SOFTWARE ENGINEERING LABORATORY SERIES

SEL-86-003

FLIGHT DYNAMICS SYSTEM SOFTWARE DEVELOPMENT ENVIRONMENT (FDS/SDE) TUTORIAL

JULY 1986

Goddar
Greenb

SOFTWARE ENGINEERING LABORATORY SERIES SEL-86-003

FLIGHT DYNAMICS SYSTEM SOFTWARE DEVELOPMENT ENVIRONMENT (FDS/SDE) TUTORIAL

JULY 1986

NASA
National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

FOREWORD

The Software Engineering Laboratory (SEL) is an organization sponsored by the National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC) and created for the purpose of investigating the effectiveness of software engineering technologies when applied to the development of applications software. The SEL was created in 1977 and has three primary organizational members:

NASA/GSFC (Systems Development and Analysis Branch)
The University of Maryland (Computer Sciences Department)
Computer Sciences Corporation (Flight Systems Operation)

The goals of the SEL are (1) to understand the software development process in the GSFC environment; (2) to measure the effect of various methodologies, tools, and models on this process; and (3) to identify and then to apply successful development practices. The activities, findings, and recommendations of the SEL are recorded in the Software Engineering Laboratory Series, a continuing series of reports that includes this document.

The primary contributors to this document are

John Buell (Computer Sciences Corporation)
Philip Myers (Computer Sciences Corporation)

Single copies of this document can be obtained by writing to

Frank E. McGarry
Code 552
NASA/GSFC
Greenbelt, Maryland 20771

ABSTRACT

A sample development scenario using the Flight Dynamics System Software Development Environment (FDS/SDE) is presented. The SDE uses a menu-driven, fill-in-the-blanks format that provides online help at all steps, thus eliminating lengthy training and allowing immediate use of this new software development tool.

TABLE OF CONTENTS

SECTION 1 - INTRODUCTION

The Flight Dynamics System Software Development Environment (FDS/SDE) provides an exciting new method for developing software. It uses a menu-driven, fill-in-the-blanks format that permits the developer to input, edit, compile, link, and execute software. Online help is always available, thus eliminating a lengthy training period and allowing immediate use of this new tool.

This tutorial demonstrates the use of the SDE following this simple eight-step scenario:

1. A software developer logs on to the IBM timesharing system (TSO) and enters the SDE (pages 3-2 through 3-14).

2. An IBM job card is formatted (pages 3-15 through 3-21).

3. A main program and two subroutines are entered into an existing PANVALET library (pages 3-22 through 3-49).

4. The program and subroutines are compiled with the IBM VS FORTRAN compiler to produce object modules (pages 3-50 through 3-74).

5. The object modules are merged together (link edited) into an executable program (load module) (pages 3-75 through 3-88).

6. The executable program is run (pages 3-89 through 3-97).

7. All listing files are sent to a printer for output (pages 3-98 through 3-105).

8. The developer updates the SDE analysis log, exits the SDE, and logs off TSO (pages 3-106 through 3-109).

At the beginning of the tutorial, each step of the developer's actions is shown on a separate page to highlight the specific action being taken. Later, all actions taken by a developer before pressing the <enter> key are shown on the same page. Shaded areas denote the changes from panel to panel (display to display). The underscore character in some of the shaded areas is not typed in; it simply marks the current cursor position as it actually appears on the screen.

The SDE has many more functions that aid the developer to create programs. All the capabilities of the Interactive System Productivity Facility (ISPF) are detailed in Reference 1. Reference 2 explains the use of all SDE functions that have been added to the basic ISPF.

SECTION 2 - BASIC SDE TERMS AND CONCEPTS

Before beginning the tutorial, a few basic terms and concepts of ISPF processing must be explained. The terms--panels, commands, help, and messages--and the concepts--levels, navigation, and jumps--are used extensively in discussions of ISPF and the SDE. Brief descriptions of the 327x keyboard and display screen and the ISPF editor (used in the scenario to create FORTRAN source modules) are also presented.

2.1 SDE TERMS

● Panels--Four basic types of panels are used in the tutorial, although more are available on the ISPF.

- Selection panels display a list of options in a menu format along with an associated option code. The selected option code is entered in a field near the top of the screen that displays OPTION===>.
- Parameter entry panels are reachea after selections have been made. Parameters for a specific function, such as file names or listing options, are entered on these panels.
- Data entry panels are used almost exclusively by the editor. They allow data to be input to the computer.
- Help panels provide instructions on how to use a specific function.

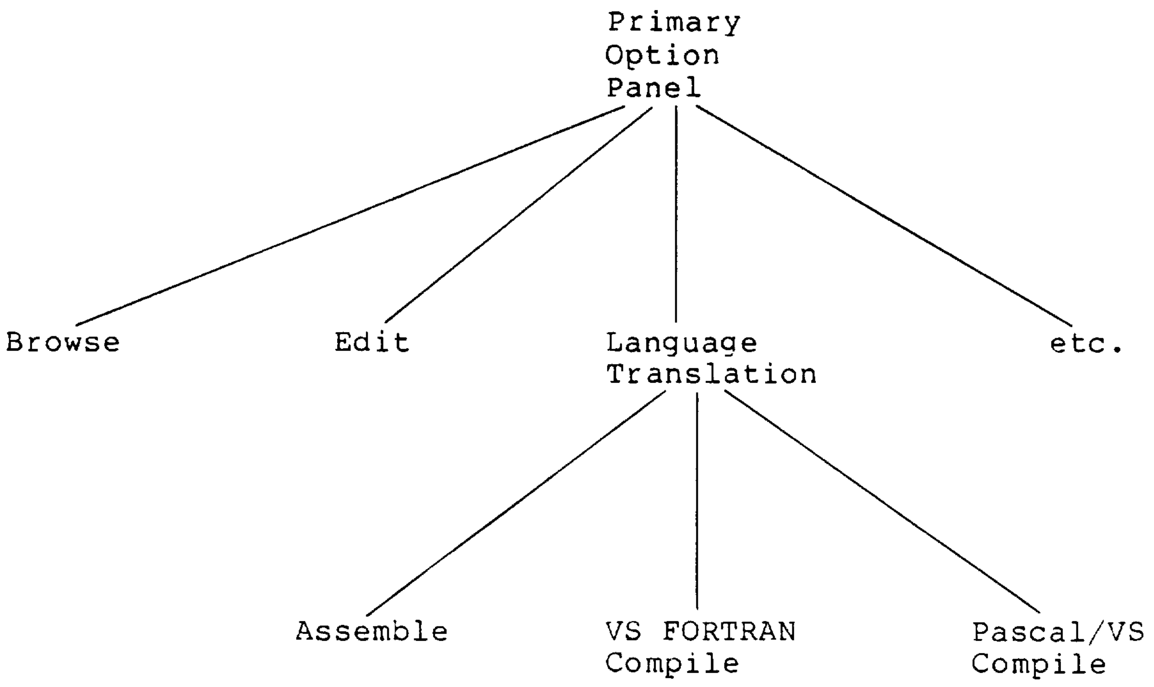
● Commands--ISPF and SDE commands (print, down, up, etc.) can be entered in the parameter entry field next to the COMMAND===> or OPTION===> prompts. The ISPF tutorial (Option 8.T) explains all commands.

- Help--Help is available online, pertinent to whatever panel is being displayed, whenever needed. Help is obtained by pressing program function key 1 <PF1> or by typing "help" next to the COMMAND===> or OPTION===> prompt.

- Messages--Messages are displayed in short and/or long format. The short format always appears in the upper right corner of the display. A long message, which further explains the short message, is displayed when <PF1> is pressed. When a short message is being displayed, help is obtained by pressing <PF1> twice (once to get the long message, once more for the help panel).

2.2 SDE CONCEPTS

- Levels--The SDE may be visualized as a tree-structured environment. The primary panel is the trunk, which has major branches. Each branch can in turn have more branches, depending on how many subselections are available. For example,



- Navigation--Navigation through the SDE (traversing the tree) is achieved by selecting codes displayed next to the various options on each selection panel.

- Jumps--Random access to option nodes can be achieved by entering an equal sign (=) and an option sequence in any parameter entry field, bypassing the intermediate option selections. This capability is demonstrated in scenario step 5.

2.3 327x KEYBOARD AND DISPLAY SCREEN

The keys on the 327x-type terminals are used to facilitate the use of ISPF and SDE functions. Table 2-1 describes the hardware keys used on the 327x keyboard. Table 2-2 describes the default program function (PF) keys used by the ISPF, as defined for this tutorial; they can be changed using option 0. The text symbol is used throughout the scenario to specify a particular key. The keyboard symbol varies slightly among 327x terminals made by different manufacturers.

The displays presented in the sample development scenario in Section 3 are surrounded by a facsimile of a 327x display screen. The lower portion of each display is the 327x system status area. Table 2-3 describes the symbols that may be displayed in this status area throughout the scenario.

2.4 ISPF EDITOR

The sample development scenario uses the ISPF editor to create FORTRAN source modules. Before beginning the scenario, a few basic editor commands should be explained. The ISPF editor is designed to take advantage of the capabilities of the 327x-type display devices. These devices transmit and receive data in block mode, meaning that data are transmitted to and from the computer in large, screen-size blocks.

Table 2-1. Hardware Keys

Key Name	Keyboard Symbol	Text Symbol	Purpose
Alt	ALT	<alt>	Used to perform any function illustrated on the front of keys
Backward tab	⇤	<BT>	Moves the cursor left to the previous entry field
Forward tab	⇥	<FT>	Moves the cursor right to the next entry field
New line	↵	<NL>	Moves the cursor down to the next entry field
Home	↖	<home>	Moves the cursor to the first data entry field on the top left (alternate function of forward tab)
Cursor select	CURSOR SEL	<csel>	Toggles the cursor between a box and an underscore character
Clear screen	CLEAR	<cls>	Clears the entire display area and positions the cursor in the top left of the screen (alternate function of cursor select)
Erase EOF	ERASE EOF	<EEOF>	Erases all characters from the cursor position to the right edge of a data entry field
Arrow keys	↑↓ ← →	-	Moves the cursor up/down/left/right one position, not to next entry field
Enter	ENTER	<enter>	Transmits all information entered to computer
Insert	a	<ins>	Puts screen into insert mode; characters typed will not overwrite those characters already displayed on the screen (see <res>)
Delete	a̸	<del>	Deletes character displayed at the current cursor position
Reset	RESET	<res>	Takes the terminal out of insert mode and unlocks the keyboard after an illegal action has been flagged (when the stick figure can be seen in the status area)

Table 2-2. Program Function Keys

Key Name	Keyboard Symbol	Text Symbol	Purpose
Help	PF1	<help>	Displays a long message if a short message is being displayed; displays a help panel if no message is present or if pressed a second time while a message is being displayed
Split screen	PF2	<split>	Splits the screen into a second copy of ISPF/SDE at the current cursor position
End	PF3	<end>	Ends the current display
Return	PF4	<rtn>	Returns to the primary option panel
Find next	PF5	<fnxt>	Finds next occurrence of a pattern in edit or browse
Replace Next	PF6	<rnxt>	Replaces next occurrence of a pattern in edit only
Scroll Up	PF7	<up>	Moves the display window up
Scroll down	PF8	<down>	Moves the display window down
Swap	PF9	<swap>	Moves the cursor to the opposite window
Scroll left	PF10	<left>	Moves the display window to the left
Scroll right	PF11	<right>	Moves the display window to the right
Cursor	PF12	<crsr>	Moves the cursor to the top left entry position in the current window

Table 2-3. System Status Area Symbols

Name	Symbol	Meaning
Clock	◷	Wait
System	SYSTEM	CPU is processing last request
Stick figure	🯅	Something was typed in an area of the display that is not a valid input area; press the <res> key to recover from this problem; use the <FT>, <BT>, or <NL> keys to move the cursor into an input area
caret	^	Insert mode is turned on
Caps lock	⬆	The <caps lock> key has been pressed

The 327x device processes information displayed on the screen and modifies it until the user presses <enter>. At that time, the screen is transmitted to the computer, a response is received from the computer, and the screen is rewritten.

The ISPF editor has the capability to insert, delete, move, copy, and sort single lines or blocks of lines anywhere in a file, at the discretion of the developer. A few basic editing capabilities are described below. The developer is, however, urged to use the online help available to learn all of the editor's capabilities.

