP [N] ?');
    Read(Kbd,ch);
      if ch in ['N','n'] then begin
        ReSetUp;
        Goto label1
      end;
    Iflag:=0;
    Close(InFile);
    end;
{*************************************************************************}
Overlay Procedure InitFF;
     {Freeform Entry}
  label  label0,label1;
  begin
    ClearScreen; GotoXY(1,1);
     {Ask user for input filename.  If ASCII, produce corresponding binary
file, using TextBi2.}
    Repeat
      WriteLn('Enter Input Filename (with extension) : ');
      ReadLn(TextInFileName);
        InFileName:=TextInFileName;
      Assign(InFile,InFileName);
      {$I-} Reset(InFile) {$I+}; OK:=(IOresult=0);
      if not OK then WriteLn('Cannot find file ',InFileName);
      Close(InFile); WriteLn;
    until OK;
    Repeat
      WriteLn('Is input file in ASCII [A] or Pascal binary (.bin) [B] ?');
      Read(Kbd,FileType);
      if FileType in ['A','a'] then begin
        Delete(InFileName,(length(InFileName)-2),3);
        InFileName:=InFileName+'bin';
        TextBi2(TextInFileName,InFileName);
        Assign(InFile,InFileName);
        ClearScreen; GotoXY(1,1);
        WriteLn('The binary input file ',InFileName,' has been created');
        WriteLn('and will be used for input by HIMP.');
      end;
    until FileType in ['A','a','B','b'];
      Delete(TextInFileName,(length(TextInFileName)-2),3);
      InMapName:=TextInFileName+'map';
    WriteLn('Input Data File : ',InFileName);
    WriteLn('Input Map File  : ',InMapName);
```

```
    MessageW;
     {Ask user if map image exists.  If yes, point to filename.map.  If
      no, draw image base on filename.bin.  Set flag to indicate if image
      exists already.}
    DrawTextW(2.,20.,1,'Does map file already exist (Y/N) ?');
      Read(Kbd,ch);
        if ch in ['Y','y'] then Iflag:=1 else Iflag:=0;
Label0:
    Setup; ScaleWindow;
    MessageW;
    DrawTextW(2.,5.,1,'HIT RETURN TO CONTINUE');
    Rep; ClearScreen;
     {Load in image if it exists.}
        if Iflag=1 then begin
          LoadWindow(1,-1,-1,InMapName); Reset(InFile);
          Repeat Read(InFile,s); Write(OutFile,s); until EOF(InFile);
        end;
     {Draw image if it does not exist already.}
        if Iflag=0 then begin
          MapWindow;
          LatLong;
          SaveWindow(1,InMapName);
          Iflag:=1;
        end;
     {Store image.}
      StoreWindow(1);
      MessageW;
      DrawTextW(2.,5.,1,'Modify Map [Y] ? ');
      Read(Kbd,ch);
        if not (ch in ['Y','y']) then goto label1;
     {Scan for new limits; draw new map.}
      Scan;
  ScaleWindow;
    MessageW;
      DrawTextW(2.,5.,1,'Ready for new map');
      DrawTextW(2.,35.,1,'Hit Return to commence');
    Rep;
    MapWindow;
              Reset(InFile); Rewrite(OutFile);
              Read(InFile,s); Read(InFile,s);
  Latlong; StoreWindow(1); Iflag:=0;
label1: MapWindow; SaveWindow(1,OutMapName);
    MessageW;
     {Redraw option.}
  DrawTextW(2.,5.,1,'CONTINUE, OR REMODIFY [N] ?');
  Read(Kbd,ch);
    if ch in ['N','n'] then begin
        Close(InFile); Close(OutFile); Iflag:=1;
        Goto label0;
      end;
    Close(InFile);
  end;
```

```
{*************************************************************************}
Overlay Procedure DepthContour;  {Depth Contour Entry}
  begin
    SetLineStyle(4); LS:=4;
    Repeat
     {Write (-1) flags to output, denoting a new contour line.}
      s[1]:=-1.0; s[2]:=-1.0; Write(OutFile,s);
      ClearScreen; GotoXY(1,1); WriteLn('Current linestyle is ',LS);
      Repeat
     {Ask user for new input file.}
        WriteLn('Enter Input Filename (with extension) : ');
        ReadLn(TextInFileName); InFileName:=TextInFileName;
        Assign(InFile,InFileName);
        {$I-} Reset(InFile) {$I+}; OK:=(IOresult=0);
        if not OK then WriteLn('Cannot find file ',InFileName);
        Close(InFile); WriteLn;
      until OK;
      Repeat
     {Ask user if file is ASCII or binary.  If ASCII, make a corresponding
      binary file.}
        WriteLn('Is input file in ASCII [A] or Pascal binary (.bin) [B] ?');
        Read(Kbd,FileType);
        if FileType in ['A','a'] then begin
          Delete(InFileName,(length(InFileName)-2),3);
          InFileName:=InFileName+'bin';
          TextBi2(TextInFileName,InFileName);
          Assign(InFile,InFileName);
          ClearScreen; GotoXY(1,1);
          WriteLn('The binary input file ',InFileName,' has been created');
          WriteLn('and will be used for input by HIMP.');
        end;
      until FileType in ['A','a','B','b'];
      WriteLn('Input Data File : ',InFileName); WriteLn;
     {New line type option.}
      WriteLn('Select new line style [Y] ? '); Read(Kbd,ch);
      if ch in ['Y','y'] then LineStyle;
      HitRetToCon; ClearScreen; Reset(InFile);
      MapWindow; LoadWindow(1,-1,-1,OutMapName);
      Latlong1;
      SaveWindow(1,OutMapName); StoreWindow(1);
      MessageW;
     {Ask user if an additional contour is desired.  If yes, close input
      file and repeat procedure.}
      DrawTextW(2.,5.,1,'ADD ADDITIONAL CONTOURS [Y] ?');
      Read(Kbd,ch); Close(InFile);
    until not (ch in ['Y','y']);
    SetLineStyle(0); Iflag:=0;
  end;
{*************************************************************************}
procedure Select;
     {Select mapping option.}
label label0;
```

```
  begin
    ClearScreen; GotoXY(1,1);
     {Based on selected option, call the corresponding procedure.}
label10:    WriteLn('Select Hawaii Mapping Option : '); WriteLn;
    WriteLn('  (1) Complete Archipelago (Low Res)');
    WriteLn('  (2) Windward Group (Med Res)');
    WriteLn('  (3) Leeward Group (Med Res)');
    WriteLn('  (4) Windward Group (Hi Res)');
    WriteLn('  (5) Leeward Group (Hi Res)');
    WriteLn('  (6) Individual Windward Isle (Hi Res)');
    WriteLn('  (7) Individual Leeward Isle (Hi Res)');
    WriteLn('  (8) Freeform Entry '); WriteLn;
    WriteLn('SELECT BY NUMBER');
    Read(Kbd,ch);
    Case ord(ch) of
      49 : InitCALR;
      50 : InitWGMR;
      51 : InitLGMR;
      52 : InitWGHR;
      53 : InitLGHR;
      54 : InitIWIHR;
      55 : InitILIHR;
      56 : InitFF;
         else begin
     {If user enters a nonoption key, display available option message and
      re-inquire.}
           ClearScreen; GotoXY(1,1);
           WriteLn('AVAILABLE OPTIONS ARE (1) - (8)');
           Goto label10;
         end;
    end;
  end;
{*************************************************************************}
Overlay Procedure DataPoint;
     {Procedure to display and record data points.}
label label0,label1;
procedure jump;
     {Procedure for jumping from one data point to another, without a line
      connection.}
  begin
    read(InFile1,t); x1:=t[2]; y1:=t[1]; read(InFile1,t); x1:=t[2]; y1:=t[1];
  end;
  begin
    Repeat
      ClearScreen; GotoXY(1,1);
     {Ask user which type of data is represented.}
      WriteLn('WHAT IS THE DATA TYPE ?'); WriteLn('  (1) Station / catch ');
      WriteLn('  (2) Trackline '); WriteLn('  (3) Longline ');
      Read(Kbd,ch); WriteLn; Val(ch,aflag,err);
    until ch in ['1','2','3'];
```

```
Case aflag of
  1 : WriteLn('    STATION /CATCH DATA');
  2 : WriteLn('    TRACKLINE DATA ');
  3 : WriteLn('    LONGLINE DATA ');
end; Delay(1000);
ClearScreen; GotoXY(1,1);
Repeat
 {Ask user for input filename.}
  WriteLn('Enter Input Data Filename (with extension) : ');
  ReadLn(TextInDataName);
  InDataName:=TextInDataName;
  Assign(InFile1,InDataName);
  {$I-} ReSet(InFile1) {$I+}; OK:=(IOresult=0);
  if not OK then WriteLn('Cannot find data file ',InDataName);
  Close(InFile1); WriteLn;
until OK;
Repeat
 {Ask if file is ASCII or binary.  If ASCII, create corresponding
  binary file by using Text Bi.}
  WriteLn('Is input data file in ASCII [A] or Pascal binary
  (.bin) [B] ?');
  Read(Kbd,FileType);
  if FileType in ['A','a'] then begin
    Delete(InDataName,(length(InDataName)-2),3);
    InDataName:=InDataName+'bin';
    TextBi3(TextInDataName,InDataName);
    ClearScreen; GotoXY(1,1);
    WriteLn('The binary input file ',InDataName,' has been created');
    WriteLn('and will be used for input by HIMP.');
  end;
until FileType in ['A','a','B','b'];
WriteLn('Input Data File :',InDataName); WriteLn;
 {Ask for output filename.}
WriteLn('Enter Output Data Filename (no extension) [OutData] : ');
WriteLn('(Binary File Containing Coordinates and Data)');
 {Default "OutData.***"}
ReadLn(OutDataName); if OutDataName='' then OutDataName:='OutData';
  OutDataName:=OutDataName+'.bin';
WriteLn('Output Data File : ',OutDataName); WriteLn;
Assign(InFile1,InDataName); Reset(InFile1);
Assign(OutFile1,OutDataName); ReWrite(OutFile1);
WriteLn; HitRetToCon;
 {Select data symbol for screen display.}
DataSymbolSelect; bins:=1;
ClearScreen; GotoXY(1,1);
WriteLn('Select Symbol Scale Proportional to Data Value [Y] ?');
 {Data partitioning option.}
Read(Kbd,ch); scale1:=0;
if ch in ['y','Y'] then PtValueSelect else PtScaleSelect;
 {If yes, call PtValueSelect to determine number of data bins and
  corresponding partitions.  If no, just select single symbol display
  size from data.}
```

```
MapWindow; LoadWindow(1,-1,-1,OutMapName);
 {Bypass first two records, then write limits out to the first two
  records in the output file.}
Read(InFile1,t); Read(InFile1,t);
t[1]:=latmin; t[2]:=latmax; t[3]:=0.0; Write(OutFile1,t);
t[1]:=longmin; t[2]:=longmax; t[3]:=0.0; Write(OutFile1,t);
 {If data type is trackline (aflag = 2), connect data points with
  breaks whenever 0-records are encountered.}
if aflag = 2 then begin
  Read(InFile1,t); x1:=t[2]; y1:=t[1];
  Repeat
    Read(InFile1,t); if t[2]=0.0 then jump; x2:=t[2]; y2:=t[1];
    Drawline(x1,y1,x2,y2); x1:=x2; y1:=y2;
  until EOF(InFile1);
  Reset(InFile1);
end;
 {If data type is longline (aflag = 3), connect paired points, with
  addition breaks  whenever 0-records are encountered.}
if aflag = 3 then begin
  Repeat
    Read(InFile1,t); if t[2]=0.0 then jump; x1:=t[2]; y1:=t[1];
    if EOF(InFile1) then begin
      x2:=x1; y2:=y1;
    end else begin
      Read(InFile1,t); x2:=t[2]; y2:=t[1];
      Drawline(x1,y1,x2,y2);
    end;
  until EOF(InFile1);
 {Reset file (point to first record) for second run through data.}
  Reset(InFile1);
end;
if cflag = 2 then begin
 {If data values are to be used as display symbols (cflag = 2), then
  capture value, convert to character, and display at corresponding
  location, if with map limits.}
    Repeat
      Read(InFile1,t); if t[2]=0.0 then begin
        write(OutFile1,t); goto label0; end;
      a:=t[2]; b:=t[1]; c:=t[3];
        Str(trunc(t[3]),Stg);
 {mm = significant figures.}
        nn:=length(Stg)+mm;
        Str(t[3]:nn:mm,Stg);
      if (a>=longmin) and (a<=longmax) and
         (b>=latmin) and (b<=latmax) then begin
 {If partition data, assign a symbol size.}
        if bins in [2..5] then begin
          if (t[3]>=bin0) and (t[3]<=bin1) then scale:=scale1;
          if (t[3]>bin1) and (t[3]<=bin2) then scale:=scale2;
          if ((bins>=3) and (t[3]>bin2) and (t[3]<=bin3))
              then scale:=scale3;
```

