A GENERAL FORTRAN COMPUTER PROGRAM FOR USE WITH THE IBM 2250 DISPLAY UNIT

by

Harlan Langtry Morrison
United States Naval Postgraduate School

THESIS

A GENERAL FORTRAN COMPUTER PROGRAM FOR USE WITH THE IBM 2250 DISPLAY UNIT

by

Harlan Langtry Morrison

April, 1970

This document has been approved for public release and sale; its distribution is unlimited.
A General FORTRAN Computer Program for
Use with the IBM 2250 Display Unit

by

Harlan Langtry, Morrison
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1961

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the
Naval Postgraduate School
April, 1970
ABSTRACT

A General computer program was designed, using the FORTRAN IV language, to utilize the IBM 2250 Display Unit in conjunction with the IBM System/360 Operating System. The user may adapt an associated program so that it may be an integral part of this general program. The general program provides the user with the ability to display a static information array and an input data array. The user may insert input data, for his associated program, from on-line input devices or directly through the IBM 2250 alphanemic keyboard and may change this data from the keyboard. The associated program may be executed repeatedly, subject to system time constraints, and both alphanemic and graphical output data from the associated program may be displayed after each execution.
# TABLE OF CONTENTS

I. INTRODUCTION .................................................................................. 5

II. DESCRIPTION OF THE GENERAL FORTRAN PROGRAM .................. 7
   A. CAPABILITIES OF THE GFP ......................................................... 7
   B. GRAPHIC DATA SETS ............................................................... 8
   C. DATA ARRAYS ........................................................................... 8
   D. DISPLAY IN IGDS1 AREA .......................................................... 11
   E. PROGRAM FUNCTION KEYBOARD ............................................. 12

III. ASSOCIATED PROGRAM COMPATABILITY REQUIREMENTS ............ 16
   A. GENERAL .................................................................................. 16
   B. DIMENSIONING OF ARRAYS ..................................................... 17
   C. SPECIFIC VARIABLE INITIALIZATION ....................................... 19
   D. EXTERNAL INSERTION OF INPUT DATA .................................... 19

IV. SAMPLE GFP EXECUTION CHAIN OF EVENTS ................................. 21
   A. DISPLAY OF INPUT1 ARRAY ...................................................... 22
   B. DISPLAY, CHANGING AND STORING OF INPUT2 DATA ............... 23
   C. DISPLAY OF OUT1 DATA .......................................................... 25
   D. GRAPH LABELING ................................................................. 26
   E. DISPLAY OF LINE GRAPHS ...................................................... 27
   F. PRINTING OF INPUT2 AND OUT1 ARRAYS ON LINE PRINTER ....... 28

V. SUMMARY AND RECOMMENDATIONS .............................................. 29

APPENDIX A FLOW CHART OF GFP .................................................. 30
APPENDIX B LIST OF VARIABLES USED BY GFP ............................... 34
APPENDIX C GFP COMPUTER PROGRAM LISTING ............................... 38
LIST OF REFERENCES ........................................................................ 58
INITIAL DISTRIBUTION LIST .......................................................... 59
FORM DD 1473 ................................................................................. 61
I. INTRODUCTION

The use of visual display units associated with present generation computer systems creates a man-machine interface which allows the user to communicate directly with the computer, during the execution of his program. As a teaching aid, the use of a visual display unit affords the visual presentation of information to the student as it is being discussed and computational results may be graphically displayed immediately, while the information is fresh in his mind. In addition, with the ability to change data or alter sequences of operations, comparisons may be rapidly made between various options in a problem to determine which may be relevant or where further investigation is warranted.

The purpose of this thesis was to construct a general graphic display program for the IBM 2250 Display Unit. This equipment is associated with the IBM System/360 Operating System. This program is not limited in use as a teaching aid; its applicability may be extended to all situations where graphic display is desired.

This thesis presents one method of utilizing the IBM 2250 Display Unit through a General FORTRAN Program called GFP. The GFP was designed such that another associated computer program may be incorporated, through appropriate modifications, as an integral part of the GFP. The input and output data from the associated program and the associated program may then be displayed on the screen of the display unit and changes to the input data may be made using the graphic display unit keyboard.
The use of the GFP does not require that the user have previous practical experience with the IBM designed Graphic Subroutine Package; however, it is expected that he be familiar with the general information concerning graphic programming services and the IBM 2250 as described in Ref. 2. It is to be remembered that this GFP is not a unique method for using the display unit, but only one possible aid for using this equipment. A person familiar with graphic subroutines and functions could develop a graphic program suited exactly to the specific requirements of his problem, thereby making the best use of the capabilities of the display unit. This program, however, allows the relatively inexperienced user to take advantage of some of the capabilities of the display unit, rather than none at all.

The associated program which was used during the development of the GFP, and is included in the GFP listing in Appendix C, is a minesweeper simulation program, written by Professor A. F. Andrus and Miss P. R. Hoang of the Naval Postgraduate School. The complete description of this example associated program is provided in Ref. 1. This associated program was modified both for use with the IBM System/360 Operating System and the GFP.

Chapter II provides a general description of the capabilities of the GFP, the graphic data sets, arrays, and the program function keyboard keys used. Chapter III presents the detailed instructions, using the GFP listing in Appendix C, for making an associated program compatible with the GFP. Chapter IV provides a possible chain of events during the execution of the GFP and describes the physical operations which occur.
II. DESCRIPTION OF THE GENERAL FORTRAN PROGRAM

The General Fortran Program was written in accordance with the instructions as specified in Refs. 2 and 3. The design of the IBM 2250 allows only for EBCDIC characters to be displayed on and read into the program from the screen of the Display Unit. Therefore, all data displayed on the screen must be in the A format. To overcome the obstacle of the input of numerical data and the display of numerical output data, computed by the associated program, a direct access disc storage device was utilized. This allows, with the use of input and output arrays, for an alphameric/numerical interface through appropriate read and write statements formatted for use with a direct access disc storage device. The DEFINE FILE statement in the GFP specifies direct access disc storage for this purpose.

A. CAPABILITIES OF THE GFP

In conjunction with an associated program, the GFP will allow the user to perform the following operations.

1. A specified array of information may be read into the GFP and displayed on the screen in 25-line pages.

2. The input data to be used by an associated program may be read into the GFP and displayed. This data will be displayed in 20-line pages. The user may also enter this data directly from the alphameric keyboard or modify data which was previously read.

3. The associated program may be executed and re-executed as desired, subject to system restrictions on total allowable execution time of the GFP.
4. A rectangular coordinate system grid may be displayed and labeled. The graphical data limits of the lines to be drawn on this graph must be specified by the user.

5. Output data, both tabular and graphical, from the associated program may be displayed. The tabular data is displayed in 25-line pages. The two line graphs, labeled "1" and "2," may be displayed simultaneously.

6. The input and output data, which was used and produced by the associated program, may be printed on the line printer.

B. GRAPHIC DATA SETS

The GFP utilizes three graphic data sets named IGDS1, IGDS2, and IGDS3. IGDS1 encompasses the lower half of the screen and is used for all alphameric input and output displays. IGDS2 encompasses the upper half of the screen and is used to display rectangular coordinate graph axes and the desired numerical scales and labels for these axes. The data for the line graphs is generated by the associated program. The axes are permanently displayed in the center of the IGDS2 area, the length of each axis being four tenths of the length of the entire screen. IGDS3 encompasses that square screen area bordered on the left and bottom by the permanently displayed axes which were initialized in IGDS2. The intersection of the two axes is the lower left corner of IGDS3. The data limits for IGDS3 are variable and detailed instructions for setting these limits are discussed in Chapter III. Figure 1 shows the locations of the graphic data sets on the display screen.

C. DATA ARRAYS

The GFP uses eight arrays for storage of the user's input data and the output data from the associated program. These arrays are named
Figure 1
Location of Graphic Data Sets on Screen
INPUT1, INPUT2, OUT1, OUT2, OUT3, OUT4, OUT5, and OUT6. INPUT1, INPUT2, and OUT1 arrays are blanked out by a data statement upon initial execution of the GFP. The following paragraphs present a description of these eight arrays.

The INPUT1 array is dimensioned for 500 lines of 18 words, each word containing four characters. The user may utilize this array for display of the associated program or other alphameric information in IGDS1. This is a static array; the information must be read in at the beginning of the program and stored for subsequent display. INPUT1 cannot be used for other than this purpose.

The INPUT2 array is dimensioned for 50 lines of 18 words, each word containing four characters. Its use is to store input data utilized by the associated program. This data may be displayed in IGDS1, changed by use of the alphameric keyboard if desired, and may be transferred into the direct access storage disc by the user. The data in this array may also be printed on the line printer if desired.

The OUT1 array is dimensioned for 350 lines of 18 words, each word containing four characters. It is the storage array for alphameric data output from the associated program. The information in this array may be displayed in IGDS1, printed on the line printer, or blanked out as desired by the user. Blanking out of this array after an execution of the associated program would erase all previous data, and would preclude the possibility of displaying or printing erroneous output data after the next execution of the associated program.