- Inserting characters--To insert a character, press the <ins> key on the keyboard (note the ^ in the status area), position the cursor on the screen where characters are to be inserted, and begin typing. Characters to the right of the cursor will be pushed to the right as new characters are inserted. If pushing the characters would make the one on the far right of the line go off the screen, the keyboard will lock. Press <res> to recover. Then use the <right> key to move the display window so that inserted characters and the moving characters are in the display area.

If the editor will not allow characters to be inserted when there are apparently no characters on the line (it displays the stick figure and clicks when any key is struck), it is trying to shift blank characters to the right while new characters are being typed. This can be remedied either by moving the cursor to the right of the last nonblank character on the line and pressing <EEOF> or by typing "nulls on" in the command line area. "Nulls on" tells the editor to pad empty lines with null characters instead of blanks.

- Changing a character--To change a character, simply position the cursor on top of the character to be changed and type over it. If insert mode is on (^ is in the status area), press <res> to turn it off.

- Deleting characters--To delete a character, position the cursor on the character to be deleted and press the <del> key. All characters on the right of the cursor will be shifted to the left by one column.

- Moving a line--To move a line, mark the destination by typing an "a" or "b" on the line number the line is to be moved after (a) or before (b), and mark the source by typing an "m" on the line to be moved. Press <enter>, and the line marked with an "m" will be moved.

- Copying a line--To copy a line, mark the destination by typing an "a" or "b" on the line number the line is to be copied after (a) or before (b), and mark the source by typing a "c" on the line to be copied. Press <enter>, and the line marked with a "c" will be copied.

- Deleting lines--To delete lines from a file, move the cursor into the left side of the display with the <NL> key. Position it on each line number you want to delete, and type a "d" to mark it. After all lines to be deleted have been marked, press <enter> and all the lines marked will disappear from the file.

- Inserting a line--To insert a new line in a file, move the cursor into the numbered area of the display with the <NL> key. Position it on the line number the new line is to be inserted after, and type an "i". Press <enter>, and a line with a row of dots in the line number area will appear. Position the cursor where desired with the <arrow> keys, and type in the new line. When <enter> is pressed, another row of dots will appear and the process can be repeated. The line insertion process ends when <enter> is pressed but nothing is typed.

- Inserting many lines--To insert many new lines in a file, move the cursor into the numbered area of the display with the <NL> key. Position it on the line number the new lines are to be inserted after and type an "i<n>", where <n> is the number of lines to be inserted. Press <enter>, and <n> rows of dots will appear in the numbered area. Position the cursor where desired with the <arrow> keys, and type in the new information.

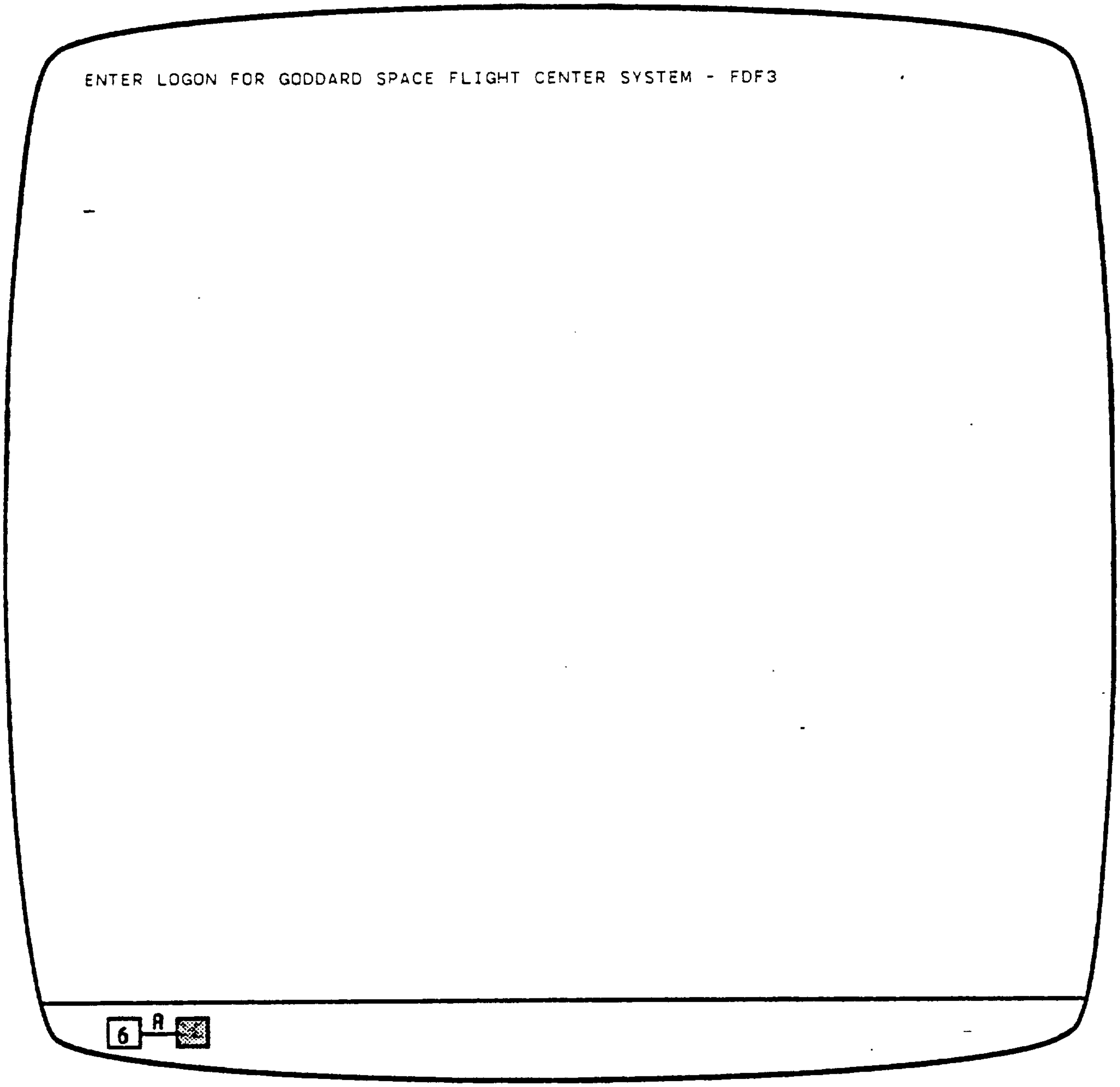
- Displaying next page of file--To display the next page of a file, press <PF8> or type "down" in the command line input area.

- Displaying previous page of file--To display the previous page of a file, press <PF7> or type "up" in the command line input area.

SECTION 3 - SAMPLE DEVELOPMENT SCENARIO

Step one of the scenario requires logging on to TSO. The procedure differs depending on whether the terminal is connected directly to the FDF3 at Goddard Space Flight Center (GSFC) or is connected through a modem. Most of the 327x terminals connected to the FDF3 at GSFC are direct-connect terminals. Developers using these terminals should begin the scenario on page 3-2. Developers using terminals that are connected to the FDF3 through a modem should begin the scenario on page 3-4.

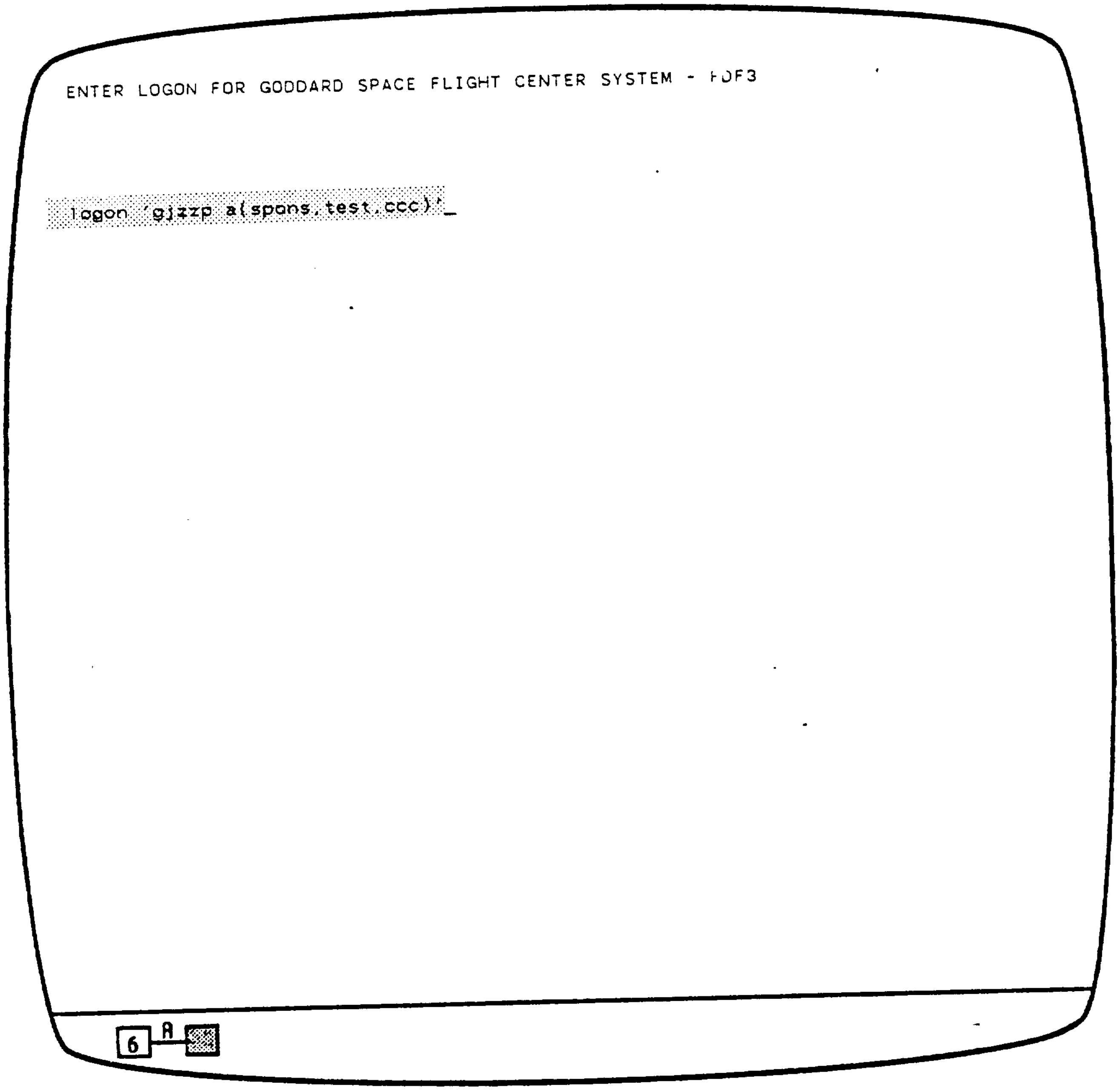
Scenario Step 1: Logging on TSO and entering SDE



```
ENTER LOGON FOR GODDARD SPACE FLIGHT CENTER SYSTEM - FDF3

_
```

This is what will appear on the top line of the terminal before you log on to the 4341 computer. If the terminal that you want to use does not display the request seen above, press and hold the <alt> key and the <csel> key. This performs the alternate function of that key, which is clear-screen. If you still don't get the request message, tell your supervisor.