```
                if ((bins>=4) and (t[3]>bin3) and (t[3]<=bin4))
                    then scale:=scale4;
                if ((bins>=5) and (t[3]>bin4) and (t[3]<=bin5))
                    then scale:=scale5;
              end;
      {If size not 0, then display and write to output file.}
              if scale<>0 then begin
                DrawTextW(a,b,scale,Stg);
                Write(OutFile1,t);
              end;
    label0: end;
         until EOF(InFile1);
     end
     else begin
      {If symbols previously have been defined (not data values), then
       display.}
         Repeat
           Read(InFile1,t); if t[2]=0.0 then begin
             write(OutFile1,t); goto label1; end;
          a:=t[2]; b:=t[1]; c:=t[3]; Str(t[3],Stg);
      {Display if data within image limits.}
          if (a>=longmin) and (a<=longmax) and
             (b>=latmin) and (b<=latmax) then begin
      {If partition data, assign a symbol size.}
              if bins in [2..5] then begin
                if (t[3]>=bin0) and (t[3]<=bin1) then scale:=scale1;
                if (t[3]>bin1) and (t[3]<=bin2) then scale:=scale2;
                if ((bins>=3) and (t[3]>bin2) and (t[3]<=bin3))
                    then scale:=scale3;
                if ((bins>=4) and (t[3]>bin3) and (t[3]<=bin4))
                    then scale:=scale4;
                if ((bins>=5) and (t[3]>bin4) and (t[3]<=bin5))
                    then scale:=scale5;
              end;
      {If size not 0, then display and write to output file.}
              if scale<>0 then begin
                DrawTextW(a,b,scale,symbol);
                Write(OutFile1,t);
              end;
 label1:  end;
         until EOF(InFile1);
     end;
  if (scale1>scale) then scale0:=scale1 else scale0:=scale;
     {Indicates data may be plotted.}
  Dataflag:=True;
Close(InFile1); Close(OutFile1);
end;
{*************************************************************************}
Overlay Procedure StationPoint;
     {Procedure to display station locations on map.}
  begin
    ClearScreen; GotoXY(1,1);
```

```
Repeat
 {Ask for input filename and type.}
  WriteLn('Enter Input Station Filename (with extension) : ');
  ReadLn(TextInStatName);
    InStatName:=TextInStatName;
  Assign(InFile1,InStatName);
  {$I-} ReSet(InFile1) {$I+}; OK:=(IOresult=0);
  if not OK then WriteLn('Cannot find station file ',InStatName);
  Close(InFile1); WriteLn;
until OK;
 {As what type of file.  If ASCII, create corresponding binary file.}
 Repeat
  WriteLn('Is file in ASCII [A] or Pascal binary (.bin) [B] ?');
  Read(Kbd,FileType);
  if FileType in ['A','a'] then begin
    Delete(InStatName,(length(InStatName)-2),3);
    InStatName:=InStatName+'bin';
 {Convert ASCII to binary.}
    TextBi3(TextInStatName,InStatName);
    ClearScreen; GotoXY(1,1);
    WriteLn('The binary input file ',InStatName,' has been created');
    WriteLn('and will be used for input by HIMP.');
  end;
until FileType in ['A','a','B','b'];
WriteLn('Input Station File :',InStatName); WriteLn;
WriteLn('Enter Output Station Filename (no extension) [OutStn] : ');
WriteLn('(Binary File with Station Locations)');
ReadLn(OutStatName); if OutStatName='' then OutStatName:='OutStn';
  OutStatName:=OutStatName+'.bin';
WriteLn('Output Station File : ',OutStatName); WriteLn;
Assign(InFile1,InStatName); Reset(InFile1);
Assign(OutFile1,OutStatName); ReWrite(OutFile1);
HitRetToCon;
SymbolSelect;
PtScaleSelect; Scalestn:=scale;
MapWindow;
LoadWindow(1,-1,-1,OutMapName);
Read(InFile1,t); Read(InFile1,t);
t[1]:=latmin; t[2]:=latmax; Write(OutFile1,t);
t[1]:=longmin; t[2]:=longmax; Write(OutFile1,t);
if cflag = 2 then begin
  Repeat
    Str(StnNo,Stg);
    Read(InFile1,t);
    if (t[2]>=longmin) and (t[2]<=longmax) and
       (t[1]>=latmin) and (t[1]<=latmax) then begin
         DrawTextW(t[2],t[1],scalestn,Stg);
         Write(OutFile1,t);
       end;
    StnNo:=StnNo+1;
  until EOF(InFile1);
end else begin
```

```
      Repeat
        Read(InFile1,t);
        if (t[2]>=longmin) and (t[2]<=longmax) and
           (t[1]>=latmin) and (t[1]<=latmax) then begin
             DrawTextW(t[2],t[1],scalestn,symbol);
             Write(OutFile1,t);
           end;
      until EOF(InFile1);
    end;
      Close(InFile1); Close(OutFile1);
      Stationflag:=True;
     {Indicates station locations may be  plotted.}
  end;
{*********************************************************************}
begin
  Initiate; Banner;
     {Initiate program, set up files, display HIMP banner.}
  bin0:=0.0; bin1:=0.0; bin2:=0.0; bin3:=0.0; bin4:=0.0; bin5:=0.0;
  ClearScreen;
    DefineWindow(2,40,0,75,40);
    DefineWorld(2,0.0,50.0,50.0,0.0);
    MessageW;
    DrawTextW(2.0,5.0,1,'HIT RETURN TO COMMENCE');
    Rep;
    SetBackgroundColor(0);                {BLACK}
  Select;
     {Display options menu to initiate selected map display, set limits,
      and produce new map display.}
  MessageW;
     {Option for adding additional contour lines to map.}
    DrawTextW(2.,15.,1,'DISPLAY CONTOURS, FROM EXTERNAL ');
    DrawTextW(2.,25.,1,'FILES, ON MAP [Y] ?'); Read(Kbd,ch);
    if ch in ['Y','y'] then DepthContour; Close(OutFile);
  {HitRetToCon;
  MapWindow;}
  Stationflag:=False; Dataflag:=False;
     {Asks if user would like to display station locations or data or both.
If yes, flag(s) is(are) set to True.}
  MessageW;
    DrawTextW(2.,20.,1,'DISPLAY STATION LOCATIONS ON MAP [Y] ?');
    Read(Kbd,ch);
    if ch in ['y','Y'] then begin
     {Display station locations, select symbol and character corresponding
      string (for plotter).}
      StationPoint; stnsymbol:=symbol; stnsymbolstg:=symbolstg; stncflag:=cflag;
    end;
  MessageW;
    DrawTextW(2.,20.,1,'DISPLAY DATA ON MAP [Y] ?');
    Read(Kbd,ch);
    if ch in ['y','Y'] then begin
     {Display data, select symbol(s) and corresponding character string(s)
      (for plotter).}
```

```
      DataPoint; datasymbol:=symbol; datasymbolstg:=symbolstg;
                 datacflag:=cflag;
    end;
    Delay(1000); MessageW;
    HitRetToCon;
     {Initiate plotting sequence.}
  CreatePlot;
    WriteLn; WriteLn('FINIS');
    Delay(1000);
    LeaveGraphic;
end.


{*************************************************************************}
     {HIMPPLOT: Inclusion file; Plot subroutines and support utilities.}
{*************************************************************************}
Overlay Procedure StationPlot;
     {StationPlot: Subroutine used to plot station locations.}
{*************************************************************************}
var
     {Symbol width and height variables.}
  wsz,hsz                    :real;
begin
  WriteLn('FOR PLOT OF STATIONS :'); PenSelect;
  MessageW;
     {Option prompts for selecting new pen and/or plot symbol.}
    DrawTextW(2.0,10.0,1,'SELECTED SYMBOL IS :');
    DrawTextW(20.0,20.0,2,symbol);
    DrawTextW(2.0,40.0,1,'ACCEPT THIS PLOT SYMBOL [Y or CR] ?');
    Read(Kbd,ch);
      if not (ch in ['Y','y',chr(13)]) then SymbolSelect;
     {Plot symbol size = multiple of selected screen symbol size.}
    wsz:=0.2*scalestn; hsz:=0.28*scalestn;
    Str(wsz:5:3,wStg); Str(hsz:5:3,hStg);
    ClearScreen;
  MessageW;
     {Option to select new symbol size.}
    Stg:=wStg+' X '+hStg+' CM.';
    DrawTextW(2.0,10.0,1,'DEFAULT PLOT SYMBOL SIZE IS ');
    DrawTextW(7.0,20.0,1,Stg);
    DrawTextW(2.0,30.0,1,'KEEP THIS SIZE [Y]');
    DrawTextW(2.0,40.0,1,' OR SELECT NEW SIZE [N] ?');
    Read(Kbd,ch);
     {Symbol size slection procedure.}
      if ch in ['N','n'] then begin
        ClearScreen; GotoXY(1,1);
        WriteLn('Enter new symbol width (cm.) + Return :');
        Read(wStg); WriteLn;
        WriteLn('Enter new symbol height (cm.) + Return :');
        WriteLn('(Should be 1.4X symbol width.)');
        Read(hStg); WriteLn;
      end;
```

```
 {Send symbol and symbol size to plotter.}
Stg:='SI'+wStg+','+hStg+';'; Async_Send_String(Stg);
 {Open and reset input file.}
Assign(InFile1,OutStatName); ReSet(InFile1);
MessageW; DrawTextW(2.0,20.0,1,'READY TO PLOT STATION LOCATIONS');
DrawTextW(2.0,40.0,1,'HIT RETURN TO COMMENCE'); Rep;
 {Pass first two records.}
Read(InFile1,t); Read(InFile1,t); ClearScreen;
 {Ready pen for plotting.}
  Async_Send_String('PU;');
 {If symbol is standard defined, execute this procedure.  Locate plot
  position based on geographic position (t(1), t(2)), plot scaling (p2x
  - p1x) and (p2y - p1y), and map scaling (gxm and gym).  Convert to
  character string and send to plotter.  Repeat until EOF.}
  if cflag=0, then begin
    Async_Send_String(symbolstg);
    Repeat
      Read(InFile1,t);
      xi:=p1x-Round(Int(p2x-p1x)*((t[2]-longmax)/gxm));
      yi:=p1y+Round(Int(p2y-p1y)*((t[1]-latmin)/gym));
      Str(xi,sx1); Str(yi,sy1);
      Stg:='PA'+sx1+','+sy1+';';
      Async_Send_String(Stg);
    until EOF(InFile1);
  end;
 {If symbol is program defined, execute this procedure.  Symbol and
  size sent to plotter at every step.}
  if cflag=1 then begin
    Repeat
      Read(InFile1,t);
      xi:=p1x-Round(Int(p2x-p1x)*((t[2]-longmax)/gxm));
      yi:=p1y+Round(Int(p2y-p1y)*((t[1]-latmin)/gym));
      Val(wStg,wsz,err); Val(hStg,hsz,err);
      dxi:=xi-trunc(288.0*wsz); dyi:=yi-trunc(200.0*hsz);
      {1.4*400*wsz/2}
      Str(dxi,sx1); Str(dyi,sy1);
      Stg:='PA'+sx1+','+sy1+';'+symbolstg;
      Async_Send_String(Stg);
    until EOF(InFile1);
  end;
 {Execute this procedure if symbol is the station number.  Same
  procedures as those above, except numbers are incremented.}
  if cflag=2 then begin
    StnNo:=FirstStnNo;
    Repeat
      Str(StnNo,symbolstg); Str(((-length(symbolstg))div 2),stglength);
      Read(InFile1,t);
      xi:=p1x-Round(Int(p2x-p1x)*((t[2]-longmax)/gxm));
      yi:=p1y+Round(Int(p2y-p1y)*((t[1]-latmin)/gym));
      Str(xi,sx1); Str(yi,sy1);
      Stg:='PA'+sx1+','+sy1+';CP'+stglength+',-.25;LB'+
      symbolstg+chr(3)+';';
```