The OUT2 array is dimensioned for one line of 18 words, each word containing four characters. It is the storage array for one line of data from the INPUT2 array which is determined to be in error by the
associated program. The associated program must have a method to check input data for correctness, as in the example associated program in Appendix C, and to read that line of data into the array, if this array is to serve any purpose.

The OUT3 array is dimensioned for ten words and is the storage array for the dependent variable for graphical display in IGDS3. The array is filled directly during the execution of the associated program.

The OUT4 array is dimensioned for ten words and is the storage array for the independent variable for graphical display in IGDS3. The array is filled directly during execution of the associated program. This array, in conjunction with the OUT4 array, provides the data points for display of the line graph labeled "1."

The OUT5 and OUT6 arrays are dimensioned for ten words and are used to store the data in the OUT3 and OUT4 arrays respectively. These arrays provide data points for display of the graph labeled "2." Since the values in OUT3 and OUT4 arrays are set during execution of the associated program, arrays OUT5 and OUT6 provide storage for data points from a previous execution of the associated program. Both graphs "1" and "2" may be displayed simultaneously for comparison.

D. DISPLAY IN IGDS1 AREA

The INPUT1 or OUT2 array data may be displayed on the screen in 25-line increments, and the INPUT2 array data may be displayed on the screen in 20-line increments. These increments will be referred to as pages. Through the use of the applicable keys on the program function keyboard, these pages may be advanced or rewound to allow for display of the entire contents of any of the three arrays. The advance and rewind operations may be accomplished in ten or one page increments or ten and
five line increments. The display of the last set of data in an array will occupy less than a full page when the number of lines of data in this set is less than the number of lines on a page. For example, in the example associated program there are 34 lines of data in the INPUT2 array. The first page displayed will include the first 20 lines, but after advancing the display one page, only 14 lines will be displayed, lines 21 through 34. If, while advancing the pages in an array, the number of the first line to be displayed is advanced beyond the specified limits of that array, a message will be displayed indicating the end of the input/output array.

E. PROGRAM FUNCTION KEYBOARD

The GFP uses 29 function keys. The use of each key is described below.

Key 1 has two uses. Its primary use is to display the first 25 lines of data, stored in the INPUT1 array, in the IGDS1 area. In addition, when used immediately after any of keys 2 through 9 or 11, the specified 25 lines data in the INPUT1 array is displayed in IGDS1.

Key 2 has two uses. Its primary use is to advance the line counters, which determine the next array segment to be displayed in IGDS1, ten 25-line pages beyond their present positions. When used immediately after any of keys 2 through 9 or 11, the specified 20 lines of data in the INPUT2 array is displayed in IGDS1.

Key 3 has two uses. Its primary use is to reset the line counters, which determine the next array segment to be displayed in IGDS1, ten 25-line pages back from their present positions. When used immediately after any of keys 2 through 9 or 11, the specified 25 lines of data in the OUT1 array is displayed in IGDS1.
Key 4 advances the line counters, which determine the next array segment to be displayed in IGDS1, one 25-line page beyond their present positions.

Key 5 resets the line counters, which determine the next array segment to be displayed in IGDS1, one 25-line page back from their present positions.

Key 6 advances the line counters, which determine the next array segment to be displayed in IGDS1, ten lines beyond their present positions.

Key 7 resets the line counters, which determine the next array segment to be displayed in IGDS1, ten lines back from their present positions.

Key 8 advances the line counters, which determine the next array segment to be displayed in IGDS1, five lines beyond their present positions.

Key 9 resets the line counters, which determine the next array segment to be displayed in IGDS1, five lines back from their present positions.

Key 10 displays in IGDS1 the first 20 lines of the input data stored in the INPUT2 array. It also displays the instructions for changing the data in the INPUT2 array.

Key 11 advances the line counters, which determine the next array segment to be displayed in IGDS1, 20 lines beyond their present positions. This key is designed to assist in advancing the data in the INPUT2 array by one page increments because the specified page length for INPUT2 array data is 20 lines rather than 25 lines.
Key 12 displays a cursor under the first character position in the first line of displayed INPUT2 data. This key is only used when INPUT2 data is displayed.

Key 13 advances the cursor one line and displays the cursor under the first character of the next line.

Key 14 removes the cursor from the display.

Key 15 transmits the page of displayed INPUT2 data into the INPUT2 array. If this data has been changed on the display screen with the alphameric keyboard, the new data is stored in the INPUT2 array. This data must be read into the INPUT2 array before the next page of data is displayed.

Key 16 transfers the data in the INPUT2 array to the direct access disc storage area. The associated program reads its input data from this intermediate disc storage space.

Key 17 blanks out the OUT1 designated section on the direct access storage disc. This key should be used only after using key 18 to prevent the loss of output data produced during execution of the associated program.

Key 18 transfers the alphameric data output from the direct access disc storage to the OUT1 array.

Key 19 displays the first 25 lines of data, stored in the OUT1 array, in IGDS1.

Key 20 displays the axes and labeling for the graphical output in IGDS2, and displays a cursor under the top zero in the column of zeros alongside the vertical axis. This zero may be changed by use of the alphameric keyboard.
Key 21 advances the cursor to the next designated line in IGDS2, and displays the cursor under the first character in that line.

Key 22 blanks out IGDS3. Keys 23 or 25 must be used to display a line in IGDS3. At the beginning of execution of the GFP an initialization line is displayed diagonally across IGDS3. The use of this key will remove that line when it is desired to display a graph.

Key 23 displays a line graph in IGDS3, utilizing data in OUT3 and OUT4 arrays, and labels this graph "1."

Key 24 transfers data stored in the OUT3 and OUT4 arrays to the OUT5 and OUT6 arrays respectively.

Key 25 displays a line graph in IGDS3, utilizing the data in OUT5 and OUT6 arrays, and labels this graph "2." The two line graphs, labeled "1" and "2" may be displayed simultaneously. Additional graphs may be simultaneously displayed but will be labeled "1" or "2." Therefore, to display more than two graphs, the user will need to use a mechanical method to determine which data produced the additional line graphs. If the user uses key 22, only the data stored in the OUT3/OUT4 and OUT5/OUT6 arrays will be available to produce line graphs until another execution of the associated program is completed.

Key 26 transfers to the line printer, the current contents of the input data stored in the INPUT2 array.

Key 27 transfers to the line printer, the current contents of the output data stored in the OUT1 array.

Key 28 initializes the execution of the associated program.

Key 29 terminates the GFP.
III. ASSOCIATED PROGRAM COMPATABILITY REQUIREMENTS

This chapter describes, in detail, the requirements to make an associated program compatible with the GFP. The GFP program is contained in Appendix C and direct referral will be made to this program using the line index numbers.

A. GENERAL

The GFP manipulates and displays the data to and from the associated program. It provides intermediate direct access disc storage space of 800 lines, each of 72 characters in length. This disc space is used to temporarily store INPUT2, OUT1, and OUT2 data during execution of the GFP. The GFP transfers the input data in the INPUT2 array to disc storage. The associated program, then, must read this data from the disc storage in a manner similar to that in the INPUT SUBROUTINE, line 462, in the example associated program. The associated program must store alphanemic output data, for later transfer by GFP to the OUT1 array, on intermediate disc space in a manner similar to that in the TITLE SUBROUTINE, line 587, in the example associated program. The associated program must store the line of improper input data, if any, on intermediate disc storage space and transfer this to the OUT2 array in a manner similar to that in the INPUT SUBROUTINE, lines 576 through 579, in the example associated program. The associated program must also store numerical data in the OUT3 and OUT4 arrays for use in the display of line graphs.

The user must always set the variables NMAX, NDATA, MAXOUT, N1, GR1, GR2, GR3, and GR4, described in section C of this chapter, to
correspond to the requirements of his associated program. The GFP may be utilized without further modifications if the user reads external data into GFP for the INPUT1 and INPUT2 arrays and the total number of 72-character lines of data stored in the INPUT1, INPUT2, OUT1, and OUT2 arrays are less than or equal to 500, 50, 350, and 1 respectively. In addition, the number of data point sets to be utilized for any line graph must not be greater than ten.

B. DIMENSIONING OF ARRAYS

The GFP utilizes the arrays described in this paragraph exclusively and their dimensions must not be changed. The one word array NULL is required for use as the null variable. The one word array IBIT is required for storage of light pen attention or end-order-sequence information by the Request Attention Information subroutine. The one word array IT is required for storage of information related to the completion of the read operation in the Read Text subroutine. The array IKEY is required for identification of elements associated with the Plot Text, Insert Cursor, and Read Text subroutines and must be dimensioned for 25 words, the maximum number of lines displayed in IGDS1 at any time.

It may be necessary for the user to modify the dimensions of the INPUT and OUT arrays, either by decreasing some of them to reduce total core requirements for the GFP with an integrated associated program, or by increasing some of them to provide sufficient storage space for the desired input and output data. If this situation occurs, the following changes to the arrays and associated program variables are necessary to ensure associated program compatibility.
If the sum of the second dimensions of INPUT2 and OUT1, and OUT2 arrays are greater than 800, the second dimension of the DEFINE FILE statement, line 36, must be appropriately increased. Because of the format of the read and write statements in the GFP, the first dimension of INPUT1, INPUT2, OUT1, and OUT2 arrays should not be changed from 18 and the second dimensions must be at least one. If the second dimension of the INPUT2 array is increased beyond 50 and/or the second dimension of OUT1 array is increased beyond 350, it will be necessary to insure that no intermediate disc storage records are inadvertently overwritten.