Scenario Step 1: Logging on TSO and entering SDE

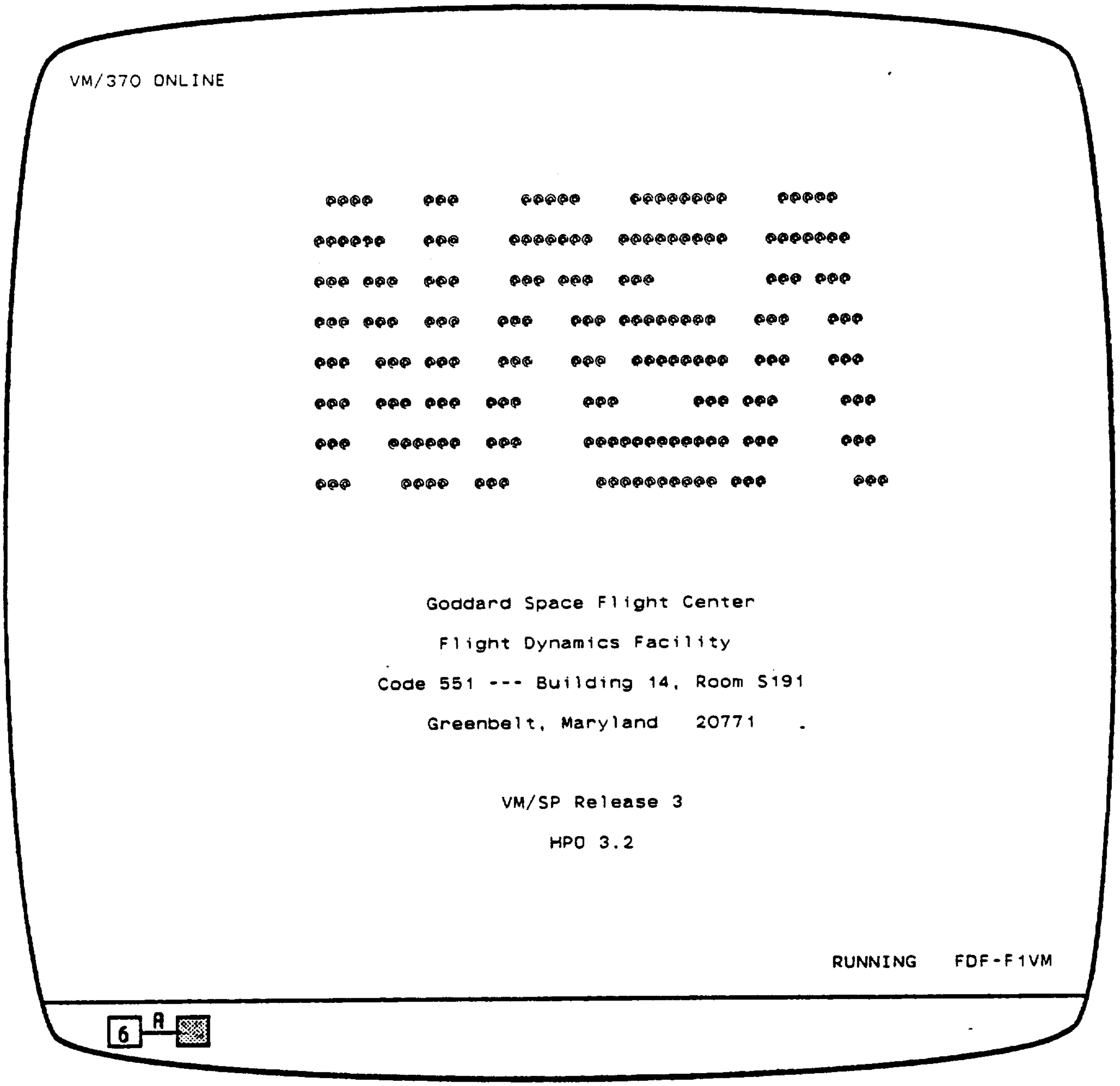


```
ENTER LOGON FOR GODDARD SPACE FLIGHT CENTER SYSTEM - FDF3


logon 'gjzzp a(spons,test,ccc)'_
```

Type in your user identifier (userid), sponsor code, and project code. You can get this information from your supervisor. The userid, sponsor, and project codes above are gjzzp, spons, and test, respectively.

Press <enter>, the screen will clear, . . . (go to page 3-10)

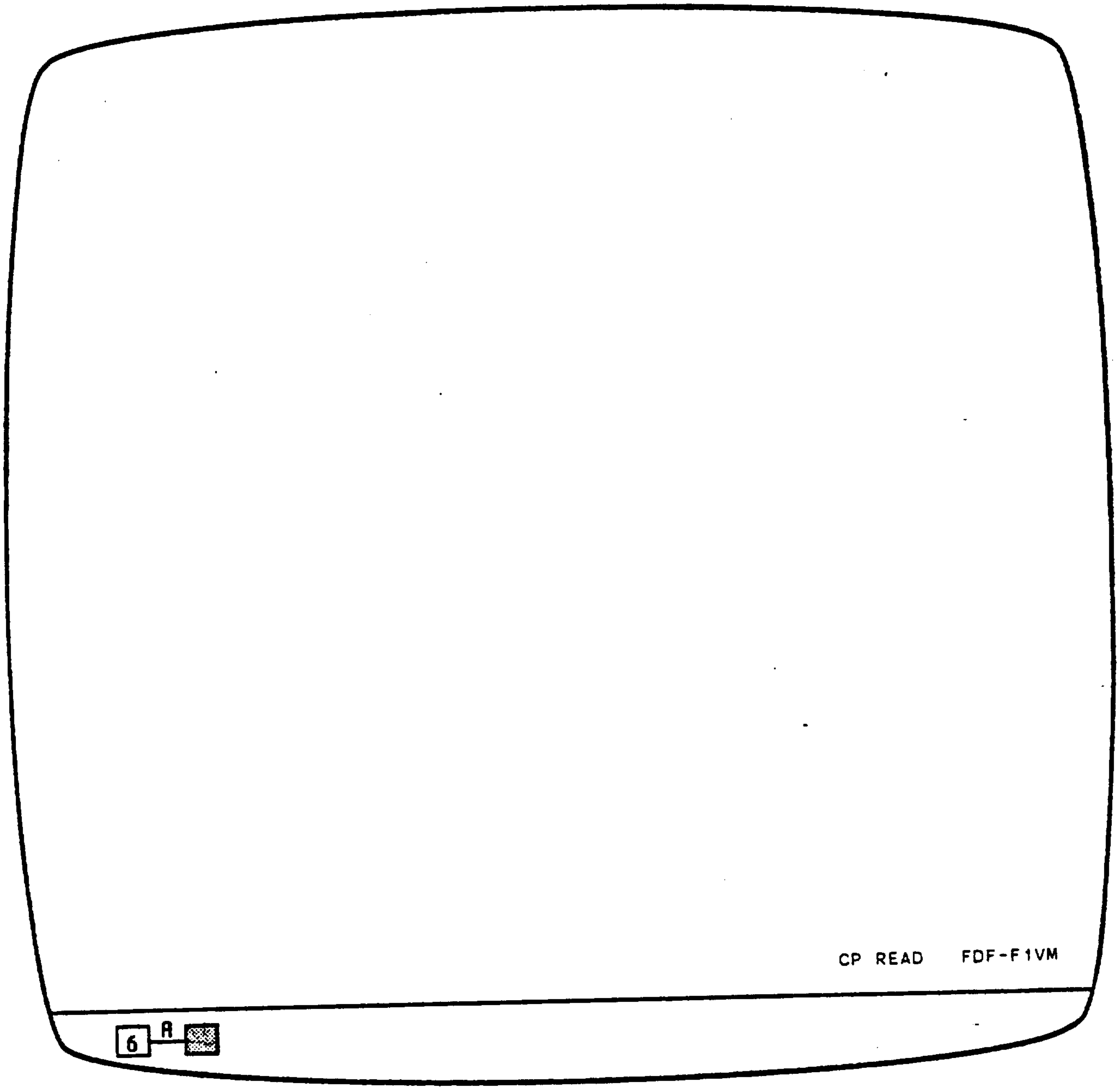
Scenario Step 1: Logging on TSO and entering SDE



```
VM/370 ONLINE


            @@@@    @@@    @@@@@    @@@@@@@@    @@@@@
           @@@@@@   @@@   @@@@@@@  @@@@@@@@@   @@@@@@@
           @@@ @@@  @@@   @@@ @@@  @@@         @@@ @@@
           @@@ @@@  @@@  @@@   @@@ @@@@@@@@  @@@   @@@
           @@@  @@@ @@@  @@@   @@@  @@@@@@@@ @@@   @@@
           @@@  @@@ @@@ @@@     @@@      @@@ @@@     @@@
           @@@   @@@@@@ @@@     @@@@@@@@@@@@ @@@     @@@
           @@@    @@@@ @@@       @@@@@@@@@@ @@@       @@@


                     Goddard Space Flight Center
                      Flight Dynamics Facility
                 Code 551 --- Building 14, Room S191
                     Greenbelt, Maryland   20771


                          VM/SP Release 3
                             HPO 3.2


                                             RUNNING    FDF-F1VM
```

This panel is displayed before any action is taken at the terminal. It is called the VM prompt panel, or the logo panel, and indicates the terminal is connected to the Flight Dynamics Facility (FDF) computer system. There are currently three cómputers in the FDF, referred to as the F1, F2, and the F3. The Software Development Environment (SDE) is only installed on the F3. The first action to take is to connect the terminal to the F3 computer. This is done by pressing the <enter> key . . .

Scenario Step 1: Logging on TSO and entering SDE

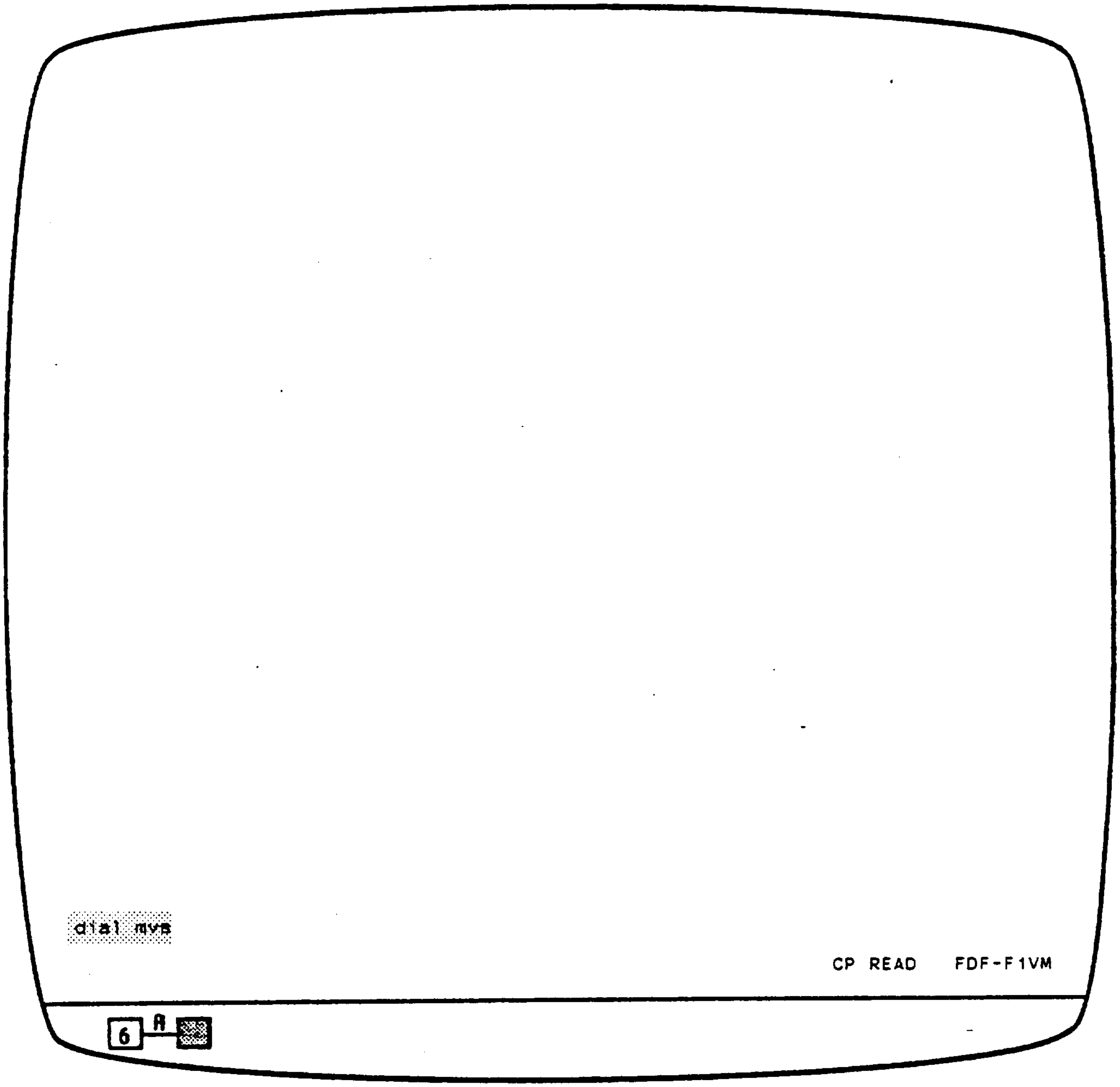


```
                                                    CP READ    FDF-F1VM
```

. . . and this panel is displayed, indicating the terminal is connected to the F1 computer ('FDF-F1VM' is displayed in the lower right corner).

Next we type in . . .