```
          Async_Send_String(Stg);
          StnNo:=StnNo+1;
        until EOF(InFile1);
      end;
     {Close input file and stop plotting stations.}
    Close(InFile1);
    Async_Send_String('SM;');
end;
{*************************************************************************}
Overlay Procedure DataPlot;
     {DataPlot: Subroutine used to plot data on map.}
{*************************************************************************}
label label0;
var
     {Height and width variables for five possible bin symbols.  Also,
      increment variable to increase symbol size for each bin, and offset
      to shift symbol when station location symbols have been previously
      plotted.}
  wsz,wsz1,wsz2,wsz3,wsz4,wsz5     :real;
  hsz,hsz1,hsz2,hsz3,hsz4,hsz5     :real;
  szar,deltasz                     :real;
  swsz,shsz                        :string[30];
  offset                           :integer;
     {Subprocedure to allow for a break in continuous data plots.  New line
      and pen may be selected at this break.}
  procedure plotjump;
     begin
        Repeat Read(OutFile1,t) until ((t[1]<>0.0) and (t[2]<>0.0));
        Async_Send_String('PU;');
        WriteLn('NEW DATA GROUP -- CHANGE LINE TYPE [Y] ?'); Read(Kbd,ch);
        if ch in ['Y','y'] then LineSelect; PenSelect;
    end;
     {Subprocedure to select new partitioning values.  May override those
      selected, in number and size, from screen development session.}
  procedure PartitionSelect;
     begin
        bin0:=0.0; bin1:=0.0; bin2:=0.0; bin3:=0.0; bin4:=0.0; bin5:=0.0;
        ClearScreen; GotoXY(1,1); bins:=1;
        Write('SELECT NUMBER OF DATA BINS (1,2,3,4, or 5) : '); ReadLn(bins);
        if bins=1 then exit;
        if bins>=2 then begin
          ClearScreen; GotoXY(1,1);
          Write('What is the minimum data value for bin 1 ?');
          ReadLn(bin0); WriteLn;
          Write('What is the maximum data value for bin 1 ?');
          ReadLn(bin1); WriteLn;
          Write('What is the maximum data value for bin 2 ?');
          ReadLn(bin2); WriteLn;
        end;
        if bins>=3 then begin
          Write('What is the maximum data value for bin 3 ?');
          ReadLn(bin3); WriteLn;
```

```
        end;
        if bins>=4 then begin
          Write('What is the maximum data value for bin 4 ?');
          ReadLn(bin4); WriteLn;
        end;
        if bins>=5 then begin
          Write('What is the maximum data value for bin 5 ?');
          ReadLn(bin5); WriteLn;
        end;
      end;
begin
     {Assign output data filename from screen development section to input
      plot file.  Selection options for pen and symbols offered here.}
  Assign(InFile1,OutDataName); ReSet(InFile1);
  WriteLn('FOR PLOT OF STATION DATA :'); PenSelect;
  MessageW;
  DrawTextW(2.0,10.0,1,'SELECTED SYMBOL IS :');
  DrawTextW(20.0,20.0,2,symbol);
  DrawTextW(2.0,40.0,1,'ACCEPT THIS PLOT SYMBOL [Y or CR] ?');
  Read(Kbd,ch);
  if not (ch in ['Y','y',chr(13)]) then DataSymbolSelect;
  if bins>1 then begin
      ClearScreen; GotoXY(1,1);
     {Prompt to see whether user would like to partition data.  If yes,
      partitioning procedure is called.}
      WriteLn('DATA IS CURRENTLY PARTITIONED INTO ',bins,' BINS.');
      WriteLn('KEEP CURRENT PARTITIONING OR SELECT NEW PARTITIONS [Y] ?');
      Read(Kbd,ch);
      if ch in ['Y','y'] then PartitionSelect;
    end;
    else begin
      ClearScreen; GotoXY(1,1);
      WriteLn('PARTITION DATA BEFORE PLOTTING [Y] ?');
      Read(Kbd,ch); if ch in ['Y','y'] then PartitionSelect;
    end;
     {Plot symbol sizes default to a multiple of screen symbol size.}
  wsz1:=0.2*scale0; hsz1:=0.288*scale0; deltasz:=0.08; szar:=1.44;
  Str(wsz1:5:3,wStg); Str(hsz1:5:3,hStg);
  Stg:=wStg+'W  X  '+hStg+'H  CM.';
  ClearScreen; GotoXY(1,1);
     {Option to change symbol sizes.}
  WriteLn('CURRENT DEFAULT PLOT SYMBOL SIZE IS ',Stg);
  if bins>1 then WriteLn('+ .04W X .058H CM. INCREMENT INCREASES.'); WriteLn;
  WriteLn('(Note : The 1.44 h/w ratio produces a plot aspect ratio of 1.)');
  WriteLn;
  WriteLn('KEEP DEFAULT, OR SELECT NEW SIZE PARAMETERS [Y] ?');
     {New symbol size(s) selection procedure.}
  Read(Kbd,ch); if ch in ['Y','y'] then begin
    ClearScreen; GotoXY(1,1);
    Write('ENTER NEW SYMBOL WIDTH (CM.) + RETURN : '); ReadLn(wsz1);
    WriteLn; WriteLn;
    Write('ENTER NEW SYMBOL HEIGHT (CM.) + RETURN : '); ReadLn(hsz1);
```

```
    WriteLn;
    if bins>1 then begin
      if wsz1<>0.0 then szar:=hsz1/wsz1 else szar:=1.4;
      WriteLn('ASPECT RATIO FOR SYMBOLS (H/W) IS ',szar);
      WriteLn('INCREMENTAL INCREASE WILL BE dW IN WIDTH');
      WriteLn('AND dH = (H/W)*dW IN HEIGHT.'); WriteLn;
      Write('ENTER INCREMENT dW (CM.) + RETURN : '); ReadLn(deltasz);
end;
  end;
  if cflag=2 then begin
     {Character offset (shift) option to prevent station and data symbols
      from conflicting.}
    offset:=0; WriteLn('Is a one character right offset desired [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then offset:=1;
  end;
     {Sizing of partitioned data symbols.}
  wsz2:=wsz1+deltasz; hsz2:=hsz1+szar*deltasz;
  wsz3:=wsz2+deltasz; hsz3:=hsz2+szar*deltasz;
  wsz4:=wsz3+deltasz; hsz4:=hsz3+szar*deltasz;
  wsz5:=wsz4+deltasz; hsz5:=hsz4+szar*deltasz;
  Async_Send_String('PU;');
     {For trackline (continuous) data.  Makes pen draw line continuously
      through data points, breaking when new data flags are encountered.}
  if aflag=2, then begin
    Assign(OutFile1,InDataName); Reset(OutFile1);
    ClearScreen; GotoXY(1,1); WriteLn('SELECT NEW LINETYPE [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LineSelect;
    Stg:='IW'+wx1+','+wy1+','+wx2+','+wy2+';'; Async_Send_String(Stg);
    Read(OutFile1,t); Read(OutFile1,t); Read(OutFile1,t);
    xi:=p1x-Round(Int(p2x-p1x)*((t[2]-longmax)/gxm));
    yi:=p1y+Round(Int(p2y-p1y)*((t[1]-latmin)/gym));
    Str(xi,sx1); Str(yi,sy1);
    Stg:='PU,PA'+sx1+','+sy1+';'; Async_Send_String(Stg);
     {Goes through data set with pen down.}
    Repeat
      Read(OutFile1,t);
      if (t[1]=0.0) and (t[2]=0.0) then Plotjump;
      xi:=p1x-Round(Int(p2x-p1x)*((t[2]-longmax)/gxm));
      yi:=p1y+Round(Int(p2y-p1y)*((t[1]-latmin)/gym));
      Str(xi,sx1); Str(yi,sy1);
      Stg:='PA'+sx1+','+sy1+';PD;'; Async_Send_String(Stg);
    until EOF(OutFile1);
    Async_Send_String('PU;IW;'); Close(OutFile1); Async_Send_String('LT;');
  end;
     {For longline (series of straight line segments) data.  Drops pen and
      then lifts pen for paired records.}
  if aflag=3 then begin
    Assign(OutFile1,InDataName); Reset(OutFile1);
    ClearScreen; GotoXY(1,1); WriteLn('SELECT NEW LINETYPE [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LineSelect;
    Stg:='IW'+wx1+','+wy1+','+wx2+','+wy2+';'; Async_Send_String(Stg);
    Read(OutFile1,t); Read(OutFile1,t);
```

```
  Repeat
    Read(OutFile1,t);
    if (t[1]=0.0) and (t[2]=0.0) then Plotjump;
    xi:=p1x-Round(Int(p2x-p1x)*((t[2]-longmax)/gxm));
    yi:=p1y+Round(Int(p2y-p1y)*((t[1]-latmin)/gym));
    Str(xi,sx1); Str(yi,sy1);
    if EOF(OutFile1) then begin
      sx2:=sx1; sy2:=sy1;
    end;
    else begin
      Read(OutFile1,t);
      xi:=p1x-Round(Int(p2x-p1x)*((t[2]-longmax)/gxm));
      yi:=p1y+Round(Int(p2y-p1y)*((t[1]-latmin)/gym));
      Str(xi,sx2); Str(yi,sy2);
    end;
    Stg:='PU,PA'+sx1+','+sy1+'PD,PA'+sx2+','+sy2+';';
    Async_Send_String(Stg);
  until EOF(OutFile1);
   {Closes file and then reopens and initiates file.  Plots data symbols
    or values.  Partitioned values plotted with corresponding symbol
    sizes.  This procedure follows data plots for longline or trackline
    data.  For station data, this procedure represents the only data
    plot.}
  Async_Send_String('PU;IW;'); Close(OutFile1); Async_Send_String('LT;');
end;
Read(InFile1,t); Read(InFile1,t);
   {For partitioned data.}
  if bins in [2..5] then begin
    Repeat
      Read(InFile1,t); wsz:=0.0;
   {Plot size determination for each data.}
      if (t[1]=0.0) and (t[2]=0.0) then goto label0;
           if (t[3]>=bin0) and (t[3]<=bin1) then begin
             if wsz1=0.0 then goto label0;
             wsz:=wsz1; hsz:=hsz1;
           end;
           if (t[3]>bin1) and (t[3]<=bin2) then begin
             wsz:=wsz2; hsz:=hsz2;
           end;
           if (t[3]>bin2) and (t[3]<=bin3) then begin
             wsz:=wsz3; hsz:=hsz3;
           end;
           if (t[3]>bin3) and (t[3]<=bin4) then begin
             wsz:=wsz4; hsz:=hsz4;
           end;
           if (t[3]>bin4) and (t[3]<=bin5) then begin
             wsz:=wsz5; hsz:=hsz5;
           end;
        if wsz=0.0 then goto label0;
        Str(wsz:6:3,swsz); Str(hsz:6:3,shsz);
        xi:=p1x-Round(Int(p2x-p1x)*((t[2]-longmax)/gxm));
        yi:=p1y+Round(Int(p2y-p1y)*((t[1]-latmin)/gym));
```