In line 379 of the example associated program, the variable W defines the first location for storing the OUT1 array on the disc. The user must specify the value of the corresponding variable in his associated program such that it is one greater than the value of the second dimension of the INPUT2 array. In lines 249 and 260, the variable I values must also range from one more than the value of the second dimension of the INPUT2 array to the sum of the second dimensions of INPUT2 and OUT1 arrays, to ensure proper blanking out or reading in of data. Also, as in lines 576 and 579 in the example associated program, the user must specify in the corresponding statement in his associated program that the initial location for storing the data for the OUT2 array be one greater than the sum of the second dimensions of the INPUT2 and OUT1 arrays. If the dimensions of INPUT1, INPUT2, or OUT1 arrays are changed, corresponding changes will be required for initial blanking out of these arrays by the DATA statement, line 32.

The dimensions of OUT3, OUT4, OUT5, and OUT6 arrays are determined by the number of points to be used for displaying each line graph in
IGDS3. The dimensions of these arrays should be increased, if necessary, and they should all be dimensioned the same value.

C. SPECIFIC VARIABLE INITIALIZATION

In lines 43 through 50, initialization variables are specified. NMAX specifies the actual number of lines of data used in the INPUT1 array. NDATA is the actual number of lines of data in the INPUT2 array and MAXOUT is the actual number of lines in the OUT1 array. These three variables determine the operations to be performed in lines 80, 86, 115, 182, 241, 269, 362, and 370. It is also necessary that the second dimensions in INPUT1, INPUT2, and OUT1 arrays be greater than or equal to the values of NMAX, NDATA, and MAXOUT respectively.

N1 is the variable designating the number of lines to be drawn on each line graph. The number of points to be stored in OUT3, OUT4, OUT5, and OUT6 arrays must be one more than this number. The first point in these arrays will be the initial starting positions for the line graphs.

The variables GR1 and GR2 designate the X and Y values respectively at the lower left corner of IGDS3, which is the intersection of the line graph axes. GR3 and GR4 designate the maximum X and Y values respectively of the data range of the line graphs, and is the upper right corner of IGDS3. GR1, GR2, GR3, and GR4 must be chosen by the user according to the specific requirements of his associated program. The GFP automatically scales the graph area for these values, line 65.

D. EXTERNAL INSERTION OF INPUT DATA

The reading of data into INPUT1 and INPUT2 arrays, lines 80 through 88, is optional and these statements may be left out of the GFP. If they are removed, then function keyboard key number one will have no
use and the INPUT2 data will have to be inserted from the alphanetic keyboard with the use of the cursor.
IV. SAMPLE GFP EXECUTION CHAIN OF EVENTS

This chapter describes one complete chain of events during execution of the GFP. This description will assume that all of the capabilities of the GFP will be utilized. It is to be remembered that the user is not required to follow this sequence exactly to utilize the GFP, and that multiple sequences of events may be utilized for multiple executions of the associated program. This chapter will also indicate faulty sequences which will prevent the user from obtaining his desired graphical display. Reference to the GFP program listing in appendix C will be made by line index number.

Upon initial execution of the GFP the INPUT1, INPUT2, and OUT1 arrays are blanked out, intermediate disc storage is defined and basic parameter variables are initialized, lines 32 through 50. The 2250 is initialized and the statement "GENERAL FORTRAN PROGRAM" is displayed in the center of IGDS1. A horizontal line is displayed across the bottom edge of IGDS2 and a diagonal line is displayed from the lower left corner to the upper right corner of IGDS3 for initialization. Data is read into INPUT1 and INPUT2 arrays. These reading operations are optional. The alarm, which is a short duration tone, is sounded. The keys 1 through 29 are enabled and the lights in these keys are lighted to indicate they are enabled. The program waits at this point for attention information provided by the user with the program function keyboard. Unless otherwise specified during the remainder of the discussion in this chapter, the program will return to line 92 and wait for the next attention information, at the completion of each operation.
A. DISPLAY OF INPUT1 ARRAY

The sequence of events for displaying INPUT1 information is listed below. A complete explanation of all of the computer operations for this process is described below the list.

1. Push key 1
2. Push one of keys 2 through 9
3. Repeat steps 1 and 2 as desired

Key 1 is pushed by the user and the first 25 lines of information, or the number specified in the variable NMAX, if it is less than 25, stored in INPUT1 array is displayed in IGDS1. The user then pushes any of keys 2 through 9 to reset the page to be displayed. The use of keys 2 through 9 resets the values of the variables NA and NB and the program goes to line 104 and disables all keys except 1, 2, and 3. The previous display will remain and the lights in the disabled keys will be turned off. Since the INPUT1 information is being displayed, key 1 will be pushed next and the new 25-line segment of INPUT1 information will be displayed. The program will go to line 92, enable and light all keys 1 through 29 again, and wait for the next attention information. It should be noted at this time that when only keys 1, 2, and 3 are enabled, any of these keys may be pushed. If key 2 is pushed, only the 20-line segment in the INPUT2 array with the first line corresponding to the present value of NA will be displayed. If key 3 is pressed, the 25-line segment in the OUT1 array corresponding to the present values of the variables NA and NB will be displayed. Therefore, the user must push the proper key to display the desired information. The procedure of displaying successive pages of INPUT1 information may be continued by repeating the two-step procedure described above. When the value of
the variable NA is incremented to a value greater than the specified value of NMAX, the program control is transferred to line 408 and the statement, "END OF INPUT/OUTPUT ARRAY USE PROGRAM FUNCTION KEYBOARD TO RETURN TO DESIRED PORTION OF PROGRAM," is displayed. Program control then returns to line 92 and waits for the next attention information.

B. DISPLAY, CHANGING AND STORING OF INPUT2 DATA

The next sequence is to display, change, and store INPUT2 data as follows:

1. Push key 10
2. Push key 12
3. Push key 13
4. Repeat steps 2 and 3 as desired
5. Push key 14
6. Push key 15
7. Push key 11 if additional pages of INPUT2 data are to be inserted
8. Push key 2
9. Repeat steps 2 through 6 as desired
10. Push key 16

The user pushes key 10 to display the first 20 lines, or the number of lines specified by the value of the variable NDATA if it is less than 20, in IGDS1. In addition, the statements, "TO MODIFY OR ENTER DATA INSERT CURSOR," and "REMOVE CURSOR WHEN EDITING IS COMPLETED," are displayed at the top and bottom respectively of IGDS1. If no INPUT2 data is read in initially, the IGDS1 area will be blank except for the above statements. By pushing key 12, the cursor will be displayed under the first character position in the first line of INPUT2 data. The use of key 13 advances the cursor one line. The user may change the input
data by proceeding to the desired line and, by use of the advance key on the alphameric keyboard, position the cursor under the desired character to be changed. After typing in the new character from the alphameric keyboard, the cursor will advance one space. If no initial INPUT2 data was read, the user will type in his complete page of data. Upon completion of the desired changes on a page of INPUT2 data, the user pushes key 14. This is not necessary but will prevent the user from inadvertently striking a key on the alphameric keyboard which would change the character where the cursor was positioned. The user then pushes key 15 which reads the displayed data into the INPUT2 array and sounds the alarm. This key must be pushed before the display is advanced or the changed data will not be entered into the INPUT2 array. If no changes had been made to the data previously read, this operation would not affect the data already in the array. The above sequence of operations may be repeated to change or insert additional pages of INPUT2 data by pushing key 11 which advances NA and NB by 20-line increments, pushing key 2, and making desired changes or insertions. If the value of the variable NA is increased to a value greater than the value of the variable NDATA, the end of input/output array message will be displayed in IGDS1. Upon the completion of all input changes or the typing of all INPUT2 data, key 16 is pushed to transfer this data to intermediate disc storage from which it may be utilized by the associated program. The alarm is again sounded to indicate completion of this operation.

Key 28 is pressed to execute the associated program. The user may utilize an input data checking procedure in his associated program which places a line of improper data into the OUT2 array and returns
control to GFP. If this checking procedure is used and the input data is in error, the program will return to line 432, print the following statements, "FOLLOWING DATA CARD CONTAINS ILLEGAL NAME," the line of improper data from the OUT2 array, and "USE PROGRAM FUNCTION KEYBOARD TO RETURN TO DESIRED PORTION OF PROGRAM," in IGDS1, and sound the alarm. If the data is correct or no data checking procedure is used and the associated program completes execution, the program will return to line 420, display the following statement in IGDS1, "ASSOCIATED PROGRAM EXECUTION COMPLETED USE PROGRAM FUNCTION KEYBOARD TO RETURN TO DESIRED PORTION OF PROGRAM," and sound the alarm.

C. DISPLAY OF OUT1 DATA

Upon completion of the execution of the associated program, the user may utilize the following sequence of keys to display OUT1 data.