Scenario Step 1: Logging on TSO and entering SDE

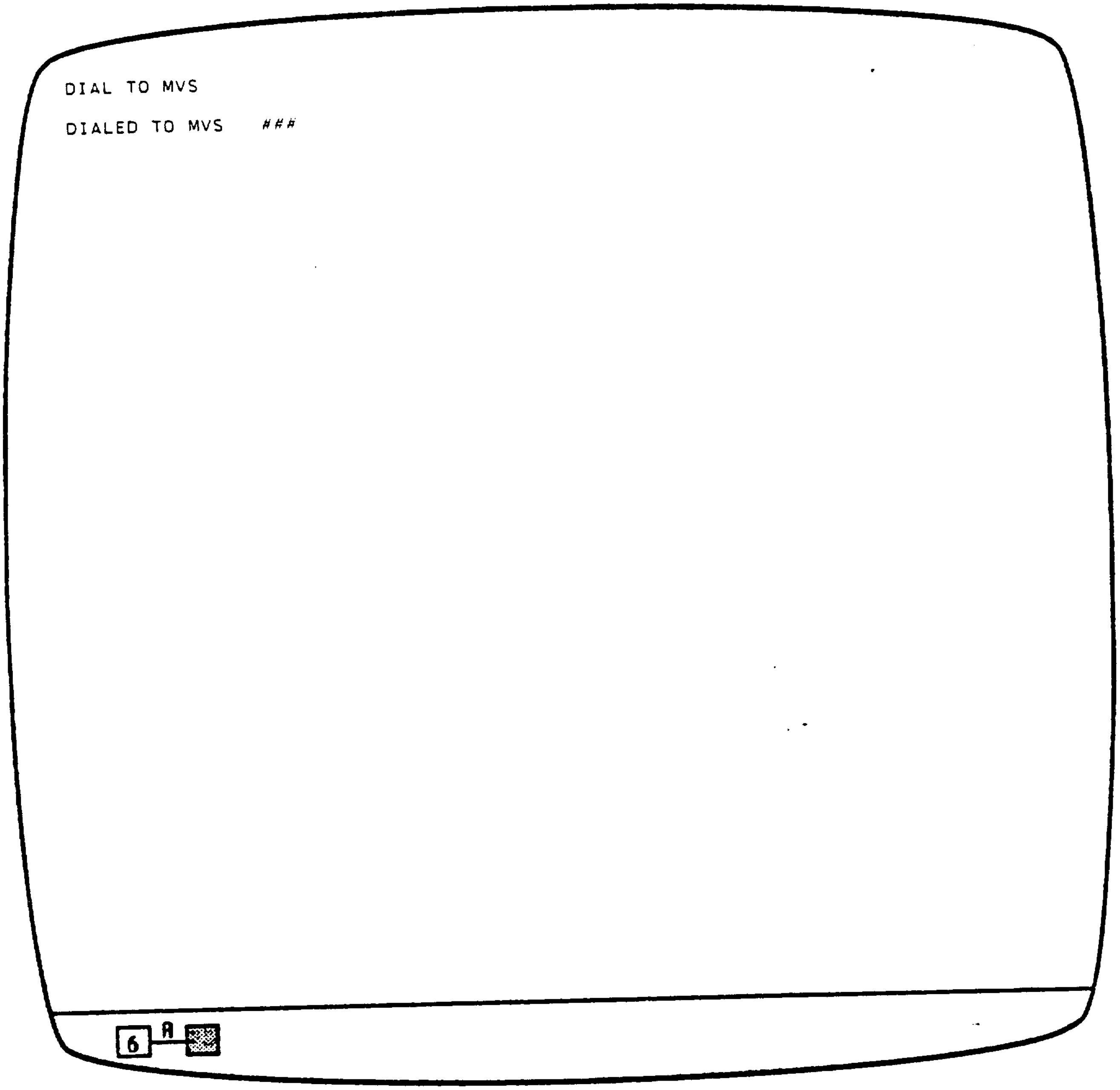


```
dial mvs
                                                        CP READ     FDF-F1VM
```

. . . 'dial mvs', then press <enter>, to use the MVS operating system.

The screen will clear . . .

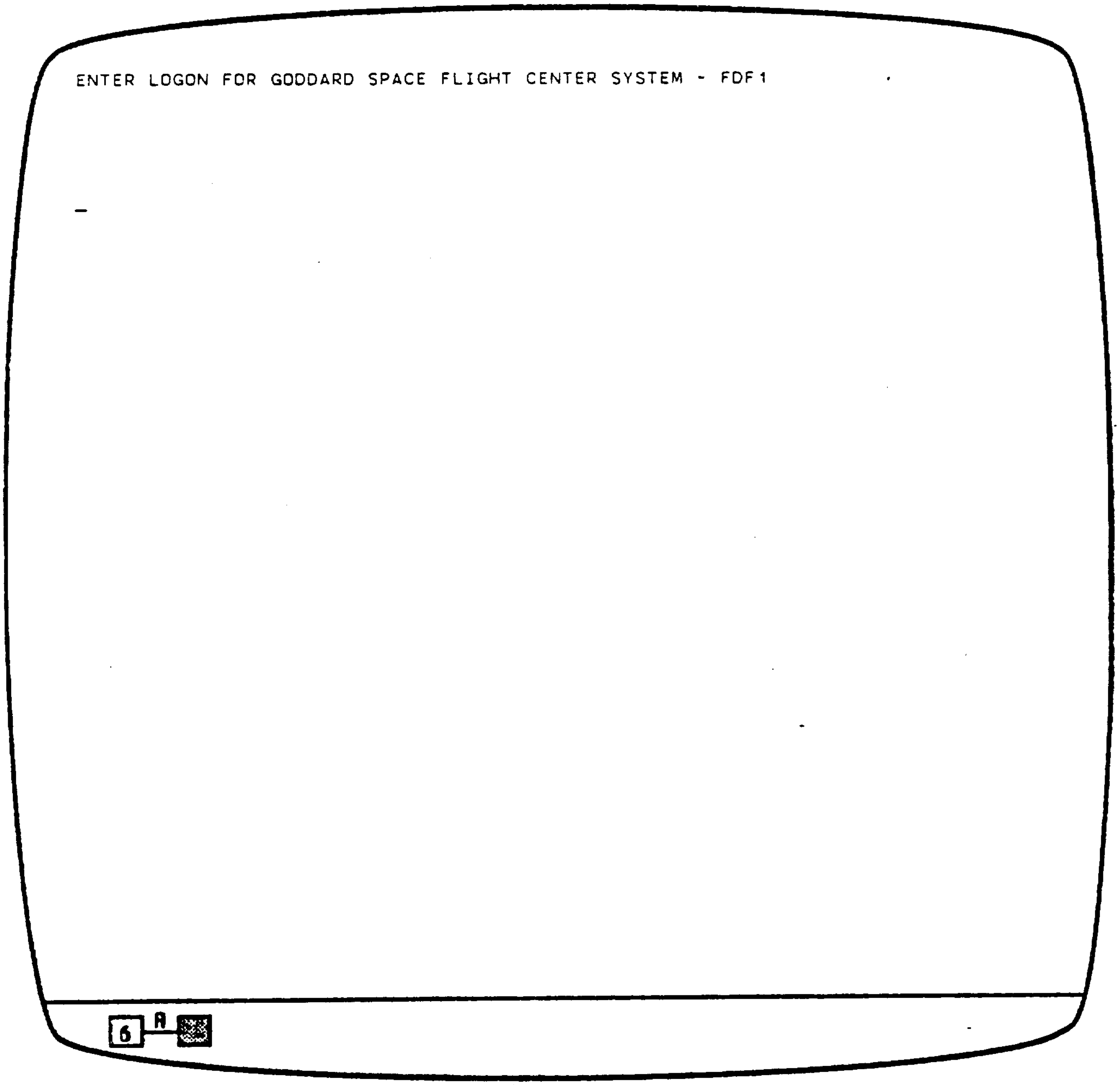
Scenario Step 1: Logging on TSO and entering SDE



```
DIAL TO MVS
DIALED TO MVS   ###
```

'DIAL TO MVS' is repeated at the top left corner of the screen. 'DIALED TO MVS' and three hexadecimal digits appear briefly two lines lower on the screen; the screen clears again, and . . .

Scenario Step 1: Logging on TSO and entering SDE

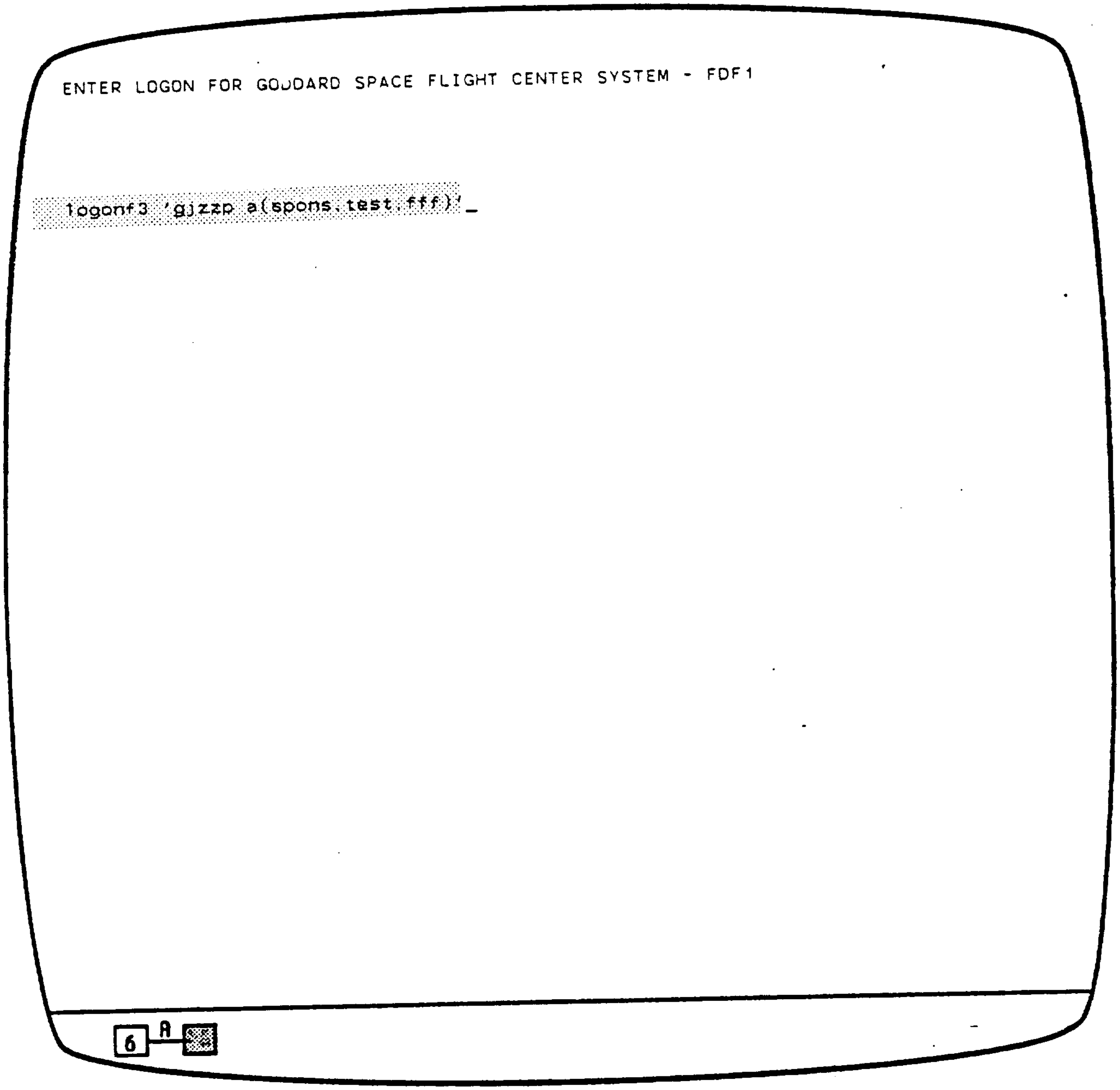


```
ENTER LOGON FOR GODDARD SPACE FLIGHT CENTER SYSTEM - FDF1



_
```

. . . this message appears on the top line of the terminal, indicating the terminal is connected to the F1 computer under the MVS operating system.

A special logon command, LOGONF3, switches the terminal from the F1 to the F3 computer.