```
          Str(xi,sx1); Str(yi,sy1);
          if cflag=0 then begin
            Stg:='SI'+swsz+','+shsz+';'+symbolstg+'PA'+sx1+','+sy1+';';
            Async_Send_String(Stg);
          end;
      {Character plot procedure, depending on symbol selected.}
          if cflag=1 then begin
            dxi:=xi-trunc(288.0*wsz); dyi:=yi-trunc(200.0*hsz);
            Str(dxi,sx1); Str(dyi,sy1);
            Stg:='SI'+swsz+','+shsz+';'+'PA'+sx1+','+sy1+';'+symbolstg;
            Async_Send_String(Stg);
          end;
          if cflag=2 then begin
            Str(trunc(t[3]),stg);
            nn:=length(stg)+mm;
            Str(t[3]:nn:mm,symbolstg);
            if offset=1 then begin
              symbolstg:=' '+symbolstg;
              Stg:='SI'+swsz+','+shsz+';'+'PA'+sx1+','+sy1+';'+';LB'+
              symbolstg+chr(3)+';';
            end;
            else begin
              Str(((-length(symbolstg))div 2),stglength);
              Stg:='SI'+swsz+','+shsz+';'+'PA'+sx1+','+sy1+';'+';CP'+
              stglength+',-.25;LB'+symbolstg+chr(3)+';';
            end;
            Async_Send_String(Stg);
          end;
label0: until EOF(InFile1);
    end
     {For nonpartitioned data.  All symbols the same size.}
    else begin
      Str(wsz1:6:3,swsz); Str(hsz1:6:3,shsz);
      Stg:='SI'+swsz+','+shsz+';';
      Async_Send_String(Stg);
      Repeat
        Read(InFile1,t);
            xi:=p1x-Round(Int(p2x-p1x)*((t[2]-longmax)/gxm));
            yi:=p1y+Round(Int(p2y-p1y)*((t[1]-latmin)/gym));
            Str(xi,sx1); Str(yi,sy1);
            if cflag=0 then begin
              Stg:=symbolstg+'PA'+sx1+','+sy1+';';
              Async_Send_String(Stg);
            end;
            if cflag=1 then begin
              dxi:=xi-trunc(288.0*wsz1); dyi:=yi-trunc(200.0*hsz1);
              Str(dxi,sx1); Str(dyi,sy1);
              Stg:='PA'+sx1+','+sy1+';'+symbolstg;
              Async_Send_String(Stg);
            end;
```

```
            if cflag=2 then begin
              Str(trunc(t[3]),stg);
              nn:=length(stg)+mm;
              Str(t[3]:nn:mm,symbolstg);
              if offset=1 then begin
                symbolstg:=' '+symbolstg;
                Stg:='PA'+sx1+','+sy1+';'+';LB'+symbolstg+chr(3)+';';
              end;
              else begin
                Str(((-length(symbolstg))div 2),stglength);
                Stg:='SI'+swsz+','+shsz+';'+'PA'+sx1+','+sy1+';'+';CP'+
                stglength+',-.25;LB'+symbolstg+chr(3)+';';
              end;
              Async_Send_String(Stg);
            end;
      until EOF(InFile1);
    end;
  Close(InFile1);
  Async_Send_String('SM;');
end;
{*************************************************************************}
Procedure DrawXTick;
     {DrawXTick: Procedure is used to plot (draw) tick marks along x-
      boundaries.  Location and tick size supplied by tick drawing
      procedures listed below.}
begin
  xi:=p1x-Round(Int(p2x-p1x)*((longtmp-longmax)/gxm));
  Str(xi,sx1); Str(yi,sy1);
  Stg:='PA'+sx1+','+sy1+';XT;';
  Async_Send_String(Stg);
end;
{*************************************************************************}
Procedure DrawYTick;
     {DrawYTick: Procedure is used to plot (draw) tick marks along y-
      boundaries.  Location and tick size supplied by tick drawing
      procedures listed below.}
begin
  yi:=p1y+Round(Int(p2y-p1y)*((lattmp-latmin)/gym));
  Str(yi,sy1); Str(xi,sx1);
  Stg:='PA'+sx1+','+sy1+';YT;';
  Async_Send_String(Stg);
end;
{*************************************************************************}
Overlay Procedure LongDegTick;
     {Tick drawing procedure for longitude (x) degree ticks.  User prompted
      for plot options and then interval size.  DrawXTick called
      successively for each tick until complete longitude range is
      covered.}
```

```
{*************************************************************************}
begin
  ClearScreen; GotoXY(1,1);
  WriteLn('LONGITUDE TICKMARK OPTIONS : '); WriteLn;
  WriteLn('  (1) Degree tickmarks along bottom border. ');
  WriteLn('  (2) Degree tickmarks along top border. ');
  WriteLn('  (3) Degree tickmarks along top and bottom borders. ');
  WriteLn('  (4) Degree grid lines. ');
  WriteLn(' else - No tickmarks or grid lines. ');
  Read(Kbd,ch);
  Write('  Desired tickmark interval (no. degrees per tick) : ');
  ReadLn(interval); step:=Int(interval);
  lattmp:=Int(longmax/step)*step;
  if ch in ['1','3'] then begin
    yi:=p1y; longtmp:=lattmp; Async_Send_String('TL2,0;');
    while longtmp>=longmin do
      begin
        DrawXTick;
        longtmp:=longtmp-step;
      end;
    end;
  if ch in ['2','3'] then begin
    yi:=p2y; longtmp:=lattmp; Async_Send_String('TL0,2;');
    while longtmp>=longmin do
      begin
        DrawXTick;
        longtmp:=longtmp-step;
      end;
    end;
  if ch = '4' then begin
     {Grid line option.  User may select line type.}
    yi:=p1y; longtmp:=lattmp;
    ClearScreen; GotoXY(1,1); WriteLn('SELECT NEW LINETYPE [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LineSelect;
    while longtmp>=longmin do
      begin
        xi:=p1x-Round(Int(p2x-p1x)*((longtmp-longmax)/gxm)); Str(xi,sx1);
        Stg:='PA'+sx1+','+wy1+';PD;PA'+sx1+','+wy2+';PU;';
        Async_Send_String(Stg);
        longtmp:=longtmp-step;
      end; Async_Send_String('LT;');
    end;
end;
{*************************************************************************}
Overlay Procedure LatDegTick;
begin
     {Tick drawing procedure for latitude (y) degree ticks.  Prompts for
      options and interval size.  Uses DrawYTick in successive calls.}
  ClearScreen; GotoXY(1,1);
  WriteLn('LATITUDE TICKMARK OPTIONS : '); WriteLn;
  WriteLn('  (1) Degree tickmarks along left border. ');
```

```
  WriteLn('  (2) Degree tickmarks along right border. ');
  WriteLn('  (3) Degree tickmarks along left and right borders. ');
  WriteLn('  (4) Degree grid lines. ');
  WriteLn(' else - No tickmarks or grid lines. ');
  Read(Kbd,ch);
  Write('  Desired tickmark interval (no. degrees per tick) : ');
  ReadLn(interval); step:=Int(interval);
  longtmp:=Int(latmax/step)*step;
  if ch in ['1','3'] then begin
    xi:=p1x; lattmp:=longtmp; Async_Send_String('TL2,0;');
    while lattmp>=latmin do
      begin
        DrawYTick;
        lattmp:=lattmp-step;
      end;
    end;
  if ch in ['2','3'] then begin
    xi:=p2x; lattmp:=longtmp; Async_Send_String('TL0,2;');
    while lattmp>=latmin do
      begin
        DrawYTick;
        lattmp:=lattmp-step;
      end;
    end;
  if ch = '4' then begin
     {Grid line option.  User may select line type.}
    xi:=p1x; lattmp:=longtmp;
    ClearScreen; GotoXY(1,1); WriteLn('SELECT NEW LINETYPE [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LineSelect;
    while lattmp>=latmin do
      begin
        yi:=p1y-Round(Int(p2y-p1y)*((lattmp-latmax)/gym)); Str(yi,sy1);
        Stg:='PA'+wx1+','+sy1+';PD;PA'+wx2+','+sy1+';PU;';
        Async_Send_String(Stg);
        lattmp:=lattmp-step;
      end; Async_Send_String('LT;');
    end;
end;
{*************************************************************************}
Overlay Procedure LongMinTick;
     {Longitude minute tick drawing procedure.  Prompts for options and
      interval.  Uses DrawXTick.}
begin
  ClearScreen; GotoXY(1,1); interval:=60;
  WriteLn('LONGITUDE TICKMARK OPTIONS : '); WriteLn;
  WriteLn('  (1) Minute tickmarks along bottom border. ');
  WriteLn('  (2) Minute tickmarks along top border. ');
  WriteLn('  (3) Minute tickmarks along top and bottom borders. ');
  WriteLn('  (4) Minute grid lines. ');
  WriteLn(' else - No tickmarks or grid lines. ');
  Read(Kbd,ch); WriteLn;
  Write('  Desired tickmark interval (no. min. per tick) : ');
```

```
  ReadLn(interval); step:=Int(interval);
  lattmp:=Int(longmax)+(Int(60.0*Frac(longmax)/step))*step/60.0;
  if ch in ['1','3'] then begin
    yi:=p1y; longtmp:=lattmp; Async_Send_String('TL1,0;');
    while longtmp>=longmin do
      begin
        DrawXTick;
        longtmp:=longtmp-step/60.0;
      end;
    end;
  if ch in ['2','3'] then begin
    yi:=p2y; longtmp:=lattmp; Async_Send_String('TL0,1;');
    while longtmp>=longmin do
      begin
        DrawXTick;
        longtmp:=longtmp-step/60.0;
      end;
    end;
  if ch = '4' then begin
     {Grid option.}
    yi:=p1y; longtmp:=lattmp;
    ClearScreen; GotoXY(1,1); WriteLn('SELECT NEW LINETYPE [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LineSelect;
    while longtmp>=longmin do
      begin
        xi:=p1x-Round(Int(p2x-p1x)*((longtmp-longmax)/gxm)); Str(xi,sx1);
        Stg:='PA'+sx1+','+wy1+';PD;PA'+sx1+','+wy2+';PU;';
        Async_Send_String(Stg);
        longtmp:=longtmp-step/60.0;
      end; Async_Send_String('LT;');
    end;
end;
{*************************************************************************}
Overlay Procedure LatMinTick;
     {Latitude minute tick drawing procedure.  Prompts for options and
      interval.  Uses DrawYTick.}
begin
  ClearScreen; GotoXY(1,1); interval:=60;
  WriteLn('LATITUDE TICKMARK OPTIONS : '); WriteLn;
  WriteLn('  (1) Minute tickmarks along left border. ');
  WriteLn('  (2) Minute tickmarks along right border. ');
  WriteLn('  (3) Minute tickmarks along left and right borders. ');
  WriteLn('  (4) Minute grid lines. ');
  WriteLn(' else - No tickmarks or grid lines. ');
  Read(Kbd,ch); WriteLn;
  Write('  Desired tickmark interval (no. min. per tick) : ');
  ReadLn(interval); step:=Int(interval);
  longtmp:=Int(latmax)+(Int(60.0*Frac(latmax)/step))*step/60.0;
```

```
  if ch in ['1','3'] then begin
    xi:=p1x; lattmp:=longtmp; Async_Send_String('TL1,0;');
    while lattmp>=latmin do
      begin
        DrawYTick;
        lattmp:=lattmp-step/60.0;
      end;
    end;
  if ch in ['2','3'] then begin
    xi:=p2x; lattmp:=longtmp; Async_Send_String('TL0,1;');
    while lattmp>=latmin do
      begin
        DrawYTick;
        lattmp:=lattmp-step/60.0;
      end;
    end;
  if ch = '4' then begin
     {Grid option.}
    xi:=p1x; lattmp:=longtmp;
    ClearScreen; GotoXY(1,1); WriteLn('SELECT NEW LINETYPE [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LineSelect;
    while lattmp>=latmin do
      begin
        yi:=p1y-Round(Int(p2y-p1y)*((lattmp-latmax)/gym)); Str(yi,sy1);
        Stg:='PA'+wx1+','+sy1+';PD;PA'+wx2+','+sy1+';PU;';
        Async_Send_String(Stg);
        lattmp:=lattmp-step/60.0;
      end; Async_Send_String('LT;');
    end;
end;
{*************************************************************************}
Overlay Procedure LongSecTick;
     {Longitude second tick drawing procedure.  Prompts for options and
      interval.  Uses DrawXTick.}
begin
  ClearScreen; GotoXY(1,1); interval:=60;
  WriteLn('LONGITUDE TICKMARK OPTIONS : '); WriteLn;
  WriteLn('  (1) Second tickmarks along bottom border. ');
  WriteLn('  (2) Second tickmarks along top border. ');
  WriteLn('  (3) Second tickmarks along top and bottom borders. ');
  WriteLn('  (4) Second grid lines. ');
  WriteLn(' else - No tickmarks or grid lines. ');
  Read(Kbd,ch); WriteLn;
  Write('  Desired tickmark interval (no. sec. per tick) : ');
  ReadLn(interval); step:=Int(interval);
  lattmp:=Int(longmax)+(Int(3600.0*Frac(longmax)/step))*step/3600.0;
  if ch in ['1','3'] then begin
    yi:=p1y; longtmp:=lattmp; Async_Send_String('TL.5,0;');
    while longtmp>=longmin do
```