1. Push key 18
2. Push key 17 if desired
3. Push key 19
4. Push one of keys 2 through 9
5. Push key 3
6. Repeat steps 4 and 5 as desired

The user pushes key 18 to transfer the alphameric output from the intermediate disc storage to the OUT1 array. The alarm will sound to indicate completion of this operation. This data may be displayed in IGDS1 in a similar manner as stated previously for the display of INPUT1 information. Key 19 is pushed to display the first 25 lines of data in the OUT1 array in IGDS1, or the value of the variable MAXOUT if it is less that 25. Any of keys 2 through 9 may now be pushed to reset the values of the variables NA and NB and will disable all
program function keys except 1, 2, and 3. Key 3 must be pushed next to display the desired segment of the OUT1 array. When the value of the variable NA is incremented to a value greater than the specified value of the variable MAXOUT, the end of input/output array message will be displayed in IGDS1. It should be noted that if key 17 is pushed after the completion of execution of the associated program and before pushing key 18, the alphanumer output data from that execution will be lost by the blanking out of that portion of the intermediate disc storage reserved for the output data. Therefore, key 17 should be used only after key 18 is pushed.

D. GRAPH LABELING

The user may label the graph to correspond to the associated program data by the following sequence.

1. Push key 20
2. Push key 21
3. Repeat step 2 as desired

Key 20 is pushed to display the line graph axes and general labeling. The labels consists of the following characters. Immediately to the left of the vertical axis is a column of 11 equally spaced zeros. To the left of the column of zeros are two lines with the words, "DEPENDENT" and "VARIABLE." Below the horizontal axis are two lines of characters; first is a row of 11 equally spaced zeros and below this line another line with the words, "INDEPENDENT VARIABLE." In addition to the display of axes and labeling, a cursor is inserted beneath the uppermost zero in the column of zeros. Key 21 may be pushed to advance the cursor to the next designated line. The specific labeling of the graph may be
accomplished through the use of the alphanemic keyboard, in the following sequence.

1. The 11 zeros to the left of the vertical axis are changed individually be selecting the desired characters from the keyboard to replace the zero indicated by the cursor. Key 21 is then pushed which places the cursor under the next zero. This procedure is repeated until all 11 zeros are changed to the desired values. Two characters may be placed on each of these lines.

2. At the completion of (1) push key 21, which places the cursor under the first zero below the horizontal axis, and change this entire line as desired to scale the horizontal axis.

3. At the completion of (2) push key 21, which places the cursor under the first character in the line "DEPENDENT", which is to the left of the vertical axis. The characters of this line may now be changed as desired.

4. At the completion of (3) push key 21, which places the cursor under the first character in the line "VARIABLE," which is directly under the line in (3) above. This line may now be changed as desired.

5. At the completion of (4) push key 21, which places the cursor under the first character in the line "INDEPENDENT VARIABLE" below the horizontal axis. This line may now be changed as desired.

The use of key 20 after the graph has been labeled by the user will reset the labels to the original characters displayed when key 20 was originally pushed.

E. DISPLAY OF LINE GRAPHS

The user may display a line graph in the following sequence.

1. Push key 22
2. Push key 23
3. Push key 24 if desired
4. Push key 22 if associated program is executed two or more times
5. Push key 23
6. Push key 25

Key 22 is pushed to remove the diagonal line from IGDS3. Key 23 is pushed to display line graph labeled "1" in IGDS3. If it is desired to save the data for this line graph for comparison with data from future executions of the associated program, key 24 must be pushed to transfer the data to the OUT5 and OUT6 arrays. The alarm will sound to indicate completion of this operation. The line graph labeled "2," using the data in OUT5 and OUT6 arrays may be displayed by using key 25.

F. PRINTING OF INPUT2 AND OUT1 ARRAYS ON LINE PRINTER

The data in INPUT2 and OUT1 arrays may be written on the line printer by pushing keys 26 and 27 respectively. The alarm will sound after each of these operations to indicate their completion.

This essentially completes the sequence of events in the execution of the GFP. If additional executions of the associated program are desired, the above sequences may be repeated.

When all desired operations are completed, the user will push key 29 which will completely terminate the execution of the GFP and the use of the 2250 for that program.

It should be remembered that this sample chain of events is only an aid to understanding the operations performed by the GFP. The user may utilize these operations in the order which satisfies his needs, limited by the few sequences indicated in this chapter which would prevent him from obtaining his desired output.
The GFP provides the user with an aid to utilizing the IBM 2250 Display Unit. The user may display a static array of information, enter, display and change input data to be utilized by the associated program and display alphameric and line graph data from the associated program. This provides the user with the ability to utilize the GFP with a wide range of associated programs and to obtain rapid display of computational results. In addition, the GFP may assist the inexperienced user to gain familiarity with the display unit and its capabilities.

It was noted during the development of this program that many variables and arrays had to be preset in the GFP and could not be changed during its execution. One possible solution to this problem may be to develop an initialization computer program which could read the desired GFP variable values and array dimensions as data, calculate necessary intermediate disc storage locations and then compile and execute the GFP. With this capability, the GFP would require no changes for individual associated programs and, in addition, could be modified to include additional variables which would automatically specify storage locations for the associated program. It is recommended that further study be conducted in this area to determine the feasibility of this type program.
APPENDIX A

FLOW CHART OF GFP

START

1. DIMENSION STATEMENTS
2. DEFINE FILE STATEMENT
3. SET NMAX, NCXDATA, MAXOUT, N1, GR1, GR2, GR3 AND GR4
4. INITIALIZE GRAPHICS
5. USER INPUT DATA FROM CARD READER?
6. READ INPUT1 AND INPUT2 DATA INTO ARRAYS?
7. DISPLAY INPUT1 ARRAY PAGE?
8. ADJUST INPUT1 DISPLAY?
9. ADVANCE/RESET INPUT1 ARRAY

DATA STATEMENT TO BLANK INPUT1, INPUT2, AND OUT1
TYPE AND COMMON STATEMENTS FOR ASSOCIATED PROGRAM
APPENDIX B

LIST OF VARIABLES USED BY GFP

A  Real variable used in conjunction with the variable B to alter the screen position to display a line of INPUT1 data in IGDS1.

B  Real variable, the value of which denotes the Y-coordinate of the initial starting position to display a line of information from INPUT1 in IGDS1.

C  Real variable, used in conjunction with the variable D to alter the screen position to display a line of INPUT2 data in IGDS1.

D  Real variable, the value of which denotes the Y-coordinate of the initial starting position to display a line of information from INPUT2 in IGDS1.

E  Real variable used in conjunction with the variable F to alter the screen position to display a line of OUT1 data in IGDS1.

F  Real variable, the value of which denotes the Y-coordinate of the initial starting position to display a line of information from OUT1 in IGDS1.

G  Real variable used in conjunction with the variable H to alter the screen position to display a zero on the vertical axis in IGDS2.

GRL  Real variable, the value of which denotes the X-coordinate of the lower left corner of IGDS3.
GR2  Real variable, the value of which denotes the Y-coordinate of the lower left corner of IGDS3.
GR3  Real variable, the value of which denotes the X-coordinate of the upper right corner of IGDS3.
GR4  Real variable, the value of which denotes the Y-coordinate of the upper right corner of IGDS3.
H    Real variable, the value of which denotes the Y-coordinate of the initial starting position to display a zero on the vertical axis in IGDS2.
I    Dummy integer variable for temporary storage throughout the program.
IATL Integer variable to which a value is assigned by the CRATL Subroutine to identify the attention level created by this call.
IBIT Variable identifying a ten-word array into which the RQATN Subroutine places additional information about light pen or end-of-order sequence attentions.
IBUTT Integer variable to which the RQATN Subroutine assigns a value to identify the source of the attention information being returned.
IGDS1 Integer variable to which a value is assigned by the INGDS Subroutine to identify the graphic data set to be created by this call.
IGDS2 Integer variable to which a value is assigned by the INGDS Subroutine to identify the graphic data set to be created by this call.
IGDS3  Integer variable to which a value is assigned by the INGDS Subroutine to identify the graphic data set to be created by this call.

IGSP  Integer variable to which a value is assigned by the INGSP Subroutine for initialization of the graphic subroutine package.

IT  Integer variable to which a value is assigned by the GSPRD Subroutine to identify the reason why the subroutine was terminated.

I2250  Integer variable to which a value is assigned by the INDEV Subroutine to identify the 2250 initialized by this call.

J  Dummy integer variable for temporary storage throughout the program.

K  Dummy variable for temporary storage.

L  Integer variable to which is assigned a value by GSP, to specify the key value in the PTEXT or GSPRD Subroutines, of a particular line of data displayed in IGDS1 or read into INPUT2 array.

M  Integer variable, the value of which is used by GSP to determine the particular line under which to display the cursor by the ICURS Subroutine.

MAXOUT  Integer variable, the value of which denotes the actual number of lines used in OUT1 array.

NA  Integer variable, the value of which denotes the first line in a sequence of information from a desired array, to be displayed in IGDS1.
NB  Integer variable, used in conjunction with NA and the value of which denotes the last line in a sequence of information from a desired array, to be displayed in IGDS1.

NDATA  Integer variable, the value of which denotes the actual number of lines used in INPUT2 array.

NMAX  Integer variable, the value of which denotes the actual number of lines used in INPUT1 array.