Scenario Step 1: Logging on TSO and entering SDE

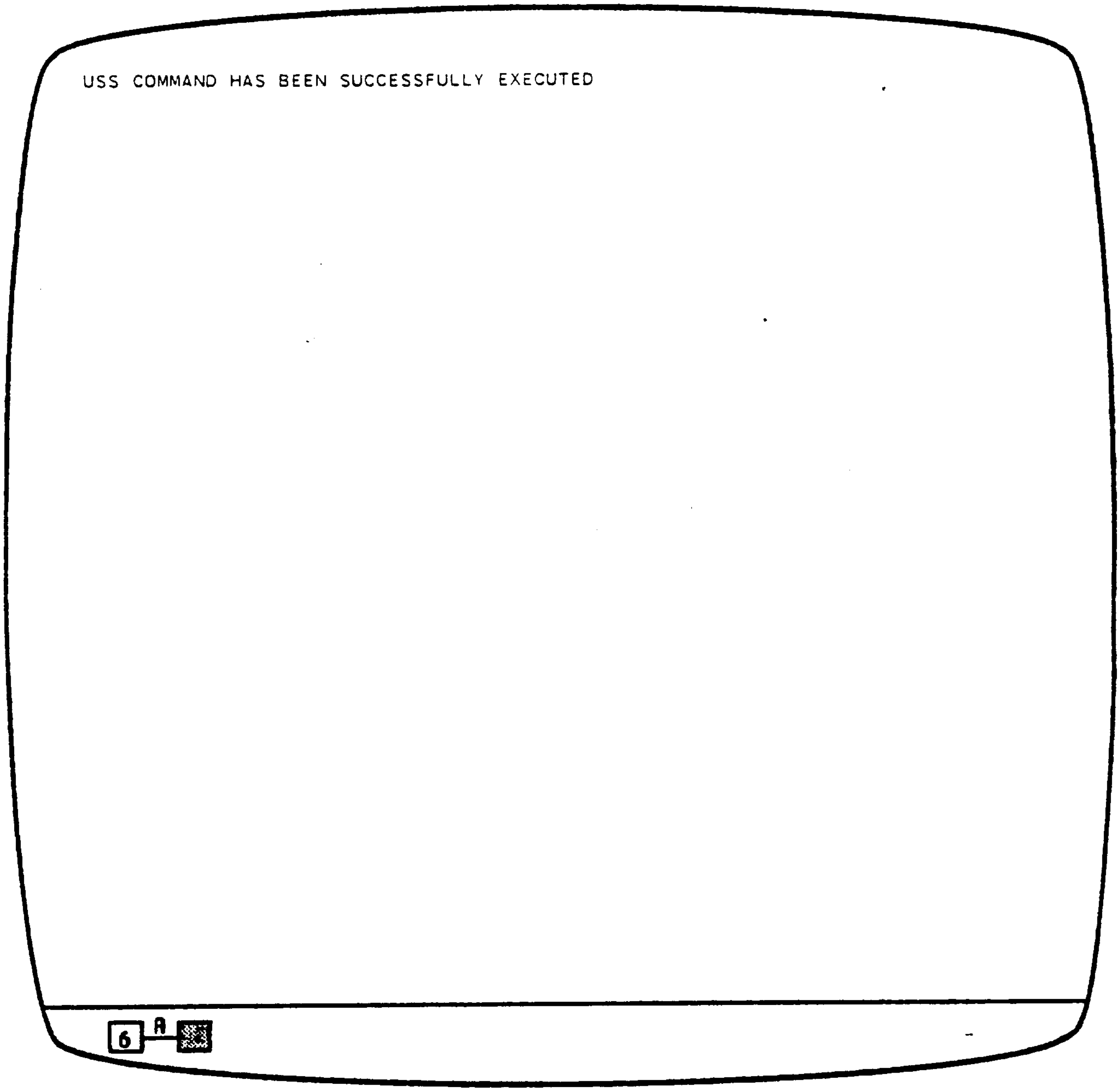


```
ENTER LOGON FOR GODDARD SPACE FLIGHT CENTER SYSTEM - FDF1


logonf3 'gjzzp a(spons.test.fff)'_
```

Type in LOGONF3, your user identifier (userid), sponsor code, and project code. You can get this information from your supervisor. The userid, sponsor, and project codes above are gjzzp, spons, and test, respectively.

Press <enter>, the screen will clear, . . .

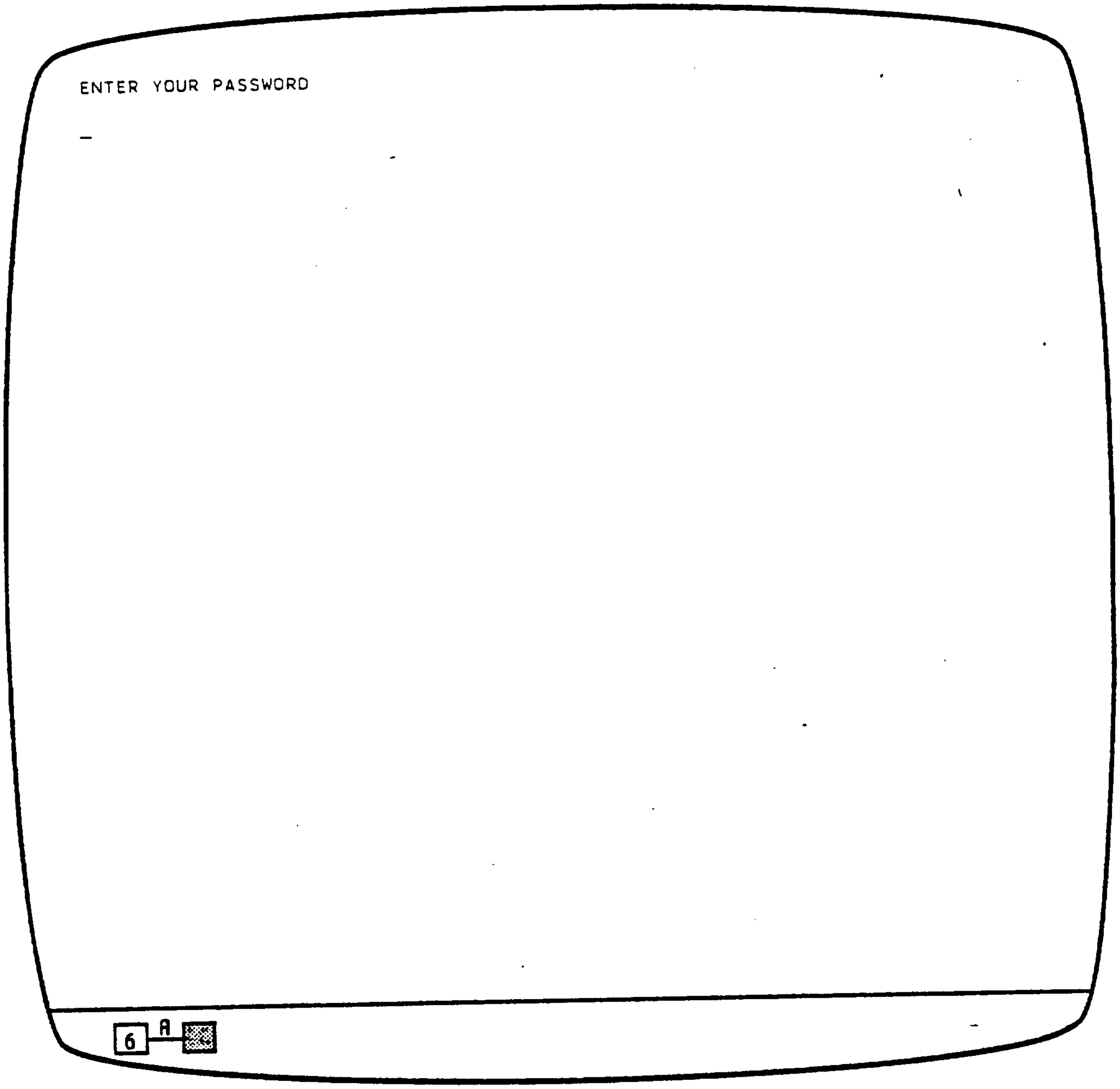
Scenario Step 1: Logging on TSO and entering SDE



```
USS COMMAND HAS BEEN SUCCESSFULLY EXECUTED
```

. . . and this message is displayed briefly after the logon command has been accepted. In a short time the screen will clear again, and . . .

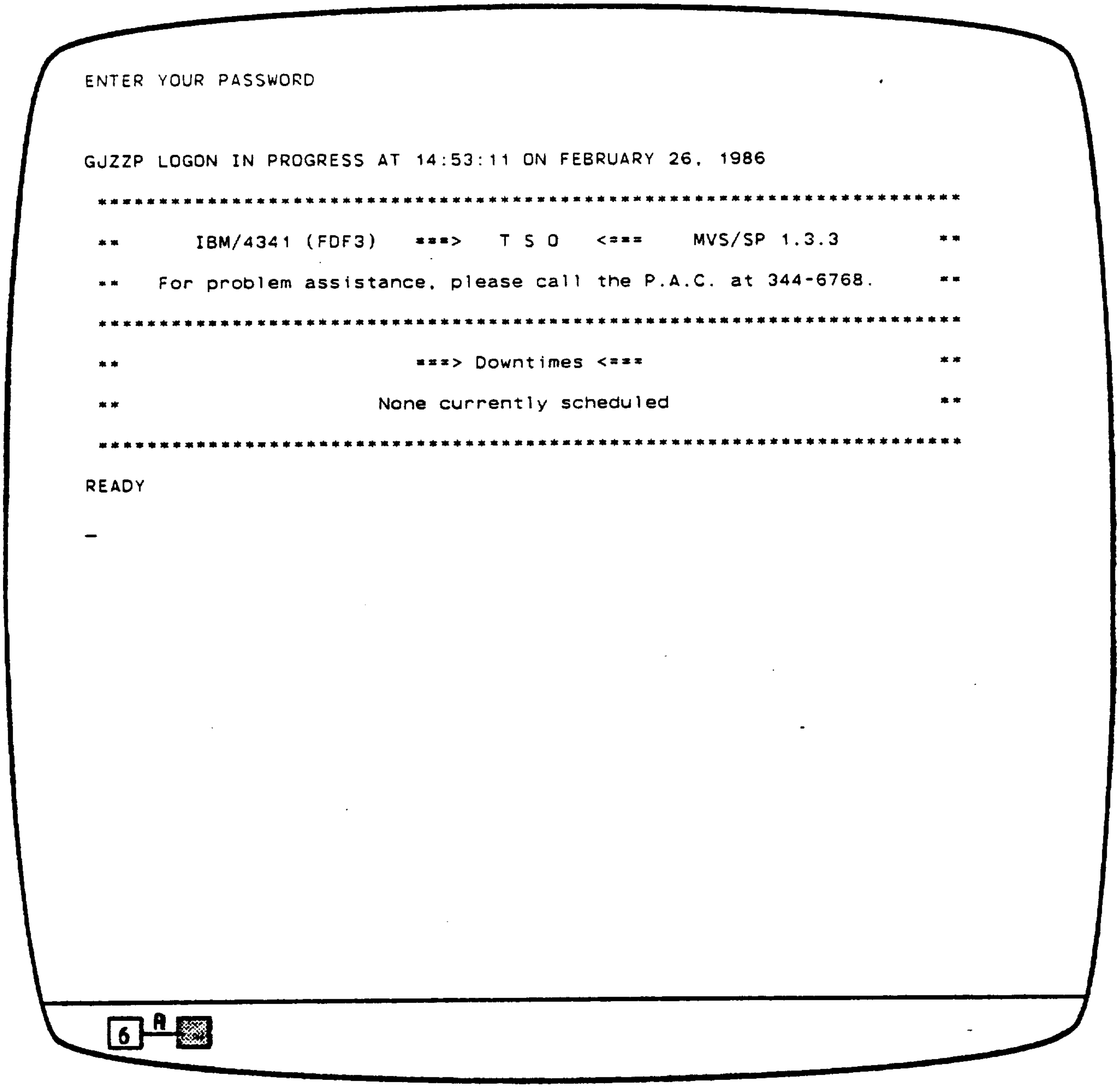
Scenario Step 1: Logging on TSO and entering SDE



```
ENTER YOUR PASSWORD

_
```

. . . you are prompted for your TSO password. Type it in. The letters you type will not be displayed. Press <enter> when you have finished; in a few seconds . . .