```
      begin
        DrawXTick;
        longtmp:=longtmp-step/3600.0;
      end;
    end;
  if ch in ['2','3'] then begin
    yi:=p2y; longtmp:=lattmp; Async_Send_String('TL0,.5;');
    while longtmp>=longmin do
      begin
        DrawXTick;
        longtmp:=longtmp-step/3600.0;
      end;
    end;
  if ch = '4' then begin
     {Grid option.}
    yi:=p1y; longtmp:=lattmp;
    ClearScreen; GotoXY(1,1); WriteLn('SELECT NEW LINETYPE [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LineSelect;
    while longtmp>=longmin do
      begin
        xi:=p1x-Round(Int(p2x-p1x)*((longtmp-longmax)/gxm)); Str(xi,sx1);
        Stg:='PA'+sx1+','+wy1+';PD;PA'+sx1+','+wy2+';PU;';
        Async_Send_String(Stg);
        longtmp:=longtmp-step/3600.0;
      end; Async_Send_String('LT;');
    end;
end;
{*************************************************************************}
Overlay Procedure LatSecTick;
     {Latitude second tick drawing procedure.  Prompts for options and
      interval.  Uses DrawYTick.}
begin
  ClearScreen; GotoXY(1,1); interval:=60;
  WriteLn('LATITUDE TICKMARK OPTIONS : '); WriteLn;
  WriteLn('  (1) Second tickmarks along left border. ');
  WriteLn('  (2) Second tickmarks along right border. ');
  WriteLn('  (3) Second tickmarks along left and right borders. ');
  WriteLn('  (4) Second grid lines. ');
  WriteLn(' else - No tickmarks or grid lines. ');
  Read(Kbd,ch); WriteLn;
  Write('  Desired tickmark interval (no. sec. per tick) : ');
  ReadLn(interval); step:=Int(interval);
  longtmp:=Int(latmax)+(Int(3600.0*Frac(latmax)/step))*step/3600.0;
  if ch in ['1','3'] then begin
    xi:=p1x; lattmp:=longtmp; Async_Send_String('TL.5,0;');
    while lattmp>=latmin do
      begin
        DrawYTick;
        lattmp:=lattmp-step/3600.0;
      end;
    end;
```

```
  if ch in ['2','3'] then begin
    xi:=p2x; lattmp:=longtmp; Async_Send_String('TL0,.5;');
    while lattmp>=latmin do
      begin
        DrawYTick;
        lattmp:=lattmp-step/3600.0;
      end;
    end;
  if ch = '4' then begin
    xi:=p1x; lattmp:=longtmp;
    ClearScreen; GotoXY(1,1); WriteLn('SELECT NEW LINETYPE [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LineSelect;
    while lattmp>=latmin do
      begin
        yi:=p1y-Round(Int(p2y-p1y)*((lattmp-latmax)/gym)); Str(yi,sy1);
        Stg:='PA'+wx1+','+sy1+';PD;PA'+wx2+','+sy1+';PU;';
        Async_Send_String(Stg);
        lattmp:=lattmp-step/3600.0;
      end; Async_Send_String('LT;');
    end;
end;
{*************************************************************************}
Overlay Procedure Cornerlabel;
     {Procedure to label any or all of the four plot corners.  Conversion
      is made from absolute longitude (0° to 360°) to normal longitude (0°
      to 180°; 0° to -180°).}
var
  cx,cy                                  :integer;
begin
  Async_Send_String('SP1;SI.14,.2;');
  Repeat ClearScreen; GotoXY(1,1);
    WriteLn('(1) LABEL UPPER LEFT CORNER'); WriteLn('(2) LABEL LOWER LEFT
    CORNER');
    WriteLn('(3) LABEL LOWER RIGHT CORNER'); WriteLn('(4) LABEL UPPER RIGHT
    CORNER');
    WriteLn('ELSE CONTINUE'); Read(Kbd,ch);
    if ch='1' then begin
      Coordlabel(longmax); cy:=p2y+100; Str(cy,sy2);
      Stg:='PA'+wx1+','+sy2+';LB'+Stg+chr(3)+';';
      Async_Send_String(Stg);
      Coordlabel(latmax); cx:=p1x-100; Str(cx,sx1);
      cy:=p2y-length(Stg)*65; Str(cy,sy2);
      Stg:='PA'+sx1+','+sy2+';DI0,1;LB'+Stg+chr(3)+';DI;';
      Async_Send_String(Stg); end;
    if ch='2' then begin
      Coordlabel(longmax); cy:=p1y-200; Str(cy,sy1);
      Stg:='PA'+wx1+','+sy1+';LB'+Stg+chr(3)+';';
      Async_Send_String(Stg);
      Coordlabel(latmin); cx:=p1x-100; Str(cx,sx1);

      Stg:='PA'+sx1+','+wy1+';DI0,1;LB'+Stg+chr(3)+';DI;';
      Async_Send_String(Stg); end;
```

```
    if ch='3' then begin
      Coordlabel(longmin); cy:=p1y-200; Str(cy,sy1);
      cx:=p2x-length(Stg)*65; Str(cx,sx2);
      Stg:='PA'+sx2+','+sy1+';LB'+Stg+chr(3)+';';
      Async_Send_String(Stg);
      Coordlabel(latmin); cx:=p2x+200; Str(cx,sx2);
      Stg:='PA'+sx2+','+wy1+';DIO,1;LB'+Stg+chr(3)+';DI;';
      Async_Send_String(Stg); end;
    if ch='4' then begin
      Coordlabel(longmin); cy:=p2y+100; Str(cy,sy2);
      cx:=p2x-length(Stg)*65; Str(cx,sx2);
      Stg:='PA'+sx2+','+sy2+';LB'+Stg+chr(3)+';';
      Async_Send_String(Stg);
      Coordlabel(latmax); cx:=p2x+200; Str(cx,sx2);
      cy:=p2y-length(Stg)*65; Str(cy,sy2);
      Stg:='PA'+sx2+','+sy2+';DIO,1;LB'+Stg+chr(3)+';DI;';
      Async_Send_String(Stg); end;
  until not (ch in ['1'..'4']); end;
{*********************************************************************}
Overlay Procedure LabelPlot;
     {Procedure to label plot.  User may repeat the procedure indefinitely,
      selecting absolute plot locations and character sizes.  Pen selection
      also supported.  Default location is for the center of the label to
      be positioned midway, left to right and 1/10 from top of the maximum
      HP plotting figure area.  This is the area just inside the paper
      boundary on standard 8 1/2" X 11" paper.  On character size, a
      zero (0) will cause the symbol to not be plotted.  Default character
      size is 187 cm X 269 cm.  Default vertical spacing between lines is
      0.269 cm.}
var
  lbx,lby,lblines,ilines            :integer;
  lbszw,lbszh,xoff,ydrop            :real;
  lbxf,lbyf,dummy,drop              :real;
  lbstg                             :String[100];
  lbszwstg,lbszhstg                 :String[10];
  lbxstg,lbystg,dropstg             :String[10];
begin
  Repeat ClearScreen; GotoXY(1,1);
  lblines:=1; lbszw:=0.187; lbszh:=0.269; dummy:=0.0;
  ydrop:=1.0; xoff:=0.0; lbxf:=0.5; lbyf:=0.9; drop:=0.0;
  WriteLn('FOR LABEL :'); PenSelect;
  ClearScreen; GotoXY(1,1);
  WriteLn('Number of label lines : ');
  Read(lblines); ClearScreen; GotoXY(1,1);
  WriteLn('Default character size is .187w X .269h (cm).'); WriteLn;
  WriteLn('Accept default, or select new size parameters [Y] ?'); WriteLn;
  Read(Kbd,ch);
  if ch in ['Y','y'] then begin
    WriteLn('Enter new character width in cm. + Return : ');
    Read(lbszw); WriteLn;
    WriteLn('Enter new character height in cm. + Return : '); Read(lbszh);
```

```
    end;
    ClearScreen; GotoXY(1,1);
    WriteLn('Center of first label line located midway between left and
    right ');
    WriteLn('paper boundaries, and 1/10 down from top boundary.'); WriteLn;
    WriteLn('Accept current location, or select new location [Y] ?');
    Read(Kbd,ch);
    if ch in ['Y','y'] then begin
      WriteLn('Enter fractional distance from left to right boundaries ');
      WriteLn('(0.0 - 1.0) [ Return to keep default] : ');
      Read(dummy); if dummy<>0.0 then lbxf:=dummy; dummy:=0.0; WriteLn;
      WriteLn('Enter Fractional distance from bottom to top boundaries ');
      WriteLn('(0.0 - 1.0) [ Return to keep default] : ');
      Read(dummy); if dummy<>0.0 then lbyf:=dummy; dummy:=0.0; WriteLn;
      WriteLn('New lateral location : ',lbxf:4:2,' from left to right
      boundaries');
      WriteLn('New vertical location : ',lbyf:4:2,' from bottom to top
      boundaries');
    end;
    if lblines>1 then begin
      ClearScreen; GotoXY(1,1);
      WriteLn('Default vertical drop for successive lines is 0.269 cm.');
      WriteLn;
      WriteLn('Keep default, or select new drop [Y] ?');
      Read(kbd,ch);
      if ch in ['Y','y'] then begin
        ClearScreen; GotoXY(1,1);
        WriteLn('Enter new vertical drop + Return : '); Read(ydrop);
      end;
    end;
  ClearScreen; GotoXY(1,1); Str(lbszw:10:4,lbszwstg);
  Str(lbszh:10:4,lbszhstg);
  Stg:='SI'+lbszwstg+','+lbszhstg+';'; Async_Send_String(Stg);
  lbx:=Round((p2x+p1x)*lbxf); lby:=Round((p2y+p1y)*lbyf);
  for ilines := 1 to lblines do begin
      WriteLn('Enter text for line ',ilines,' :'); WriteLn; Read(lbstg);
      Str(((-length(lbstg)) div 2),stglength);
      Str(lbx,lbxstg); Str(lby,lbystg); Str(drop,dropstg);
      Stg:='PA'+lbxstg+','+lbystg+';CP'+stglength+','+dropstg+';
      LB'+lbstg+chr(3)+';';
      Async_Send_String(Stg); drop:=drop-ydrop; WriteLn; WriteLn;
    end;
    WriteLn('WRITE ADDITIONAL LABEL [Y/N] ?'); Read(Kbd,ch);
    until not (ch in ['Y','y']);
end;
{*************************************************************************}
Overlay Procedure HPPlot;
     {HPPlot: Map and data plotting routine.}
const
     {HP plotting aspect ratio.}
  hpar                                    : real = 0.76816;
```

```
var
     {PontStatus: Pont activation variable.  T = open; F = close.  Factor:
      Percent of maximum HPPlot area to be used.  (Factor = 0.5 = half
      size.)}
  PortStatus                                  : Boolean;
  factor                                      : real;
begin
     {Initiate plotter, set up plot area, determine geographic/plot
      limits, based on scale factor determined by user.  Then draw four
      dots to indicate maximum HPPlot limits, and then map frame.}
  ClearScreen;
  Async_Init;
  PortStatus:=Async_Open(1,9600,n,8,1);
  WriteLn('Port Status =',PortStatus); WriteLn;
  if Iflag=0 then
    Assign(InFile,OutFileName)
    else Assign(InFile,InFileName);
  ReSet(InFile);
  Read(InFile,s); latmin:=s[1]; latmax:=s[2];
  Read(InFile,s); longmin:=s[1]; longmax:=s[2];
  gxm:=longmax-longmin; gym:=latmax-latmin;
  gar:=gym/gxm;
    if gar>=hpar
      then begin
        dpy:=7962; dpx:=Round(10365.0*hpar/gar);
      end
      else begin
        dpy:=Round(7962.0*gar/hpar); dpx:=10365;
      end;
  factor:=0.9; WriteLn('Plot factor is set at : ',factor:5:2);
  WriteLn('Proceed with this value, or select new value [Y] ?');
  Read(Kbd,ch);
  if ch in ['Y','y'] then begin
    WriteLn('Enter new plot factor (0.0< <=1.0) : '); Read(factor); end;
  dpx:=Trunc(dpx*factor); dpy:=Trunc(dpy*factor);
  p1x:=5182-(dpx div 2); p2x:=5182+(dpx div 2);
  p1y:=3981-(dpy div 2); p2y:=3981+(dpy div 2);
  Str(p1x,sx1); Str(p2x,sx2); Str(p1y,sy1); Str(p2y,sy2);
  wx1:=sx1; wx2:=sx2; wy1:=sy1; wy2:=sy2;
  Stg:='IN;IP'+sx1+','+sy1+','+sx2+','+sy2+';';
  Async_Send_String(Stg);
  Async_Send_String('SP1;PA0,0;PD;PU;PA10365,0;PD;PU;PA10365,7962;PD;PU;
  PA0,7962;PD;PU;');
  Stg:='SP 1;PA'+sx1+','+sy1+';';
  Async_Send_String(Stg);
  Stg:='PD;PA'+sx2+','+sy1+','+sx2+','+sy2+','+sx1+','+sy2+','+sx1+',
  '+sy1+';PU;';
  Async_Send_String(Stg);
     {Prompts user if ticks are desired and draws ticks.}
     {Prompts are issued only if geographic area dictates ticks are
      possible.}
```