NL  Integer variable, the value of which denotes the number of lines drawn for each line graph in IGDS3.

N2  Integer variable, the value of which denotes the number of points utilized to draw a line graph in IGDS3.

X  Real variable, the value of which denotes the X-coordinate of a particular point for displaying a line graph in IGDS3.

Y  Real variable, the value of which denotes the Y-coordinate of a particular point for displaying a line graph in IGDS3.
APPENDIX C

GFP COMPUTER PROGRAM LISTING

This appendix provides a complete listing of the GFP and includes the example associated program.
GENERAL FORTRAN PROGRAM FOR USE WITH IBM 2250 DISPLAY UNIT

DIMENSION VARIABLES FOR GFP

DIMENSION null(1), input1(18,500), input2(18,500), out1(18,350), ibiti(1), ifkey(25), it(1), out2(18,1), out3(10), out4(10), out5(10), out6(10)

DIMENSION VARIABLES FOR ASSOCIATED PROGRAM

DIMENSION nmpac(50), nacept(50), mined(300), minenr(300), minex(300), mney(300), mneyr(300), mneyx(300), mwep(50), nsamsz(100), fsamsz(100), sample(100)

DECLARE INTEGER VARIABLES FOR ASSOCIATED PROGRAM

INTEGER z,w

DECLARE VARIABLES IN COMMON FOR ASSOCIATED PROGRAM

COMMON mswp1, mswp2, nsp1, nsp2, nsmp, nsamsz, fsamsz
COMMON xsmswp, xnsnp, nsafe
COMMON nair, nmpac, nacept, mined, nrminr, minex, mney, mneyr, mneyx, mwep, msrd, nsh, ns, nsxl, nchl, nker, nyfr, nyfr, nyfr, nyfr, nyfr, nyfr, nyfr, nyfr
COMMON node, nsc, z, w
COMMON out2, common dummy

BLANK INPUT1, INPUT2, AND OUT1 ARRAYS

DATA input1, input2, out1/999000*1, 999000*1, 63000*1 /

DEFINE INTERMEDIATE DISC STORAGE AND GO TO FIRST RECORD

DEFINE FILE 8(900,72,E,IV1)

FIND(8'1)

INITIALIZE LENGTH OF INPUT1, INPUT2, AND OUT1 ARRAYS, INITIALIZE, FOR GRAPH OUTPUT, THE NUMBER OF LINES TO BE DRAWN FOR ANY GRAPH, AND THE DATA RANGE OF THE GRAPHS

NMAX=475
NCATA=34
MAXOUT=350
N1=5
GR1=0.0
GR2=0.0
GR3=25.0
GR4=50.0

C INITIALIZE 2250 AND DISPLAY INITIAL MESSAGE

NULL(1)=-5
CALL INGSP(IGSP,NULL)
CALL INDEV(IGSP,10,12250)
CALL INGDS(12250,IGDS1,9816)
CALL SGDSL(IGDS1,0.0,0.0,10.0,10.0,0.0,0.0,10.0,0.0,10.0,0.0)
CALL SDATL(IGDS1,0.0,0.0,1000.0,500.0)
CALL INGDS(12250,IGDS2,512)
CALL SGDSL(IGDS2,0.0,0.0,10.0,10.0,0.0,0.0,10.0,0.0,10.0,0.0)
CALL SDATL(IGDS2,0.0,0.0,1000.0,1000.0)
CALL INGDS(12250,IGDS3,512)
CALL SGDSL(IGDS3,1000.0,500.0,700.0,980.0,0.0,0.0,0.0,1000.0,1000.0,0.0)
CALL SDATL(IGDS3,GR1,GR2,GR3,GR4)
CALL CPATL(12250,IATL)
CALL SCHAM(IGDS1,2)
CALL PTEXT(IGDS1,'GENERAL FORTRAN PROGRAM',23,1,NULL,1,300.0,300.0,1)

C READ ASSOCIATED PROGRAM INTO INPUT1 ARRAY FROM CARD READER

DO 32 I=1,NMAX
32 READ(5,33)(INPUT1(J,1),J=1,18)
33 FORMAT(18A4)

C READ ASSOCIATED PROGRAM DATA INTO INPUT2 ARRAY FROM CARD READER

DO 34 I=1,NDATA
34 READ(5,33)(INPUT2(J,1),J=1,18)
CALL SALRM(12250)

C ENABLE AND LIGHT ATTENTION SOURCES AND REQUEST ATTENTION INFORMATION

35 CALL ENATN(IATL,1,-29)
CALL MLTIA(12250,3)
CALL ROATN(IATL,IBUTT,2,IBIT,1,-29)

C BRANCH TO ATTENTION HANDLING ROUTINES
GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23)
1,24,25,26,27,28,29,IBUUTT

DISABLE AND TURN OFF LIGHTS ON ALL ATTENTION SOURCES EXCEPT 1-3
AND REQUEST ATTENTION INFORMATION

CALL DSATN(IATL,4,-29)
CALL MLFTS(I25C,3)
CALL ROATN(IATL,IBUUTT,2,IRIT,1,-3)

BRANCH TO ATTENTION HANDLING ROUTINES
GO TO(37,39,46),IBUUTT

DISPLAY INPUT1 DATA (FIRST 25 LINES OR ENTIRE ARRAY)

1 NA=1
NB=MINT(25,NMAX)

CALL RFSET(I2GDS)
CALL SCHAM(I2GDS,1)
IF(NB GT NMAX)NB=NMAX
IF(NA GT NMAX)GO TO 58
IF(NA LT 1) GO TO 1
IF(NA LT NMAX-24) NA=NA+24
IF(NA LT NMAX-24) NR=NR24
A=0.0
DO 38 I=NA,NB
A=A+20.0
R=500.0-A

CALL PTEXT(I2GDS,INPUT1(1,1),72,NULL,NULL,1,0,0,8)
CALL EXEC(I2GDS)
GO TO 35

ADVANCE INPUT/OUTPUT DISPLAY 10 PAGES (250 LINES)

2 NA=NA+250
NR=NB+250
GO TO 36

REWIND INPUT/OUTPUT DISPLAY 10 PAGES (250 LINES)

2 NA=NA-250
NB=NR-250
GO TO 36

ADVANCE INPUT/OUTPUT DISPLAY 1 PAGE (25 LINES)
4 NA=NA+25
NR=NR+25
GO TO 36

REWIND INPUT/OUTPUT DISPLAY 1 PAGE (25 LINES)