Scenario Step 1: Logging on TSO and entering SDE

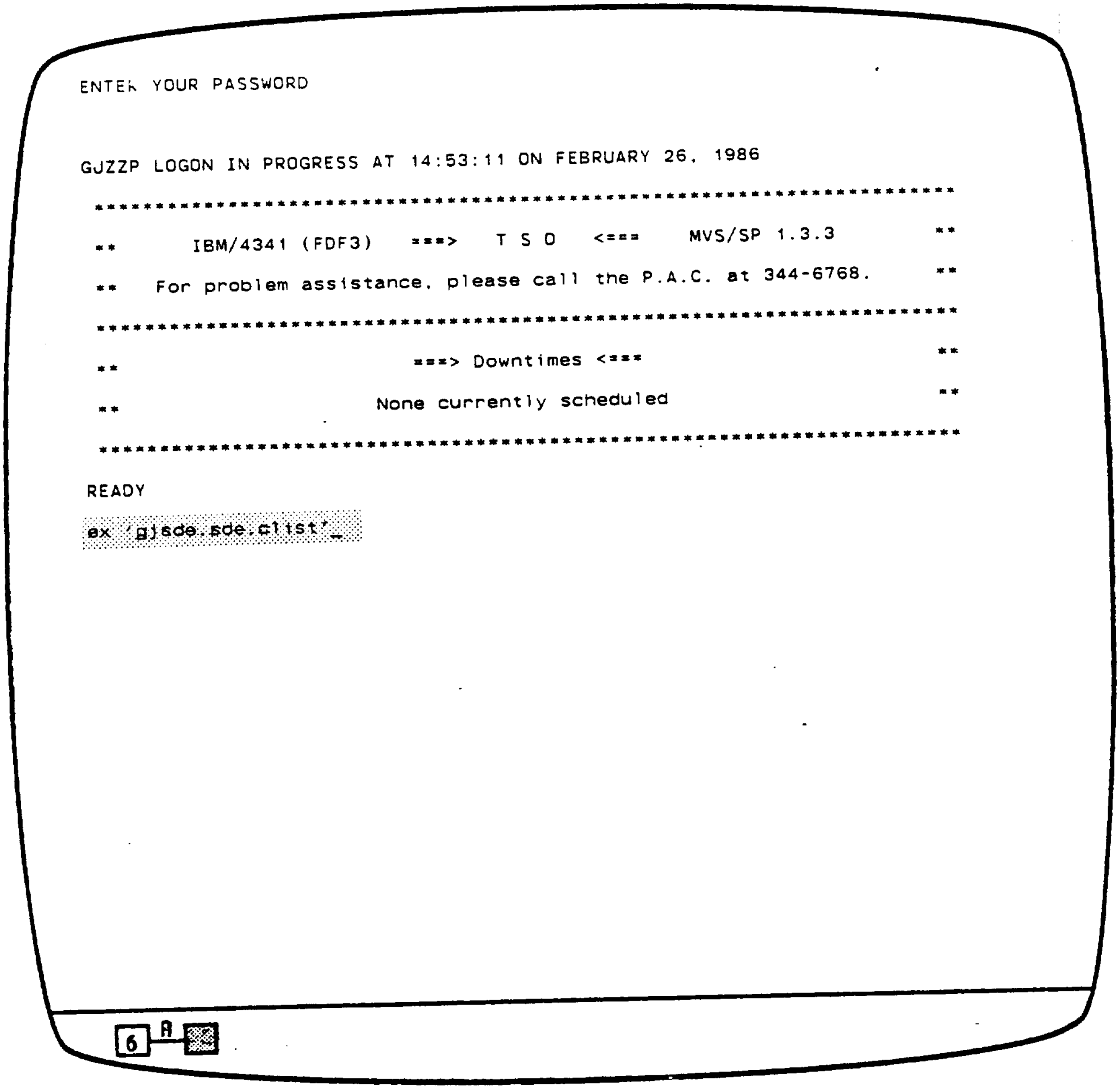


```
ENTER YOUR PASSWORD

GJZZP LOGON IN PROGRESS AT 14:53:11 ON FEBRUARY 26, 1986
 ******************************************************************
 **      IBM/4341 (FDF3)   ===>   T S O   <===    MVS/SP 1.3.3      **
 **   For problem assistance, please call the P.A.C. at 344-6768.   **
 ******************************************************************
 **                        ===> Downtimes <===                      **
 **                      None currently scheduled                   **
 ******************************************************************
READY

_
```

. . . these messages will appear. The time and date will differ from the ones shown here, and there may be some downtime scheduled to maintain the F3 computer system, but this basic format will be displayed.

We have successfully logged on to TSO and now must enter the software development environment. To do this, . . .

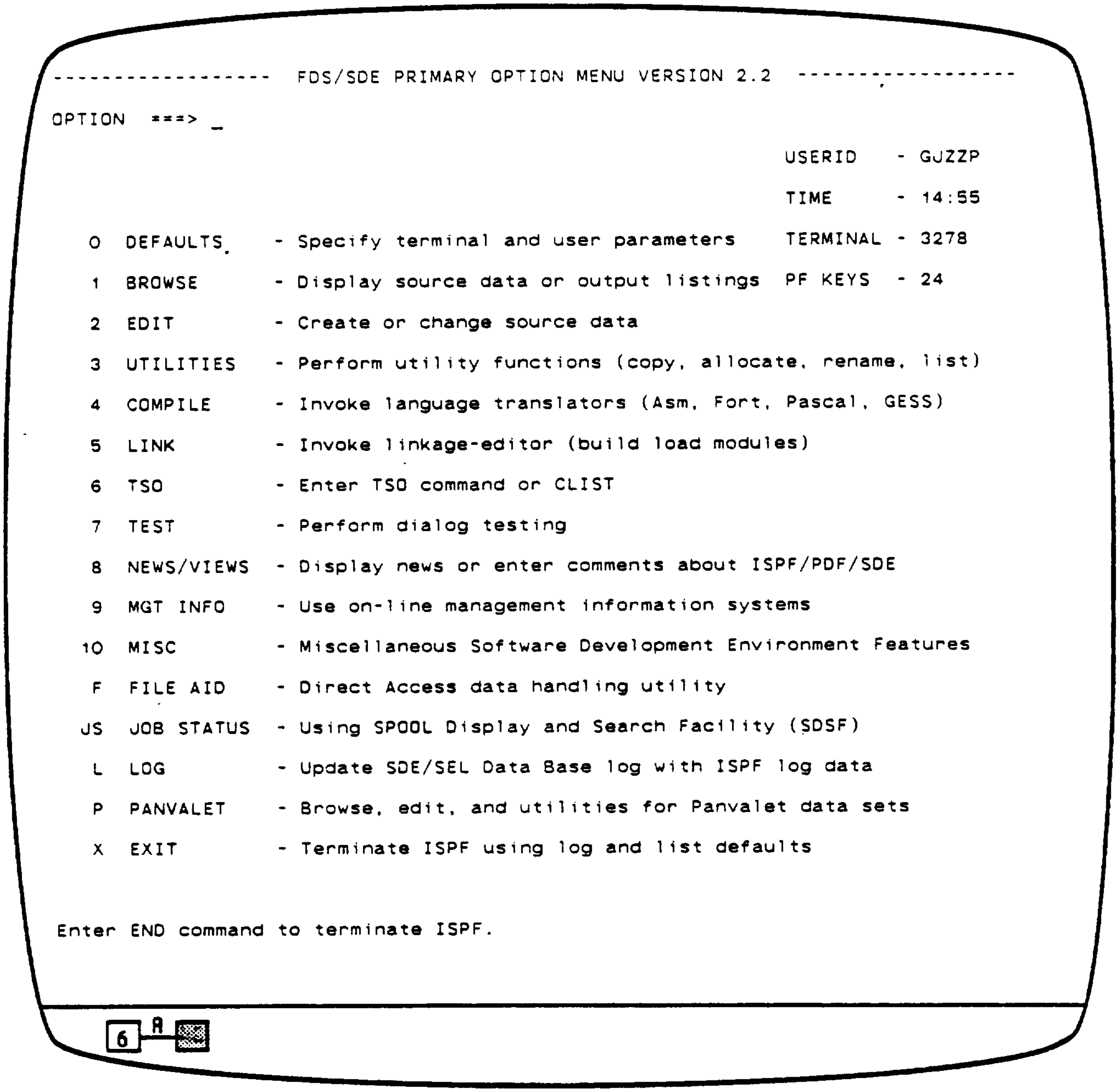
Scenario Step 1: Logging on TSO and entering SDE



```
ENTER YOUR PASSWORD


GJZZP LOGON IN PROGRESS AT 14:53:11 ON FEBRUARY 26, 1986
 ***********************************************************************
 **       IBM/4341 (FDF3)    ===>    T S O    <===    MVS/SP 1.3.3        **
 **   For problem assistance, please call the P.A.C. at 344-6768.        **
 ***********************************************************************
 **                         ===> Downtimes <===                            **
 **                       None currently scheduled                         **
 ***********************************************************************
READY
ex 'gjsde.sde.clist'_
```

. . . type in the command above and press <enter>. In a few moments . . .