```
ClearScreen; GotoXY(1,1);
  if (gxm>=1.0) or (Int(longmax)>=longmin) then begin
    WriteLn('Draw Longitude Degree Tickmarks [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LongDegTick;
    end;
ClearScreen; GotoXY(1,1);
  if (gym>=1.0) or (Int(latmax)>=latmin) then begin
    WriteLn('Draw Latitude Degree Tickmarks [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LatDegTick;
    end;
ClearScreen; GotoXY(1,1);
  if (gxm>=(1.0/60.0)) or (Int(60.0*Frac(longmax))>=(60.0*Frac(longmin)))
    then begin WriteLn('Draw Longitude Minute Tickmarks [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LongMinTick;
    end;
ClearScreen; GotoXY(1,1);
  if (gym>=(1.0/60.0)) or (Int(60.0*Frac(latmax))>=(60.0*Frac(latmin)))
  then begin
    WriteLn('Draw Latitude Minute Tickmarks [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LatMinTick;
    end;
ClearScreen; GotoXY(1,1);
  if (gxm>=(1.0/3600.0)) or (Int(60.0*Frac(60.0*Frac(longmax)))>=
    (60.0*Frac(60.0*Frac(longmin)))) then begin
    WriteLn('Draw Longitude Second Tickmarks [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LongSecTick;
    end;
ClearScreen; GotoXY(1,1);
  if (gym>=(1.0/3600.0)) or (Int(60.0*Frac(60.0*Frac(latmax)))>=
    (60.0*Frac(60.0*Frac(latmin)))) then begin
    WriteLn('Draw Latitude Second Tickmarks [Y] ?');
    Read(Kbd,ch); if ch in ['Y','y'] then LatSecTick;
    end;
   {Prompts user for new pen or line type, opens map file, and draws map.
    Stops for new contour lines to prompt for a change in pen or line
    type.}
PenSelect;
ClearScreen; GotoXY(1,1); WriteLn('SELECT NEW LINETYPE [Y] ?');
  Read(Kbd,ch); if ch in ['Y','y'] then LineSelect;
  Read(InFile,s);
ai:=0; bi:=0;
  Repeat
    Read(InFile,s);
      if (s[1]<=0) and (s[2]<=0) then begin
        ai:=0; bi:=0;
        if (s[1]<=-1.0) and (s[2]<=-1.0) then begin
          ClearScreen; GotoXY(1,1);
          WriteLn('FOR NEW CONTOUR LINE :'); PenSelect;
          WriteLn('SELECT NEW LINETYPE FOR CONTOUR [Y] ?');
          Read(Kbd,ch); if ch in ['Y','y'] then LineSelect;
        end;
      end;
```

```
          else begin
            xi:=p1x-Round(Int(p2x-p1x)*((s[2]-longmax)/gxm));
            yi:=p1y+Round(Int(p2y-p1y)*((s[1]-latmin)/gym));
            Str(xi,sx1); Str(yi,sy1);
          if ai=0 then begin
            Stg:='PU;PA'+sx1+','+sy1+';PD;';
            Async_Send_String(Stg);
          end;
          else begin
            Stg:='PA'+sx1+','+sy1+';';
            Async_Send_String(Stg);
          end;
          ai:=xi; bi:=yi;
          end;
      until EOF(InFile);
    Close(InFile); Async_Send_String('PU;');
       {Plot station location symbols, if desired.  If not, station location
        file (Stationflag = 1) was created; this prompt is not used.}
    if Stationflag then begin
      ClearScreen; GotoXY(1,1);
      WriteLn('Plot station locations [Y] ?'); Read(Kbd,ch);
        if ch in ['Y','y'] then begin
          symbol:=stnsymbol; symbolstg:=stnsymbolstg; cflag:=stncflag;
          StationPlot;
        end;
    end;
       {If data file created (Dataflag = 1), then prompts user if data
        plot is desired.}
    if Dataflag then begin
      ClearScreen; GotoXY(1,1);
      WriteLn('Plot data values [Y] ?'); Read(Kbd,ch);
        if ch in ['Y','y'] then begin
          symbol:=datasymbol; symbolstg:=datasymbolstg; cflag:=datacflag;
          DataPlot;
        end;
    end;
    Clear Screen; GotoXY(1,1);
       {Coordinate labels?}
    WriteLn('Label corner coordinates [Y] ?'); Read(Kbd,ch);
      if ch in ['Y','y'] then Cornerlabel;
    ClearScreen; GotoXY(1,1);
       {Label plot?}
    WriteLn('Label Plot [Y] ?'); Read(Kbd,ch);
      if ch in ['Y','y'] then LabelPlot;
    Async_Send_String('PU;SP;');
    Async_Close;
  end;
```

```
{*************************************************************************}
Overlay Procedure CreateASCIIFile;
     {Procedure to create post-processing file.  (Not supported at this
      time.)}
var
  PlotFile        :text;
  pos             :integer;
begin
  WriteLn('CREATING DATA FILE FOR HP PLOTTER ');
    Assign(InFile,OutFileName); Reset(InFile);
    pos:=length(OutFileName)-3;
    Delete(OutFileName,pos,4);
    OutFileName:=OutFileName+'.dat';
    WriteLn('Data File Name [',OutFileName,'] : ?');
    ReadLn(InFileName);
    if not (InFileName='') then
      OutFileName:=InFileName;
    Assign(PlotFile,OutFileName); ReWrite(PlotFile);
    while not EOF(InFile) do begin
      Read(InFile,s);
      WriteLn(PlotFile,s[1],s[2]);
      end;
    Close(InFile); Close(PlotFile);
    WriteLn('HP PLOT FILE (FOR BASIC DRIVER) HAS BEEN CREATED
                                UNDER THE NAME:
    ',OutFileName);
end;
{*************************************************************************}
 procedure CreatePlot;
      {Procedure to query user if a hard copy plot is desired.}
label label0;
var
  PlotFile                        : text;
  pos                             : integer;
begin
  ClearScreen; GoToXY(1,1);
label0:  WriteLn('SELECT PLOT OPTION :'); WriteLn;
  WriteLn('    (1) Plot Output Map ----- Plotter must be physically attached
  and ready.');
  WriteLn('    (2) Create Output Plot File ----- For transport to plotter
  (Pascal Driver) ');
  WriteLn('    (3) Exit HIMP');
  Read(Kbd,ch);
    Case ord(ch) of
            49 : begin
                   Repeat
                     HPPlot; ClearScreen; GotoXY(1,1);
                     WriteLn('Make a new plot [Y], (else exit) ?');
                     Read(Kbd,ch);
                 until not (ch in ['Y','y']);
```

```
                end;
            50 : WriteLn('HP PLOT FILE (FOR PASCAL DRIVER) HAS BEEN CREATED
                 UNDER THE NAME : ',OutFileName);
            51 : Exit;
          else begin
                ClearScreen; GotoXY(1,1);
                WriteLn('AVAILABLE OPTIONS ARE (1), (2), OR (3).');
                Goto label0;
          end;
   end;
end;


{*************************************************************************}
     {HIMPUTIL: Inclusion file containing miscellaneous support
      utilities for HIMP.}
procedure Rep;
     {Rep waits for the user to send a carriage return before allowing HIMP
      to proceed.}
  begin
    Repeat;
    Read(Kbd,ch);
    Until ch=#13;
  end;
{*************************************************************************}
procedure Initiate;
     {This procedure initiates hi.res. graphics, sets the screen colors,
      and clears the screen.}
  begin
    InitGraphic;
    HiRes;
    ClearScreen;
    SetForeGroundColor(2);                    {GREEN}
    SetBackGroundColor(0);                    {BLACK}
  end;
{*************************************************************************}
procedure MessageW;
     {MessageW brings up the comm window in the upper right corner of the
      screen.}
  begin
    SelectWorld(2);
    SelectWindow(2);
    SetBackground(0);
    Drawborder;
  end;
{*************************************************************************}
Overlay Procedure TextBi2(var TextName, Bi2Name :WorkString);
var
     {TextBi2 converts ASCII files containing two-entry records to binary
      files.}
```

```
  TextInFile          :text;
  BiOutFile           :file of sarray;
begin
  assign(TextInFile,TextName); Reset(TextInFile);
  assign(BiOutFile,Bi2Name); ReWrite(BiOutFile);
  Repeat  ReadLn(TextInFile,s[1],s[2]); Write(BiOutFile,s);
    until EOF(TextInFile);
  close(TextInFile); Close(BiOutFile);
end;
{*************************************************************************}
Overlay Procedure TextBi3(var TextName, Bi3Name :WorkString);
var
     {TextBi3 converts ASCII files containing three-entry records to binary
      files.}
  TextInFile          :text;
  BiOutFile1          :file of tarray;
begin
  assign(TextInFile,TextName); Reset(TextInFile);
  assign(BiOutFile1,Bi3Name); ReWrite(BiOutFile1);
  Repeat  ReadLn(TextInFile,t[1],t[2],t[3]); Write(BiOutFile1,t);
    until EOF(TextInFile);
  close(TextInFile); Close(BiOutFile1);
end;
{*************************************************************************}
Overlay Procedure PenSelect;
     {PenSelect allows the user to select a new pen for the Hewlett-Packard
      plotter.}
begin
  ClearScreen; GotoXY(1,2);
    WriteLn('SELECT NEW PEN NUMBER ?');
    WriteLn('(No. : 1-6.  Else keep current pen.)');
    Read(Kbd,ch);
      if ch in ['1'..'6'] then begin
        { pen:=ch; Stg:='SP'+pen+';';} Stg:='SP'+ch+';';
         WriteLn('New pen : ',ch);
         Async_Send_String(Stg);
       end;
end;
{*************************************************************************}
Overlay Procedure LineSelect;
begin
     {Line select allows the user to change line type during the plotting
      of a contour or track.}
  ClearScreen; GotoXY(1,1); Async_Send_String('LT;');
  WriteLn('SELECT NEW LINE TYPE :'); WriteLn('(1)  .        .        .');
  WriteLn('(2)  ____    ____    ____');
  WriteLn('(3)  _____   _____   _____ ');
  WriteLn('(4)  _______._______._______.');
  WriteLn('(5)  _____ _ _____ _ _____ _');
  WriteLn('(6)  ____ _ _ ____ _ _ ____ _ _ '); WriteLn('Else  ____________');
```

```
  Read(Kbd,ch); if ch in ['1'..'6'] then begin
    Stg:='LT'+ch+';'; WriteLn('New linetype :',ch);
    Async_Send_String(Stg); end;
end;
{*********************************************************************}
Overlay Procedure LineStyle;
begin
     {LineStyle allows the user to select a line type for the screen
      display of a contour or track.}
  Repeat
    Clearscreen; GotoXY(1,1); WriteLn('SELECT LINE STYLE :');
    WriteLn('(0)  **************** ');
    WriteLn('(1)  *   *   *   *   ');
    WriteLn('(2)  *****   ***** ');
    WriteLn('(3)  ***  *  ***  *');