5 NA=NA-25
NR=NR-25
GO TO 36

ADVANCE INPUT/OUTPUT DISPLAY 10 LINES

6 NA=NA+10
NR=NR+10
GO TO 36

REWIND INPUT/OUTPUT DISPLAY 10 LINES

7 NA=NA-10
NR=NR-10
GO TO 36

ADVANCE INPUT/OUTPUT DISPLAY 5 LINES

8 NA=NA+5
NR=NR+5
GO TO 36

REWIND INPUT/OUTPUT DISPLAY 5 LINES

9 NA=NA-5
NR=NR-5
GO TO 36

DISPLAY ASSOCIATED PROGRAM INPUT2 DATA AND CURSOR INSTRUCTIONS

10 NA=1
NB=MINC (20,NDATA)

IF (NA.GT.NDATA) NA=NDATA
IF (NA.GT.NDATA) GO TO 58
IF (NA.LT.1) GO TO 10
IF (NA.LT.NDATA-19) NR=NA+19
IF (NA.GT.NDATA-19) NB=NDATA
CALL RESET (IGDS1)
CALL SCHAM (IGDS1,2)
CALL PTEXT (IGDS1,'TO MODIFY OR ENTER DATA INSERT CURSOR',37,1,NULL)
1,1,20,0,470,0)
CALL PTEXT (IGDS1,'REMOVE CURSOR WHEN EDITING IS COMPLETED',39,2,NULL)
CALL EXEC(IGDS1)
CALL SCHAM(IGDS1, 1)
C=0.0
L=0
DO 40 I=NA, NB
C=C+20.0
L=470.0-C
40 CALL PTEXT(IGDS1, INPUT2(1,1), 72, NULL, IKEY(L), 1, 0, 0, D)
CALL EXEC(IGDS1)
GO TO 35
C ADVANCE INPUT2 DATA DISPLAY 1 PAGE (20 LINES)
C 11 NA=NA+20
NB=NB+20
GO TO 36
C INSERT CURSOR UNDER FIRST LINE OF INPUT DATA
C 12 M=1
CALL ICURS(IGDS1, NULL, IKEY(M), 1)
GO TO 35
C ADVANCE CURSOR 1 LINE
C 13 CALL RCURS(IGDS1)
M=M+1
CALL ICURS(IGDS1, NULL, IKEY(M), 1)
GO TO 35
C REMOVE CURSOR
C 14 CALL RCURS(IGDS1)
GO TO 35
C READ GRAPHICS INTO INPUT2 DATA ARRAY
C 15 L=0
DO 41 M=NA, NB
L=L+1
41 CALL GSPOD(IGDS1, INPUT2(1, M), 72, I, NULL, IKEY(L))
CALL SALR(M(12250))
GO TO 35
C TRANSFER INPUT2 DATA TO INTERMEDIATE DISC STORAGE
16 DO 42 I=1,NDATA
42 WRITE(*',133)(INPUT2(J,1),J=1,18)
   FIND(8',1)
   CALL SALRM(I2250)
   GO TO 35
C
C BLANK OUT1 OUTPUT SECTION ON INTERMEDIATE DISC STORAGE AREA
C
17 DO 43 I=51,400
43 WRITE(*',144)
44 FORMAT(72H
   1)
   FIND(8',1)
   CALL SALRM(I2250)
   GO TO 35
C
C XFER ASSOCIATED PROGRAM OUTPUT TO OUTPUT DISPLAY ARRAY OUT1
C
18 K=0
19 DO 45 I=51,400
20 K=K+1
45 READ(8',1,33)(OUT1(J,K),J=1,18)
   CALL SALRM(I2250)
   GO TO 35
C
C DISPLAY ASSOCIATED PROGRAM OUT1 DATA
C
19 NA=1
20 NR=MINDO(25,MAXOUT)
21 IF(NA.GT.MAXOUT)NA=MAXOUT
22 IF(NA.LT.1)GO TO 19
23 IF(NA.LT.MAXOUT-24)NR=NA+24
24 IF(NA.GT.MAXOUT-24)NR=MAXOUT
25 CALL RESET(IGDS1)
26 CALL SCHAM(IGDS1,3)
27 E=0.0
28 L=0
29 DO 47 I=NA,NA
30 E=E+20.0
31 F=500.0-E
32 L=L+1
33 IF(E.GT.200.0)GO TO 47
34 IF(E.LT.0.0)GO TO 47
35 IF(E.GT.1000.0)GO TO 47
36 IF(E.LT.-1000.0)GO TO 47
37 CALL PTEXT(IGDS1,OUT1(1,1),72,NULL,IKEY(L),1,0,F)
38 CALL EXEC(IGDS1)
39 GO TO 35
C
C DISPLAY GRAPH AXES -- INSERT CURSOR TO LABEL GRAPH
C
20 CALL RESET(IGDS2)
CALL SCHAM(IGDS2, 3)
G=0.0
DO 48 I=1, 11
G=G+36.3
H=100.0*G
48 CALL PTEXT(IGDS2, 'O', 1, I, NULL, 1, 260.0, 0, H)
CALL PTEXT(IGDS2, 'O', 0, 0, 0, 0, 0, 0, 31, 12, NULL, 1, 30, 0)
CALL PTEXT(IGDS2, 'RE', 9, 13, NULL, 1, 10.0, 0.800, 0)
CALL PTEXT(IGDS2, 'VARIABLE', 9, 14, NULL, 1, 10.0, 0.750, 0)
CALL PTEXT(IGDS2, 'INDEPENDENT', 20.15, NULL, 1, 400.0, 0.525, 0)
CALL MVPPOS(IGDS2, 0.0, 500.0)
CALL PLINE(IGDS2, 1000.0, 500.0)
CALL MVPPOS(IGDS2, 700.0, 580.0)
CALL PLINE(IGDS2, 300.0, 580.0)
CALL PLINE(IGDS2, 300.0, 980.0)
CALL EXEC(IGDS2)
M=1
CALL ICURS(IGDS2, M, NULL, 1)
GO TO 35

C ADVANCE CURSOR 1 LINE (TO CHANGE GRAPH LABELING)
C
21 CALL RECurs(IGDS2)
M=M+1
CALL ICURS(IGDS2, M, NULL, 1)
GO TO 35
C BLANK OUT IGDS3

C 22 CALL RESET(IGDS3)
GO TO 35
C DISPLAY LINE GRAPH 1
C
23 X=OUT4(1)
Y=OUT3(1)
CALL STPOS(IGDS3, X, Y)
N2=N1+1
DO 49 I=2, N2
X=OUT4(I)
Y=OUT3(I)
49 CALL PLINE(IGDS3, X, Y)
CALL PTEXT(IGDS3, 'I', 1, 1, NULL, 1, X, Y)
CALL EXEC(IGDS3)
GO TO 35
C
C TRANSFER OUT3/OUT4 DATA TO OUT5/OUT6
24 N2=N1+1
DO 50 I=1,N2
OUT5(I)=OUT3(I)
50 OUT6(I)=OUT4(I)
CALL SALRM(I2250)
GO TO 35

C DISPLAY LINE GRAPH
25 X=OUT6(I)
Y=OUT5(I)
CALL STPOS(IGDS3,X,Y)
N2=N1+1
DO 51 I=2,N2
X=OUT6(I)
Y=OUT5(I)
51 CALL PLINE(IGDS3,X,Y)
CALL PTEXT(IGDS3,'2',1,2,NULL,1,X,Y)
CALL EXEC(IGDS3)
GO TO 35

C WRITE CURRENT CONTENTS OF INPUT2 ARRAY ON LINE PRINTER
26 DO 52 I=1,NDATA
52 WRITE(6,53)(INPUT2(J,I),J=1,18)
53 FORMAT(' ',I8.4)
CALL SALRM(I2250)
GO TO 35

C WRITE CURRENT CONTENTS OF OUT1 ARRAY ON LINE PRINTER
27 DO 54 J=1,MAXOUT
54 WRITE(6,53)(OUT1(J,I),J=1,18)
CALL SALRM(I2250)
GO TO 35

C INSERT USER ASSOCIATED PROGRAM
PROGRAM MSF
2R Z=1
W=51
IDUM=0
55 CALL RNG(0,RN)
CALL INPUT(55,57)
CALL RNG(NRND,RN)
CALL TITLE
CALL MS2
IDUM=IDUM+1
OUT3(IDUM)=DUMY
OUT4(IDUM)=NMSP
GO TO 55

END OF ASSOCIATED PROGRAM

RETURN TO GFP AT COMPLETION OF EXECUTION OF ASSOCIATED PROGRAM
56 GO TO 59

RETURN TO GFP IF ILLEGAL DATA CARD ENCOUNTERED
57 GO TO 60

TERMINATE PROGRAM
29 CALL TMGSP(IGSP)
CALL EXIT

DISPLAY END OF INPUT/OUTPUT ARRAY MESSAGE
58 CALL RESET(IGDS1)
CALL SCHAM(IGDS1,1)
CALL PTEXT(IGDS1,'END OF INPUT/OUTPUT ARRAY',25,1,NULL,1,10C,0,400)
CALL PTEXT(IGDS1,'DISPLAY END OF INPUT/OUTPUT ARRAY MESSAGE',25,1,NULL,1,10C,0,400)
CALL EXEC(IGDS1)
GO TO 35

DISPLAY ASSOCIATED PROGRAM EXECUTION COMPLETED MESSAGE
59 CALL RESET(IGDS1)
CALL SCHAM(IGDS1,1)
CALL PTEXT(IGDS1,'ASSOCIATED PROGRAM EXECUTION COMPLETED',38,1,NULL,1,50,0,400)
CALL PTEXT(IGDS1,'DISPLAY ASSOCIATED PROGRAM EXECUTION COMPLETED MESSAGE',38,1,NULL,1,50,0,400)
CALL EXEC(IGDS1)
CALL SALRM(12250)
GO TO 35

DISPLAY ILLEGAL DATA MESSAGE
60 CALL RESET(IGDS1)
CALL SCHAM(IGDS1,1)
CALL PTEXT(IGDS1,'FOLLOWING DATA CARD CONTAINS ILLEGAL NAME',41,1,GFP00433)
CALL PTEXT(IGDS1, "OUT? (1,1), 72, 2, NULL, 1, 10, 0, 300, 0")
CALL PTEXT(IGDS1, 'USE PROGRAM FUNCTION KEYBOARD TO RETURN TO DES')
1END PORTION OF PROGRAM, 69, 3, NULL, 1, 20, 7, 200, 0
CALL EXEC(IGDS1)
CALL SARM(12250)
GO TO 35
END

C
INSERT SUBROUTINES, IF ANY, FOR ASSOCIATED PROGRAM

SUBROUTINE RNG(N,RN)
NR=N
IF(NR)10,10,20
10 IX=12133
NR=NR+1
20 DO 50 I=1,NR
50 IY=IX*65539
IF(IY)5,5,6
IF(IX)5,6,6
5 IY=IY+2147483647+1
6 RN=IY
RX=R(N*R+4656613E-9
50 IX=IY
RETURN
END