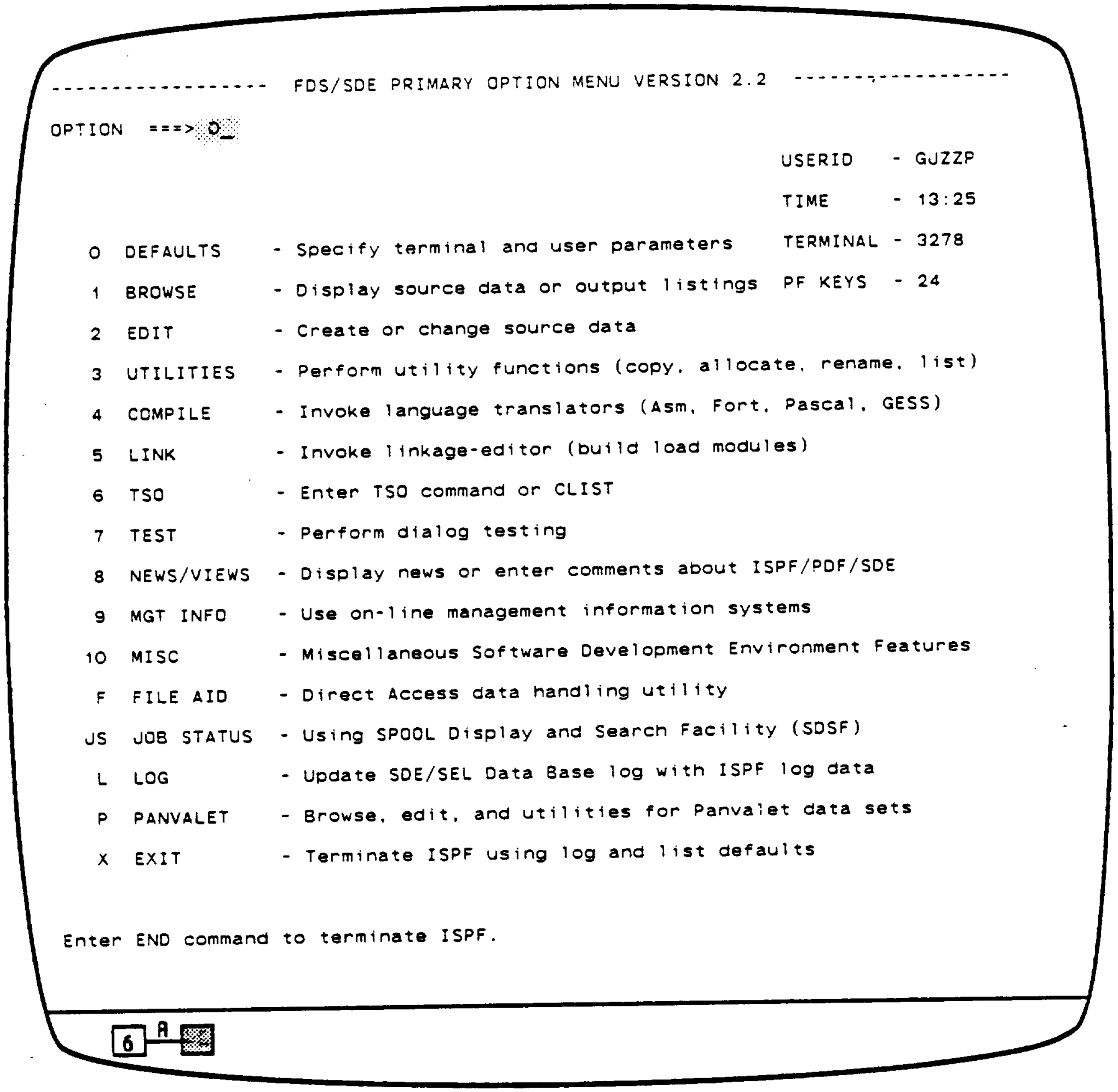
Scenario Step 1: Logging on TSO and entering SDE



```
------------------  FDS/SDE PRIMARY OPTION MENU VERSION 2.2  ------------------
OPTION  ===> _
                                                          USERID    - GJZZP
                                                          TIME      - 14:55
   0  DEFAULTS    - Specify terminal and user parameters  TERMINAL - 3278
   1  BROWSE      - Display source data or output listings  PF KEYS  - 24
   2  EDIT        - Create or change source data
   3  UTILITIES   - Perform utility functions (copy, allocate, rename, list)
   4  COMPILE     - Invoke language translators (Asm, Fort, Pascal, GESS)
   5  LINK        - Invoke linkage-editor (build load modules)
   6  TSO         - Enter TSO command or CLIST
   7  TEST        - Perform dialog testing
   8  NEWS/VIEWS  - Display news or enter comments about ISPF/PDF/SDE
   9  MGT INFO    - Use on-line management information systems
  10  MISC        - Miscellaneous Software Development Environment Features
   F  FILE AID    - Direct Access data handling utility
  JS  JOB STATUS  - Using SPOOL Display and Search Facility (SDSF)
   L  LOG         - Update SDE/SEL Data Base log with ISPF log data
   P  PANVALET    - Browse, edit, and utilities for Panvalet data sets
   X  EXIT        - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

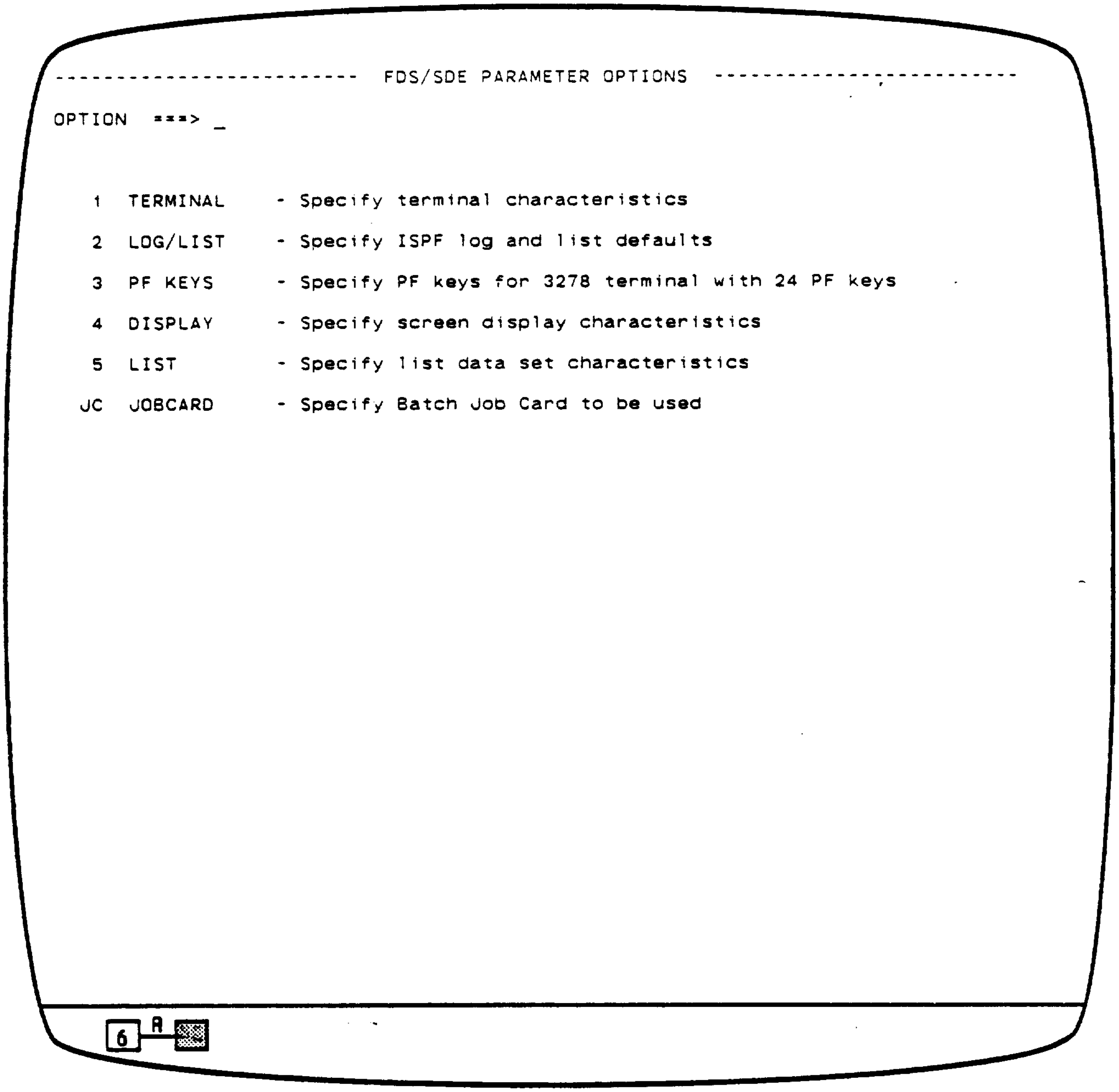
This panel is displayed as we enter the Software Development Environment (SDE). It is called the primary option panel and lists all of the major options available to the SDE developer. The underscore character in the upper left is only marking the current cursor position, it never has to be typed in. In order to perform step 2 of our scenario, we have to define an IBM job card. Job cards are used to provide accounting information and several other types of information used to process background jobs.

Scenario Step 2: Defining IBM Job Card



```
------------------  FDS/SDE PRIMARY OPTION MENU VERSION 2.2  ------------------

OPTION  ===> 0_
                                                          USERID   - GJZZP

                                                          TIME     - 13:25

  0  DEFAULTS    - Specify terminal and user parameters    TERMINAL - 3278

  1  BROWSE      - Display source data or output listings  PF KEYS  - 24

  2  EDIT        - Create or change source data

  3  UTILITIES   - Perform utility functions (copy, allocate, rename, list)

  4  COMPILE     - Invoke language translators (Asm, Fort, Pascal, GESS)

  5  LINK        - Invoke linkage-editor (build load modules)

  6  TSO         - Enter TSO command or CLIST

  7  TEST        - Perform dialog testing

  8  NEWS/VIEWS  - Display news or enter comments about ISPF/PDF/SDE

  9  MGT INFO    - Use on-line management information systems

 10  MISC        - Miscellaneous Software Development Environment Features

  F  FILE AID    - Direct Access data handling utility

 JS  JOB STATUS  - Using SPOOL Display and Search Facility (SDSF)

  L  LOG         - Update SDE/SEL Data Base log with ISPF log data

  P  PANVALET    - Browse, edit, and utilities for Panvalet data sets

  X  EXIT        - Terminate ISPF using log and list defaults


Enter END command to terminate ISPF.
```

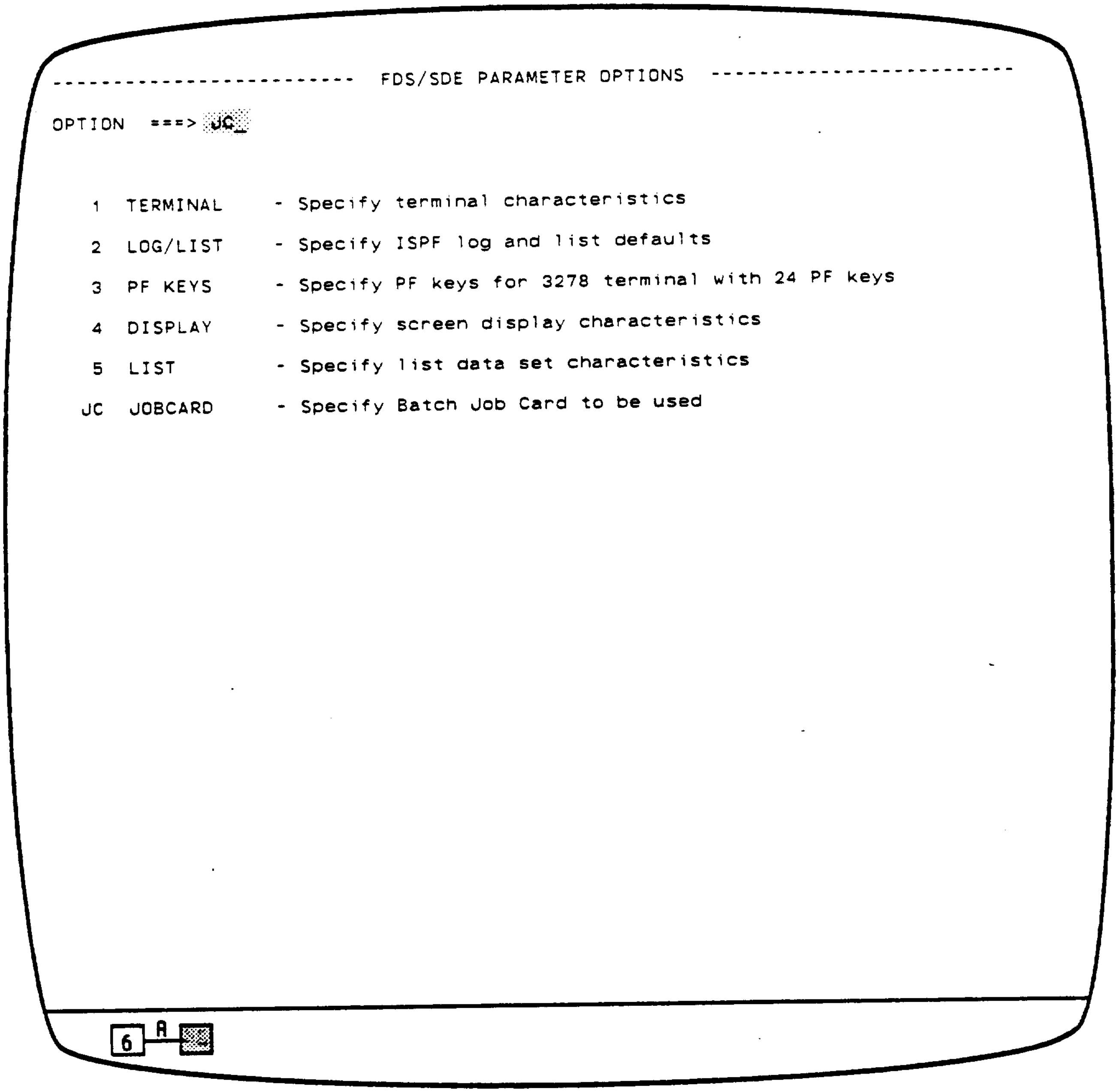
Type a 0 in the option selection field to select DEFAULTS and press the <enter> key.