    WriteLn('(4)  *** *** *** ***'); ReadLn(LS); SetLineStyle(LS);
  until LS in [0..4];
end;
{*********************************************************************}
Overlay Procedure Coordlabel(var coordtmp:real);
var
     {Coordlabel converts map limits, longitudes, and latitudes from
      degrees and fractions of degrees to degrees, minutes, and seconds.
      The results are converted to ASCII symbols in a form suitable for
      transmission to the Hewlett-Packard plotter.}
    degtmp,mintmp,sectmp      :integer;
    sdegtmp,smintmp,ssectmp   :string[30];
    rmin,rsec                 :real;
    lsign                     :char;
begin
  degtmp:=trunc(coordtmp); lsign:=' ';
    rmin:=60.0*frac(coordtmp); rsec:=60.0*frac(rmin);
  mintmp:=trunc(rmin); sectmp:=round(rsec);
     {If >180, then take complement angle and add minus sign ("-") to
      denote eastern longitude.}
  if coordtmp>=180.0 then begin
    degtmp:=360-degtmp; lsign:='-';
    if ((rmin>0.0) or (rsec>0.0)) then begin
      degtmp:=degtmp-1; mintmp:=60-mintmp;
      if (rsec>0.0) then begin
        mintmp:=mintmp-1; sectmp:=60-sectmp;
  end; end; end;
     {Adjustments in minutes and seconds due to rounding.}
  if sectmp=60 then begin sectmp:=00; mintmp:=mintmp+1; end;
  if mintmp=60 then begin mintmp:=00; degtmp:=degtmp+1; end;
  Async_Send_String('CA34;');
  Str(degtmp,sdegtmp); Str(mintmp,smintmp); Str(sectmp,ssectmp);
  Stg:=lsign+sdegtmp+chr(14)+chr(91)+chr(15)+smintmp+chr(39)+
  ssectmp+chr(34);
end;
```

```
{*************************************************************************}
     {HIMPSELE: Inclusion file containing symbol and scaling selection
        procedures.}
{*************************************************************************}
Overlay Procedure SymbolSelect;
     {Procedure that enables user to select a symbol for denoting station
      locations.}
  label label0;
  begin
label0:  ClearScreen; GotoXY(1,1);
     {Menu of symbols.  User may select any standard ASCII symbol via the
      keyboard.}
    WriteLn('SELECT DISPLAY SYMBOL TO DENOTE STATIONS ');
    WriteLn('  (1)    +'); WriteLn('  (2)    x');
    WriteLn('  (3)    box'); WriteLn('  (4)    filled box');
    WriteLn('  (5)    diamond'); WriteLn('  (6)    Y');
    WriteLn('  (7)    *'); WriteLn('  (8)    o');
    WriteLn('  (9)    Station Number ');
    WriteLn(' other   User defined (keyboard) symbol'); WriteLn;
    WriteLn('SELECT BY NUMBER OR BY KEYBOARD CHARACTER
                    FOR USER DEFINED SYMBOL');
    Read(Kbd,ch);
     {Set up character string to represent selected symbol.  Options 1, 2,
      6, 7, and 8 use predefined Turbo Pascal symbols.  Options 3, 4, and 5
      are custom-designed symbols defined in the HIMP. main program file.
      Option 9 uses integers and increments by 1.  Alternately, a character
      may be captured from the keyboard by entering any key other than 1-9.
      Character flag (cflag) tells HIMP the source of the symbol; cflag =
      0: ASCII-defined, cflag = 1: custom designed; cflag = 2: incremented
      integers.}
    Case ord(ch) of
        49 : begin
               symbol:=chr(27)+'1';
               symbolstg:=plus; cflag:=0;
             end;
        50 : begin
               symbol:=chr(27)+'2';
               symbolstg:=cross; cflag:=0;
             end;
        51 : begin
               symbol:=chr(27)+'3';
               symbolstg:=box; cflag:=1;
             end;
        52 : begin
               symbol:=chr(27)+'4';
               symbolstg:=fbox; cflag:=1;
             end;
        53 : begin
               symbol:=chr(27)+'5';
               symbolstg:=dia; cflag:=1;
             end;
```

```
        54 : begin;
               symbol:=chr(27)+'6';
               symbolstg:=yyy; cflag:=0;
             end;
        55 : begin;
               symbol:=chr(27)+'7';
               symbolstg:=star; cflag:=0;
             end;
        56 : begin
               symbol:=chr(27)+'8';
               symbolstg:=oh; cflag:=0;
             end;
        57 : begin
               ClearScreen;
               WriteLn('ENTER FIRST STATION NUMBER : ');
               ReadLn(FirstStnNo); Str(FirstStnNo,Stg);
               Symbolstg:='SM'+Stg+';'; cflag:=2;
               symbol:='N'; StnNo:=FirstStnNo;
             end;
          else begin
            ClearScreen;
            WriteLn('ENTER KEYBOARD CHARACTER TO BE USED
                     AS DISPLAY SYMBOL :');
            Read(Kbd,ch); symbol:=ch; symbolstg:='SM'+ch+';'; cflag:=0;
          end;
        end;
    MessageW;
     {Displays symbol and prompts user for verification.}
    DrawTextW(2.0,10.0,1,'SELECTED SYMBOL IS :');
    DrawTextW(20.0,20.0,2,symbol);
    DrawTextW(2.0,40.0,1,'Accept this symbol, or make new selection [N] ?');
    Read(Kbd,ch);
    if ch in ['N','n'] then goto label0;
  end;
{*****************************************************************************}
OverLay Procedure DataSymbolSelect;
     {Procedure that enables user to select a symbol to represent the data.
     Selection is similar to "symbol select" except for option 9 in which
     actual data values may be used.}
  label label0;
  begin
label0:  ClearScreen; GotoXY(1,1);
     {Clears screen and displays symbol menus.}
     {Character flag (cflag) tells HIMP the source of  the symbol; cflag =
     0: ASCII defined, cflag = 1: custom designed, cflag = 2:  captured
     from data value.}
    WriteLn('SELECT DISPLAY SYMBOL TO DENOTE DATA VALUES ');
    WriteLn('  (1)    +'); WriteLn('  (2)    x');
    WriteLn('  (3)    box'); WriteLn('  (4)    filled box');
    WriteLn('  (5)    diamond'); WriteLn('  (6)    Y');
    WriteLn('  (7)    *'); WriteLn('  (8)    o');
    WriteLn('  (9)    Data Value ');
```

```
WriteLn(' other   User defined (keyboard) symbol'); WriteLn;
WriteLn('SELECT BY NUMBER OR BY KEYBOARD CHARACTER
                FOR USER DEFINED SYMBOL');
Read(Kbd,ch);
  Case ord(ch) of
    49 : begin
           symbol:=chr(27)+'1';
           symbolstg:=plus; cflag:=0;
         end;
    50 : begin
           symbol:=chr(27)+'2';
           symbolstg:=cross; cflag:=0;
         end;
    51 : begin
           symbol:=chr(27)+'3';
           symbolstg:=box; cflag:=1;
         end;
    52 : begin
           symbol:=chr(27)+'4';
           symbolstg:=fbox; cflag:=1;
         end;
    53 : begin
           symbol:=chr(27)+'5';
           symbolstg:=dia; cflag:=1;
         end;
    54 : begin;
           symbol:=chr(27)+'6';
           symbolstg:=yyy; cflag:=0;
         end;
    55 : begin;
           symbol:=chr(27)+'7';
           symbolstg:=star; cflag:=0;
         end;
    56 : begin
           symbol:=chr(27)+'8';
           symbolstg:=oh; cflag:=0;
         end;
    57 : begin
           ClearScreen; GotoXY(1,1); mm:=0;
 {Prompts to determine stray length for symbol.}
           WriteLn('How many significant figures (0 for integers) ?');
           Read(ch); Val(ch,mm,err); WriteLn;
           WriteLn('Symbol will be the corresponding data value.');
           WriteLn('(Represented by the letter N.)');
           cflag:=2;
           symbol:='N';
         end;
      else begin
        ClearScreen;
        WriteLn('ENTER KEYBOARD CHARACTER TO BE USED
                 AS DISPLAY SYMBOL :');
        Read(Kbd,ch); symbol:=ch; symbolstg:='SM'+ch+';'; cflag:=0;
```

```
          end;
        end;
    MessageW;
     {Displays symbol and prompts user for verification.}
    DrawTextW(2.0,10.0,1,'SELECTED SYMBOL IS :');
    DrawTextW(20.0,20.0,2,symbol);
    DrawTextW(2.0,40.0,1,'Accept this selection [Y or CR] ?');
    Read(Kbd,ch);
    if not (ch in ['Y','y',chr(13)]) then goto label0;
  end;
{*********************************************************************}
OverLay Procedure PtScaleSelect;
     {Procedure to select symbol size displayed on screen.  User selects
sizes in multiples of screen scale 1 (see Turbo Graphix Toolbox by pressing
0-9 keys.  User accepts value as soon as non-integer key is pressed.}
  begin
    ClearScreen; ch:='1';
    Repeat
      scale:=ord(ch)-48; MessageW;
      DrawTextW(2.0,5.0,1,'SELECTED SYMBOL IS');
      DrawTextW(40.0,10.0,scale,symbol);
      DrawTextW(2.0,15.0,1,'WITH SCALE SIZE '+ch);
      DrawTextW(2.0,25.0,1,'Select new scale size (integer multiple)');
      DrawTextW(2.0,35.0,1,'or accept current scale (hit non-integer key)');
      Read(Kbd,ch);
     {Repeat cycle until non-integer key is pressed.}
    until not (ch in ['0'..'9']);
  end;
{*********************************************************************}
procedure PtValueSelect;
     {Procedure that enables user to select partition limits and
      corresponding symbol sizes.}
  begin
    bins:=1;
    ClearScreen; GotoXY(1,1);
     {Select number of bins desired.  Depending on number, the procedure
      will prompt user for limits and then call PtScaleSelect to assign a
      corresponding symbol size.}
    WriteLn('Select number of data bins [1,2,3,4, or 5] : '); Read(bins);
    if bins=1 then begin
      PtScaleSelect; scale1:=scale;
    end;
    if bins>=2 then begin
      ClearScreen; GotoXY(1,1);
      WriteLn('Enter minimum data value for bin 1 + Return : ');
      ReadLn(bin0);
      WriteLn('Enter maximum data value for bin 1 + Return : ');
      ReadLn(bin1);
      ClearScreen; GotoXY(1,1);
      WriteLn('Select scale for bin 1 symbol'); Delay(1000);
      PtScaleSelect; scale1:=scale;
      ClearScreen; GotoXY(1,1);
```

```
      WriteLn('Enter maximum data value for bin 2 + Return : ');
      ReadLn(bin2);
      ClearScreen; GotoXY(1,1);
      WriteLn('Select scale for bin 2 symbol'); Delay(1000);
      PtScaleSelect; scale2:=scale;
    end;
    if bins>=3 then begin
      ClearScreen; GotoXY(1,1);
      WriteLn('Enter maximum data value for bin 3 + Return : ');
      ReadLn(bin3); ClearScreen; GotoXY(1,1);
      WriteLn('Select scale for bin 3 symbol'); Delay(1000);
      PtScaleSelect; scale3:=scale;
    end;
    if bins>=4 then begin
      ClearScreen; GotoXY(1,1);
      WriteLn('Enter maximum data value for bin 4 + Return : ');
      ReadLn(bin4); ClearScreen; GotoXY(1,1);
      WriteLn('Select scale for bin 4 symbol'); Delay(1000);
      PtScaleSelect; scale4:=scale;
    end;
    if bins>=5 then begin
      ClearScreen; GotoXY(1,1);
      WriteLn('Enter maximum data value for bin 5 + Return ; ');
      ReadLn(bin5); ClearScreen; GotoXY(1,1);
      WriteLn('Select scale for bin 5 symbol'); Delay(1000);
      PtScaleSelect; scale5:=scale;
    end;
  end;
{*************************************************************************}
```

APPENDIX IV

BASIC DIGITIZING PROGRAM

Several BASIC programs have been written for the Summagraphics Digitizing Table. The following code lists a user-friendly version (DIGITIZE.EXE) that records the digitizing signal in the form of (latitude, longitude) records. The user may select the digitizing density in points per inch. Note that the proper format for responses to coordinate prompts is ddd:mm:ss, indicating degree, minute, and seconds. For longitudes, all values must be in the range 000:00:00 to 359:59:59. For latitudes, all values must be in the range -90:00:00 to +90:00:00 (or b90:00:00, where b = space). To accommodate the higher recording rates (higher densities), a fixed disk is recommended. The current version of DIGITIZE uses the 8087 math coprocessor and must have the BRUN3087.EXE library present to operate.

Filename: DIGITIZ.BAS