C
SUBROUTINE INPUT(*,*)
DIMENSION NRM(50), NACEPT(50), MINEQ(300), MINFNR(300), MINEX(300),
1 MINEY(300), MWEF(50), NSAMSIZ(100), FSAMSIZ(100), Sample(100), OUT2(18, 1)
C
INTEGER NAME, A/4HNAIR/, B/4HNMP/, C/4HMWFP/, D/4HMSWD/, E/4HSHWGF.
F/4HSHWGF.
17/F/4HERRR/, G/4HERRG/, H/4HNSMP/, X/4HCHW/, J/4HCHL/, K/4HERUN/,
2L/4HHRND/, M/4HNDCS/, N/4HNSH/, P/4HPSH/, Q/4HPSH/, R/4HPSH/, S/4HPSH/,
3V, W, X, Y, Z
COMMON MSWP1, MSWP2, NSP1, NSP2, NSMP, NSAMSIZ, FSAMSIZ
COMMON XMSWP1, XNSP, XNSAF
COMMON NAIR, NRM(50), NACEPT, MINEQ, MINFNR, MINEX, MINFNR, MINEX, MINEY, MNP,
1 MWEF, MNSD, NRSH, NNSW, NSXL, NCHL, NXERR1, NYERR1, NXERRS, NYERRS,
2NCHANL, NCHANL, FSAS7
COMMON NRNC, NXIN, NRND
COMMON NDSC, Z, W
C
GFP00423
GFP00424
GFP00425
GFP00426
GFP00427
GFP00428
GFP00429
GFP00440
GFP00441
GFP00442
GFP00443
GFP00444
GFP00445
GFP00446
GFP00447
GFP00448
GFP00449
GFP00450
GFP00451
GFP00452
GFP00453
GFP00454
GFP00455
GFP00456
GFP00457
GFP00458
GFP00459
GFP00460
GFP00461
GFP00442
GFP00443
GFP00444
GFP00445
GFP00446
GFP00447
GFP00448
GFP00449
GFP00450
GFP00451
GFP00452
GFP00453
GFP00454
GFP00455
GFP00456
GFP00457
GFP00458
GFP00459
GFP00460
GFP00461
GFP00442
GFP00443
GFP00444
GFP00445
GFP00446
GFP00447
GFP00448
GFP00449
GFP00450
GFP00451
GFP00452
GFP00453
GFP00454
GFP00455
GFP00456
GFP00457
GFP00458
GFP00459
GFP00460
GFP00461
CCMONG OUT
10 READ(B'7,1000)NAME,N1,N2
1000 FORMAT(A4,?16)
C
   IF(NAME='A') 40,20,40
20 NAIK=N1
   READ(B'7,1001)(NRMPAC(I),NACPT(I),I=1,NAIK)
1001 FORMAT(12?16)
   Y=(NAIR-1)/12+1
   Z=Z+Y
   MRMIN=0
   DO 30 I=1,NAIR
30 NRMLINE=NRMLINE+NRMPAC(I)
   READ(B'7,1001)(MINED(I),I=1,NRMLINE)
   Y=(NRMLINE-1)/12+1
   Z=Z+Y
   GO TO 10
C
40 IF (NAME='B') 52,50,52
50 NMSM=N1
   GO TO 10
C
52 IF (NAME='C') 58,54,58
54 READ(B'7,1001)(MWEPT(I),I=1,NMSM)
   Y=(NMSM-1)/12+1
   Z=Z+Y
   GO TO 10
C
58 IF (NAME='D') 70,60,70
60 MSWD=N1
   GO TO 10
C
70 IF (NAME='E') 90,80,90
80 NSW=N1
   NSXL=N2
   GO TO 10
C
90 IF (NAME='F') 110,100,110
100 NXPFR1=N1
   NXFRS=N2
   GO TO 10
C
110 IF (NAME='G') 130,120,130
120 NYEFR1=N1
   NYFRS=N2
   GO TO 10
C
130 IF (NAME-H) 160,140,160
140 NSMP=N1
READ(8,Z,1001)(NSAMSZ(I),I=1,NSMP)
150 ESAMSZ(I)=NSAMSZ(I)

C
160 IF (NAME-X) 180,170,180
170 NCHW=N1
NCHANW=NCHW+1
GO TO 10

C
180 IF (NAME-J) 200,190,200
190 NCHL=N1
NCHANL=Nchl+1

GO TO 10

C
200 IF (NAME-K) 220,210,220
210 RETURN 1

C
220 IF (NAME-L) 235,230,235
230 NRND=N1

GO TO 10

C
235 IF (NAME-M) 240,236,240
236 NOSC=N1

GO TO 10

C
240 IF (NAME-N) 260,250,260
250 NRSH=N1

GO TO 10

C
260 IF (NAME-O) 280,270,280
270 MSWP1=N1
MSWP2=N2
TEMP1=N1
TEMP2=N2
275 XMSWP=TEMP1/TEMP2

GO TO 10

C
280 IF (NAME-P) 300,290,300
290 NSP1=N1
NSP2=N2
TEMP3=N1
TEMP4=N2
295 XNSP=TEMP3/TEMP4
GO TO 10
C 300 IF (NAME=0) 310, 500, 310
310 WRITE(8*401, 1002) NAME, N1, N2
1002 FORMAT(12H A4, 16, 16, 1H)
1 READ(8*401, 1000) (OUT2(I, 1), I=1, 18)
1003 FORMAT(18A4)
RETURN 2
C 500 RETURN
END
C
SUBROUTINE TITLE
DIMENSION NRMPAC(50), NACEPT(50), MINED(300), MINENR(300), MIXEY(300),
MINEY(300), MWEPL(50), NSAMSZ(100), FSAMSZ(100), SAMPLE(100)
C COMMON MSWP1, MSWP2, NSP1, NSP2, NSMP, NSAMSZ, FSAMSZ
COMMON XMSPW, XNSP, XSAFE
COMMON NAI, NRMPAC, NACEPT, MINED, NRINE, MINENR, MIXEY, MINEY, NSMP,
MWEPL, NSW, NSH, NSW, NSXL, NCH, NCHL, NXERR1, NXR2, NXR1, NXR2, NXR, NXR
2 NCHANW, NCHANL, FSUP
COMMON NRNC, NXTN, NRRD
COMMON NDSZ, 7, W
C INTEGER Z, W
WRITE(8*W, 2000)
2000 FORMAT(72H)
1 W=W+1
WRITE(8*W, 2001)
2001 FORMAT(72H A PROBLEM IN MINE FIELD SIMULATION)
1 W=W+1
WRITE(8*W, 2000)
W=W+1
WRITE(8*W, 2002) NRSH, NRME
2002 FORMAT(24H NUMBER OF SHIPS SCORED (13, 26H SHIPS THROUGH A FIELD)
1 OF 13, 12H MINES ))
W=W+1
WRITE(8*W, 2000)
W=W+1
WRITE(8*W, 2003) NSP1, NSW, NSP
2003 FORMAT(5H, 12, 61H MINE SWEEPER PASSES)
1 WEEPER WIDTH = [3, 1H ]
C
2004 FORMAT(52H 
PROBABILITY OF MINE SWEEPER ELIMINATING MINE = I4,11H )
W=W+1
WRITE(8, W, 2004)
2005 FORMAT(72H 
MINE SWEEPER ENTRY POINTS )
1 W=W+1
WRITE(8, W, 2000)
2006 FORMAT(3(6H 
(1016,6H 
/))
K=(NMSP-1)/10+1
W=W+K
WRITE(8, W, 2000)
2007 FORMAT(72H 
DELIVERY VEHICLE ENTRY ERROR )
1 W=W+1
WRITE(8, W, 2000)
2008 FORMAT(37H 
X-COORDINATE I4, 2CH
W=W+1
WRITE(8, W, 2009)
2009 FORMAT(37H 
Y-COORDINATE I4, 2CH
W=W+1
WRITE(8, W, 2009)
2010 FORMAT(72H 
DELIVERY VEHICLE NUMBER ENTRY MINE DIST )
1 W=W+1
WRITE(8, W, 2010)
2011 FORMAT(72H 
NUMBER OF MINES POINT )
1 W=W+1
WRITE(8, W, 2011)
J1=1
J2=J1+NRMPAC(I)-1
J3=MINS(J1+4,J2)
WRITE(RW,2012)I, NRMPAC(I), NACEPT(I), (MINED(J), J=J1, J3)
2012 FORMAT(9H,12,12H,12,5H,14,2H,17,17,17,17,17,17,17,17,
17,1H)
W=W+1
IF(J2-J3)10,10,0
J3=J3+1
WRITE(RW,2013)(MINFD(J), J=J3, J2)
2013 FORMAT(36H,17,17,17,17,17,17,1H)
W=W+1
10 J1=J2+1
WRITE(RW,2000)
W=W+1
WRITE(RW,2000)
W=W+1
WRITE(RW,2014)NSW
2014 FORMAT(28H, EFFECTIVE SHIP WIDTH = I3)
W=W+1
WRITE(RW,2015)NSWL
2015 FORMAT(56H, MINIMUM DISTANCE BETWEEN CHANNEL AND SHIP CENTER = I13,13H)
W=W+1
WRITE(RW,2016)NSP1,NSP2
2016 FORMAT(46H, PROBABILITY OF A SHIP EXPLODING A MINE = I4,1H/I4,1)
W=W+1
WRITE(RW,2017)NCHW,NCHL
2017 FORMAT(21H, CHANNEL WIDTH = I5,27H)
W=W+1
WRITE(RW,2000)
W=W+1
WRITE(RW,2000)
W=W+1
WRITE(RW,2018)
2018 FORMAT(77H, SAMPLE SAMPLE SAMPLE SAMPLE SAMPLE SAMPLE MINIMUM)
1 MAXIMUM )
W=W+1
WRITE(RW,2019)
2019 FORMAT(72H, NUMBER SIZE MEAN VARIANCE SHIPS SAFE)
W=W+1
WRITE(RW,2000)
W=W+1
WRITE(RW,2000)
SUBROUTINE MS2
FORTRAN VERSION OF MINESWEEP PROR.

DIMENSION NRMPAC(50),NANCEP(50),MINED(300),MINENN(300),MINEX(300),
MINEY(300),MWEF(50),NSAMZ(100),FSAMZ(100),SAMPLE(100),OUT2(18,1)

INTEGER Z,W

COMMON MSWP1,MSWP2,NSP1,NSP2,NSMP,NSAMZ,FSAMZ
COMMON XMSWP,XNSP,NSAFE
COMMON NANCEP,NRMPAC,NANCEP,MINED,NRMIN,NMIN,NINX,MINEY,MINNY,NMS

NCHNW,NCHNL,FSPSE
COMMON NANCEP,NINX,NSMP
COMMON NSAMP,NCHL,NCHW,NSAMZ,NXERR1,NXERR2,NSAMZ,FSAMZ

NCHNW=NCHW+1
NCHWL=NCHW+1

DUMMY=0.0
DO 40 I=1,NSMP
MING=1000
MAXG=0
NOUT1=1
NOUT2=NSAMZ(I)
FSPSM=0
DO 20 J=1,NOUT2
CALL MINSIM
FSPSE=NSAFE
SAMPLE(J)=FSPSE
MING=MIN(MING,NSAFE)
MAXG=MAX(MAXG,NSAFE)

FSPSM=FSPSM+FSPSE
OUT30=FSPSM/NSAMZ(I)
DUMY1=DUMY1+OUT30
SAMAVE=OUT30
STORE=0
DO 30 K=1,NOUT2
TEMP=SMPE(K)-SAMEV
30 STORE=STORE+TEMP**2
OUT40=STORE/(FSAMSZ(I)-1.)
WRITE(6,W+1,2001)NOUT1,NOUT2,OUT40,MING,MAXG
2001 FORMAT(7H 13,8H I3,5H F6,2,5H F7,4,8H
13,11H I3,4H )
40 W=W+1
DUMMY=DUMY1/NSMP
RETURN
END

SUBROUTINE MINSIM

C DIMENSION NRMPAC(50),NACEPT(50),MINED(300),MINENR(300),MINEX(300)
1 MINEXY(300),MWEP(50),NSAMSZ(100),FSAMSZ(100),SAMPLE(100)
C COMMON MSWP1,MSWP2,NSP1,NSP2,NSMP,NSAMSZ,FSAMSZ
COMMON XMSWP,XNSP,NSAFE
COMMON NAIR,MRMPAC,NACEPT,MINED,MRMINE,MINENR,MINEX,MINET,NSW,
1 MWEP,NSW,NSX,NSW,NSXL,NCHW,NCHL,NXERR1,NYERR1,NXERRS,NYERRS,
2 NCHANW,NCHANL,FSPSE
COMMON NNR,NNX,NNRND
COMMON NSNC
NULL=2147483647
DO 10 I=1,NRMR
MINENR(I)=0
MINEXY(I)=NULL
MINY(I)=NULL
J=1
DO 20 I=1,NAIR
C ALL RNG(NDNS,RN)
NXIN=NYERR1*RN
MINEXY(I)=NXIN+MINED(J)-NYERR1/2
20 CALL RNG(NDNS,RN)
NXIN=NYERR1*RN
MINEXY(I)=NXIN+MACEPT(I)-NXERR1/2
MINENR(I)=J
20 J=J+MRMPAC(I)
DO 30 J=1,NRMR
21 IF (MINEXY(I)=NULL) 30,22,30
22 CALL RNG(NDNS,RN)
NXIN=NYERRS*RN
MINEXY(I)=NXIN+MINED(J)+MINEXY(J-1)-NYERRS/2
23 I=I+1
30 CONTINUE
C
CALL RNG(NRDSC,0N)
NXIN=NXFRS*0N
MINEX(J)=NXIN+MINEX(J-1)-NXFRS/2
MINFR(J)=J
30 CONTINUE

LIMIT
DO 40 I=1, NRMINE
IF (MINEX(I)) 39, 38, 38
38 IF (MINEX(I) 36, 37, 37
39 MINEX(I) = NULL-7
MINEX(I) = MINEX(I)
37 IF (MINEX(I)-NCHANL) 31, 33, 33
31 IF (MINEX(I)-NCHANW) 40, 33, 33
33 MINEX(I)=NULL-1
MINEX(I)=MINEX(I)
40 CONTINUE

SORT MINE TABLE
J=NRMINE
50 K=J
J=J-1
DO 60 I=1, J1
IF (MINEX(I)-MINEX(K)) 55, 60, 60
55 K=I
60 CONTINUE
NTI=MINEFR(J)
MINEFR(J)=MINEFR(K)
MINEFR(K)=NTI
NTI=MINEX(J)
MINEX(J)=MINEX(K)
MINEX(K)=NTI
NTI=MINEY(J)
MINEY(J)=MINEY(K)
MINEY(K)=NTI
J=J-1
IF (J-1) 70, 70, 60

MINE SWEEPER PASS
70 IF (NMSP) 140, 140, 85
65 DO 130 I=1, NMSP
NLOLIM=MREV(N)-MSWD/2-1
NUPLIM=NLOLIM+MSWD+2
DO 120 J=1, NRMINE
IF (NLOLIM-MINEX(J)) 90, 120, 120

GFP00807
GFP00808
GFP00809
GFP00810
GFP00811
GFP00812
GFP00813
GFP00814
GFP00815
GFP00816
GFP00817
GFP00818
GFP00819
GFP00820
GFP00821
GFP00822
GFP00823
GFP00824
GFP00825
GFP00826
GFP00827
GFP00828
GFP00829
GFP00830
GFP00831
GFP00832
GFP00833
GFP00834
GFP00835
GFP00836
GFP00837
GFP00838
GFP00839
GFP00840
GFP00841
GFP00842
GFP00843
GFP00844
GFP00845
GFP00846
GFP00847
GFP00848
GFP00849
GFP00850
GFP00851
GFP00852
GFP00853
GFP00854
C
C
C
C
C
C
C

90 IF (MINEX(J)-NUPLIM) 100,120,120
100 CALL RNG(NDSC,RN)
IF (RN-XMSWP) 110,120,120
110 MINEX(J)=NULL-2
MINEY(J)=MINEX(J)
120 CONTINUE
130 CONTINUE
C
SHIP START
C
NSAFE=0
NSUNK=0
NRNC=NCHW-NSXL*2
DO 190 I=1,NRSH
CALL RNG(NDSC,RN)
NXIN=RN*NRNC
NTI=NXIN
CALL RNG(NDSC,RN)
NXIN=RN*NRNC
NLOLIM=(NXIN+NTI)/2+NSXL-NSW/2-1
NUPLIM=NLOLIM+NSW+2
DO 185 N=1,NPMINE
IF (NLOLIM-MINFX(N)) 150,185,185
150 IF (MINEX(NI)-NUPLIM) 160,185,185
160 CALL RNG(NDSC,RN)
IF (RN-XNSP) 170,180,180
170 MINEX(N)=NULL-3
MINEY(N)=MINEX(N)
NSUNK=NSUNK+1
GO TO 190
180 MINEX(N)=NULL-4
MINEY(N)=MINEX(N)
185 CONTINUE
C
NSAFE=NSAFE+1
190 CONTINUE
200 RETURN
END
LIST OF REFERENCES


<table>
<thead>
<tr>
<th>No.</th>
<th>Copies</th>
<th>Initial Distribution List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td></td>
<td>Defense Documentation Center</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cameron Station</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Alexandria, Virginia 22314</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td>Library, Code 0212</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monterey, California 93940</td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td>Chief of Naval Operations (OP 96)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Department of the Navy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Washington, D. C. 20350</td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td>Department of Operations Analysis, Code 55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monterey, California 93940</td>
</tr>
<tr>
<td>5.</td>
<td></td>
<td>Asst. Professor A. F. Andrus, Code 55 As</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Department of Operations Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monterey, California 93940</td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td>LCDR Harlan L. Morrison, USN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18817 Harman Avenue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Melvindale, Michigan 48122</td>
</tr>
</tbody>
</table>
A General FORTRAN Computer Program for Use with the IBM 2250 Display Unit

Master's Thesis; April, 1970

Harlan Langtry Morrison

April, 1970

60

3

Naval Postgraduate School
Monterey, California 93940

A General computer program was designed, using the FORTRAN IV language, to utilize the IBM 2250 Display Unit in conjunction with the IBM System/360 Operating System. The user may adapt an associated program so that it may be an integral part of this general program. The general program provides the user with the ability to display a static information array and an input data array. The user may insert input data, for his associated program, from on-line input devices or directly through the IBM 2250 alphanemic keyboard and may change this data from the keyboard. The associated program may be executed repeatedly, subject to system time constraints, and both alphanemic and graphical output data from the associated program may be displayed after each execution.
### Key Words

<table>
<thead>
<tr>
<th></th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
<td>WT</td>
</tr>
</tbody>
</table>

1. General Fortran Program
2. GFP
3. IBM 2250 Display Unit
A General FORTRAN computer program for use with the IBM 2250 Display Unit.