```
 10    ON ERROR GOTO ERRORR
 20    CLS:COLOR 11
 30    PRINT:PRINT "PROGRAM FOR DIGITIZING MAPS"
 40    PRINT:PRINT "all data will be stored on hard disk C:, in the ";
       "current subdirectory "
 50    COLOR 10:PRINT:INPUT "Press ENTER to continue",DUMMY
 60 BEGINN:
 70   CLS: COLOR 11
 80   OPEN "com1:2400,E,7,2" AS #1
 90   PRINT:INPUT "                          ALIGN CHART ON TABLE (Y/N) ";Z$
100   IF NOT (Z$="Y" OR Z$="y") THEN GOTO INITIATE
110   PRINT #1,CHR$(27);"M1"
120   CLS:PRINT:PRINT:PRINT "                 PRESS ";:COLOR 14:
      PRINT "YELLOW, ";:COLOR 15:PRINT "WHITE, ";:COLOR 9:PRINT "BLUE ";:
      COLOR 11:PRINT "CURSOR BUTTONS TO INPUT DATA"
130   PRINT:PRINT "                 PRESS ";:COLOR 10:PRINT "GREEN ";:
      COLOR 11:PRINT "BUTTON TO END ALIGNMENT PROCEDURE":PRINT
140   INPUT #1,X,Y,C,D
150   COLOR 11:PRINT "                ";:PRINT X,Y
160   IF C=4 THEN GOTO INITIATE   'TERMINATE IF GREEN BUTTON PUSHED
170   GOTO 140
180 INITIATE:
190   CLS:PRINT:INPUT "                 CREATING A NEW FILE OR APPENDING ";
      "TO AN OLD? (N/O) ";A$:PRINT
200   COLOR 10:INPUT "                ENTER DATA FILENAME (name.DAT)   ",
      GG$:GG$="C:"+GG$:COLOR 11
210   IF LEFT$(A$,1) = "N" OR LEFT$(A$,1) = "n" THEN GOTO NEWFILE
220 OLD:
230   PRINT:PRINT:PRINT "WHEN APPENDING TO AN EXISTING FILE THE MIN/MAX ";
      "VALUES"
240   PRINT:PRINT "OF THE X/Y AXES MUST ALSO BE TRUE FOR THE APPENDED DATA."
```

```
250   OPEN "I",#2,GG$:INPUT #2,LATMIN,LATMAX,LONGMIN,LONGMAX
260   PRINT:COLOR 10:PRINT "Y-AXIS RANGE = ",LATMIN,"-",LATMAX:
      PRINT "X-AXIS RANGE = ",LONGMIN,"-",LONGMAX
270   PRINT:PRINT:INPUT "ARE THESE CORRECT (Y/N) ";Z$:CLOSE #2:COLOR 11
280   IF LEFT$(Z$,1)="y" OR LEFT$(Z$,1)="Y" THEN GOTO OLDFILE
290   GOTO INITIATE
300 OLDFILE:
310   OPEN "A",#2,GG$:EE=1:GOTO FILE
320 NEWFILE:
330   OPEN GG$ FOR OUTPUT AS #2
340 FILE:
350   CALL CHARINIT
360   CALL CHARGO
370   CLS:INPUT "DIGITIZE A NEW FILE (Y/N) ?";Z$
380   IF LEFT$(Z$,1)="y" OR LEFT$(Z$,1)="Y" THEN GOTO BEGINN
410   END
420 SUB CHARINIT STATIC
430 SHARED X0,Y0,X2,Y2,L0,A0,L2,A2,EE,LATMAX,LATMIN,LONGMAX,LONGMIN
440   PRINT #1,CHR$(27);"M1"
450   CLS:PRINT:PRINT "ENTERING MAP COORDINATE LIMITS (LATITUDE AND ";
      "LONGITUDE)"
460   COLOR 10:PRINT:PRINT "PRESS ANY CURSOR BUTTON TO INPUT DATA"
470   PRINT:PRINT "THEN ENTER CORRESPONDING LATITUDE AND LONGITURE ";
      "COORDINATES"
480   COLOR 11:PRINT:PRINT:PRINT "INPUT UPPER LEFT CORNER OF MAP"
490   INPUT #1,X0,Y0,C,D
500   IF EE=1 THEN A0=LATMAX:L0=LONGMAX:GOTO LRC
510   GOSUB LLIN
520   A0 = LAT:L0 = LON
530 LRC:
540   PRINT:PRINT "ENTER LOWER RIGHT CORNER OF MAP"
550   INPUT #1,X2,Y2,C,D
560   IF EE=1 THEN A2=LATMIN:L2=LONGMIN:GOTO XIT
570   GOSUB LLIN
580   A2 = LAT:L2 = LON:Z = 0
590   PRINT #2,A2,A0,Z:PRINT #2,L2,L0,Z
600 XIT:
610   EXIT SUB
620 LLIN:
622   PRINT "ENTER LATITUDE (-90,00,00 TO 90,00,00)"
624   PRINT "NOTE: FOR SOUTH LATS, USE ONLY MINUS SIGN FOR DEGREES ";
      "(-5,30,10; NOT -5,-30,-10)."
630   PRINT:INPUT "LATITUDE -- DEGREES,MINUTES,SECONDS (SEPARATED BY ";
      "COMMAS) :";DEG,MIN,SEC
650   IF DEG>0 THEN LAT = DEG+MIN/60+SEC/3600
      ELSE LAT = DEG-MIN/60-SEC/3600
654   PRINT "ENTER ABSOLUTE LONGITUDE (000,00,00 TO 360,00,00, NO SIGN)"
660   PRINT: INPUT "LONGITUDE -- DEGREES,MINUTES,SECONDS (SEPARATED BY ";
      "COMMAS) :";DEG,MIN,SEC
```

```
680   LON = DEG+MIN/60+SEC/3600
684   PRINT "LAT, LONG :  ":PRINT USING "####.#####";LAT;LON;
686   FOR IW = 1 TO 5000
688   NEXT IW
689   RETURN
690 END SUB
700 SUB CHARGO STATIC
710   SHARED X0,Y0,X2,Y2,A0,A2,L0,L2
712   PRINT:PRINT " LAT = " LAT "   LONG = " LON
720   CLS:INPUT "INPUTING CONTOURS OR SINGLE POINTS (C/S) ";A$
730   IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN CC=1:GOTO PTIN1
740 DENSITY:
750   CLS:COLOR 10:PRINT "PLEASE ENTER DIGITIZING DENSITY OPTION":
      COLOR 11
760   PRINT:PRINT "  (1) 10 PPI (POINTS PER INCH)"
770   PRINT "  (2) 20 PPI ":PRINT "  (3) 50 PPI"
780   PRINT "  (4) 100 PPI":PRINT:INPUT "       SELECT BY NUMBER :",D$
790   IF LEFT$(D$,1) = "1" THEN PRINT #1,CHR$(27);"I100":GOTO DENEND
800   IF LEFT$(D$,1) = "2" THEN PRINT #1,CHR$(27);"I050":GOTO DENEND
810   IF LEFT$(D$,1) = "3" THEN PRINT #1,CHR$(27);"I020":GOTO DENEND
820   IF LEFT$(D$,1) = "4" THEN PRINT #1,CHR$(27);"I010":GOTO DENEND
830   GOTO DENSITY
840 DENEND:
850   PRINT #1,CHR$(27);"M2"     'SET GRAPHIC TABLET FOR STREAM MODE
860 PTIN1:
863   SCREEN 8:
865   SXM=640:SYM=200:GXM=48:GYM=36
870   CLS:LOCATE 1,25:COLOR 14:PRINT "YELLOW BUTTON - TRACK LINE "
880       LOCATE 2,25:COLOR 15:PRINT "WHITE BUTTON - TICK MARKS "
890       LOCATE 3,25:COLOR 10:PRINT "GREEN BUTTON - CHANGE LINE TYPES "
900       LOCATE 4,25:COLOR 9:PRINT "BLUE BUTTON - QUIT":COLOR 11
901   AX0=INT(X0*SXM/GXM):AY0=SYM-INT(Y0*SYM/GYM)
902   AX2=INT(X2*SXM/GXM):AY2=SYM-INT(Y2*SYM/GYM)
903   PSET(AX0,AY0):CIRCLE STEP(0,0), 3
904   PSET(AX2,AY2):CIRCLE STEP(0,0), 3
905   PSET(AX0,AY2):CIRCLE STEP(0,0), 3
906   PSET(AX2,AY0):CIRCLE STEP(0,0), 3
910   INPUT #1,A,B,C,D
920   IF D <> 0 THEN INPUT #1,D
930   X = (((A-X2)/(X0-X2))*(L0-L2))+L2
940   Y = (((B-Y2)/(Y0-Y2))*(A0-A2))+A2
950   IF C = 1 THEN Z = 0:COLOR 14
960   IF C = 2 THEN Z = 1:BEEP:COLOR 15
970   IF C = 3 THEN BEEP:BEEP:GOTO PTEND
980   IF C = 4 THEN X = 0:Y = 0:Z = 0:BEEP:COLOR 10
990   PRINT #2,Y,X,Z
995    LOCATE 25,25:PRINT "                                    ";
1000  LOCATE 25,25:PRINT USING "#######.##";Y;X;Z;
1002  AX1 = INT(A*SXM/GXM):AY1 = SYM-INT(B*SYM/GYM)
1003  PSET(AX1,AY1)
1005  IF C = 2 THEN CIRCLE STEP (0,0), 2
1007  IF C = 4 THEN CIRCLE STEP (0,0), 2
```

```
1008   GOTO 910
1010 PTEND:
1015   PRINT #2,Y,X,Z;
1020   CLOSE:SCREEN 0:COLOR 11
1030   END SUB
1040 ERRORR:
1045   IF ERR=24 THEN BEEP:CLS:PRINT "GRAPHICS TABLET NOT SWITCHED ON":
       END
1050   IF ERR=57 THEN CLOSE:BEEP:CLS:PRINT "GRAPHICS TABLET ERROR HAS ";
       "OCCURED":GOTO 1100        'GRAPHICS TABLET NOT RESPONDING
1060   IF ERR=53 THEN CLS:PRINT "FILE NOT FOUND": GOTO 1100
1070   CLS:COLOR 12
1080   PRINT "FATAL ERROR NUMBER "ERR " HAS OCCURRED IN LINE "ERL:PRINT
1090   PRINT "PROGRAM HAS BEEN TERMINATED":END
1100   COLOR 11:PRINT:INPUT "Press ENTER to resume program execution",
       DUMMY
1110   IF ERR=53 THEN RESUME INITIATE
1120   IF ERR=57 THEN RESUME BEGINN
```

APPENDIX V

HIMP HARD DISK INSTALLATION INSTRUCTIONS

1. On the hard disk, create the HIMP system directories by using the following DOS commands, from the hard disk prompt (denoted here by C:\>).

   a) C:\> MD HIMP
   b) C:\> CD HIMP
   c) C:\HIMP> MD HMAP
   d) C:\HIMP> MD SOURCE (Optional directory)
   e) C:\HIMP> MD EX (Optional directory)

2. Copy the executable HIMP files and runtime support files from the A:HIMP directory to the hard disk HIMP directory.

   a) Place HIMP floppy into the A: drive.
   b) C:\HIMP> A: (Change to A: drive)
   c) A:\> CD HIMP (Change to HIMP dir)
   d) A:\HIMP> copy *.* C:
   (Copies 7 HIMP and 5 Turbo support files.)

3. Copy map library from floppy to hard disk.

   A:\HIMP> copy hmap*.* C:hmap

4. Copy source code from floppy to hard disk. (Optional)

   A:\HIMP> copy source*.* C:source

5. Copy example files from floppy to hard disk. (Optional)

   A:\HIMP> copy ex*.* C:ex