```
------------------------- FDS/SDE PARAMETER OPTIONS -------------------------

OPTION  ===> _


   1  TERMINAL     - Specify terminal characteristics

   2  LOG/LIST     - Specify ISPF log and list defaults

   3  PF KEYS      - Specify PF keys for 3278 terminal with 24 PF keys

   4  DISPLAY      - Specify screen display characteristics

   5  LIST         - Specify list data set characteristics

  JC  JOBCARD      - Specify Batch Job Card to be used
```

The FDS/SDE parameter options selection panel is displayed . . .

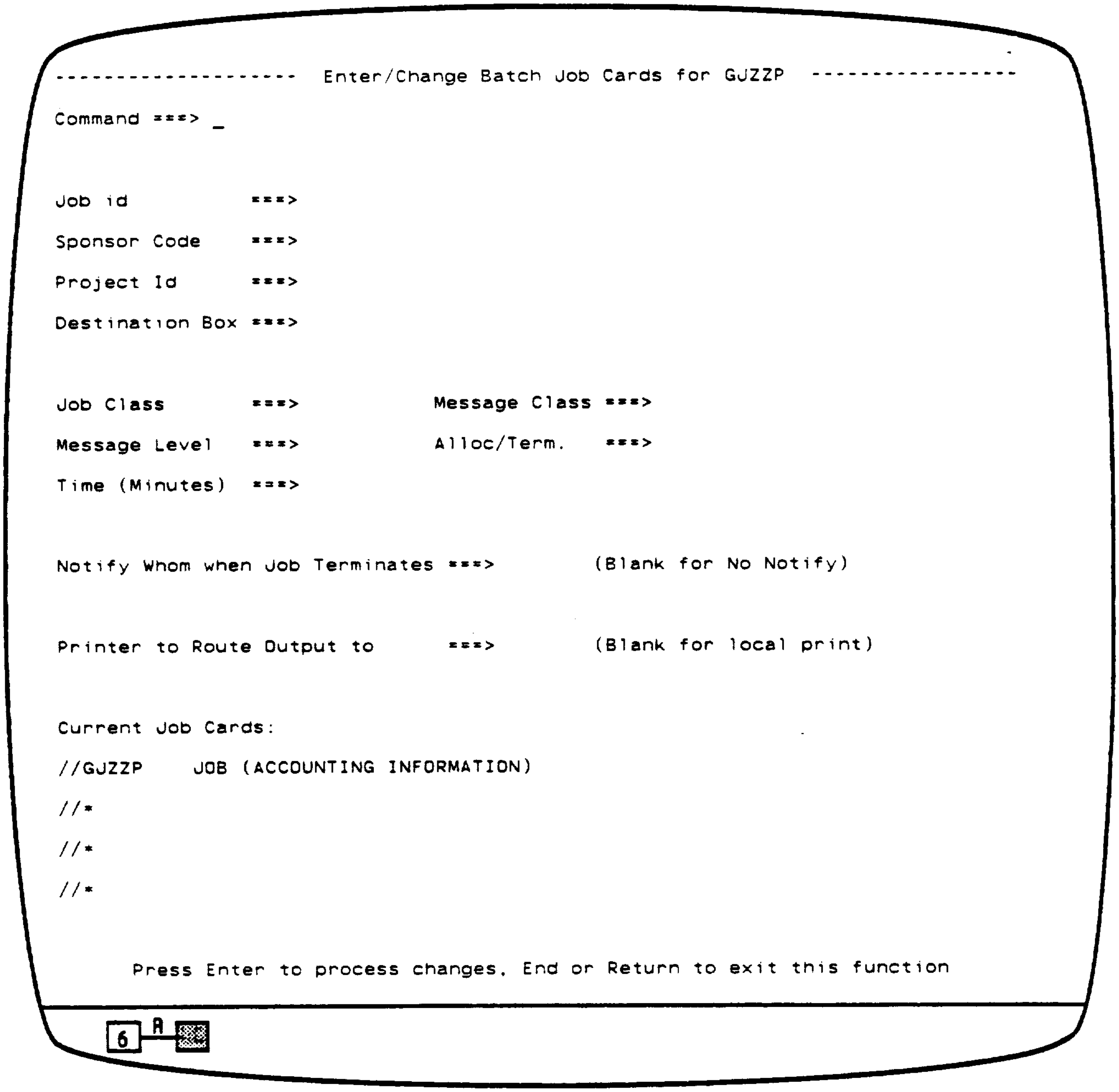
Scenario Step 2: Defining IBM Job Card



```
------------------------ FDS/SDE PARAMETER OPTIONS ------------------------
OPTION  ===> JC_


    1   TERMINAL     - Specify terminal characteristics

    2   LOG/LIST     - Specify ISPF log and list defaults

    3   PF KEYS      - Specify PF keys for 3278 terminal with 24 PF keys

    4   DISPLAY      - Specify screen display characteristics

    5   LIST         - Specify list data set characteristics

   JC   JOBCARD      - Specify Batch Job Card to be used
```

. . . Type in JC to select the job card option, press <enter>, and . . .

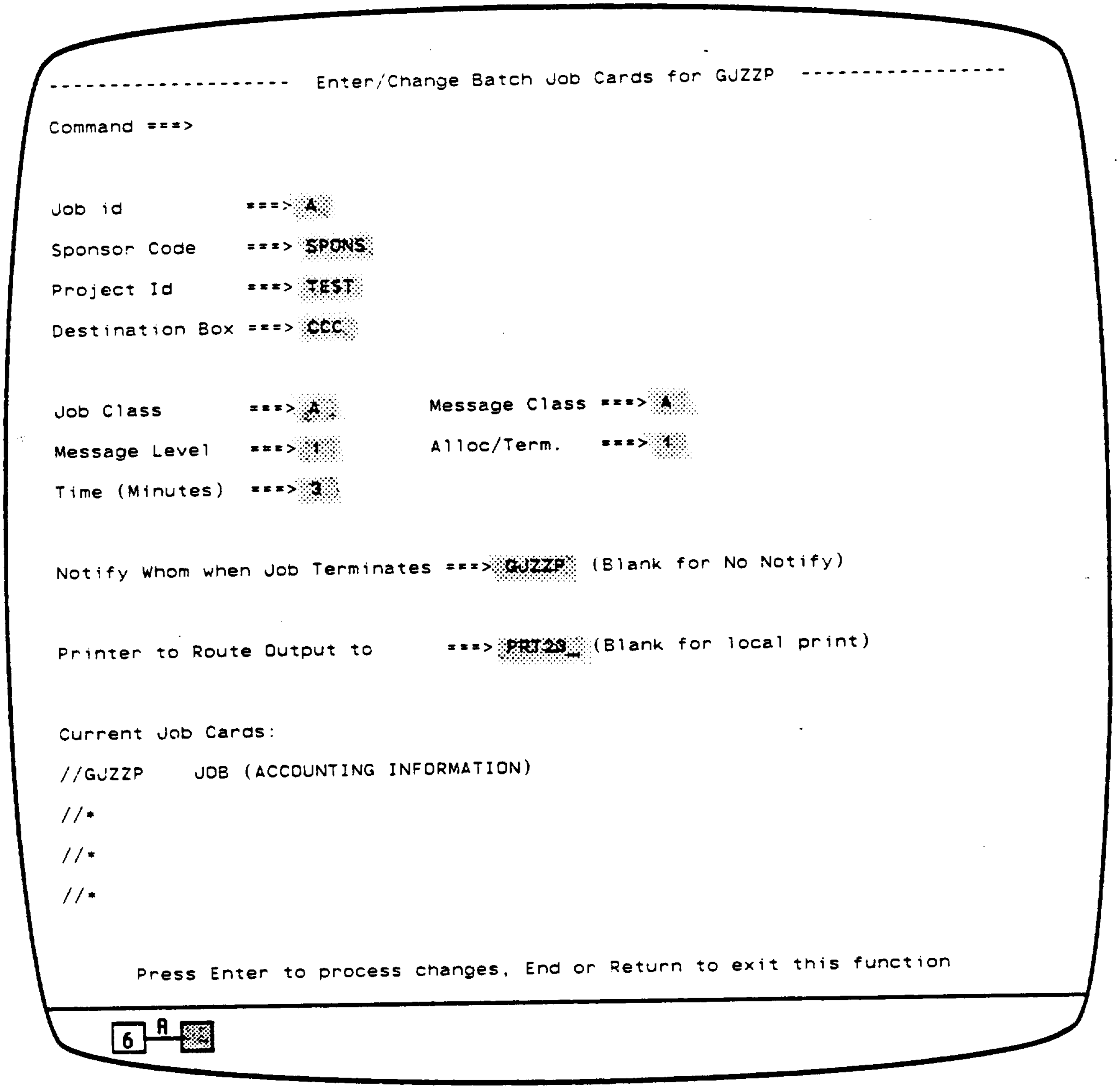
Scenario Step 2: Defining IBM Job Card



```
--------------------  Enter/Change Batch Job Cards for GJZZP  ----------------
Command ===> _


Job id           ===>

Sponsor Code     ===>

Project Id       ===>

Destination Box  ===>


Job Class        ===>            Message Class ===>

Message Level    ===>            Alloc/Term.   ===>

Time (Minutes)   ===>


Notify Whom when Job Terminates ===>        (Blank for No Notify)


Printer to Route Output to      ===>        (Blank for local print)


Current Job Cards:

//GJZZP    JOB (ACCOUNTING INFORMATION)

//*

//*

//*


     Press Enter to process changes, End or Return to exit this function
```

. . . the parameter entry panel for the change job card function is displayed. The actual job cards displayed at the bottom may differ on your display the first time you enter this function. In a moment they will be very similar.

Scenario Step 2: Defining IBM Job Card

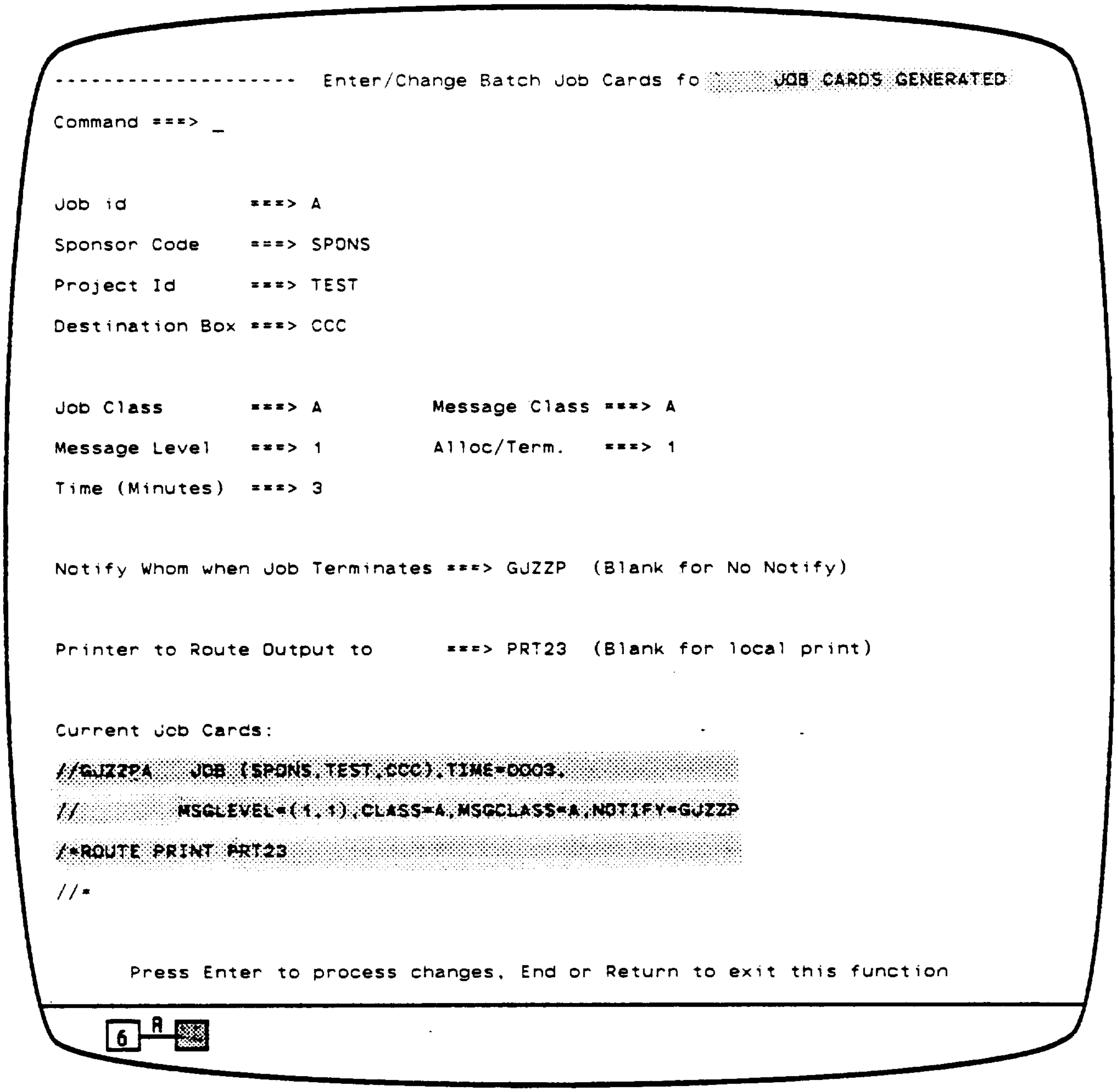


```
--------------------  Enter/Change Batch Job Cards for GJZZP  ----------------

Command ===>


Job id          ===> A

Sponsor Code    ===> SPONS

Project Id      ===> TEST

Destination Box ===> CCC


Job Class       ===> A           Message Class ===> A

Message Level   ===> 1           Alloc/Term.   ===> 1

Time (Minutes)  ===> 3


Notify Whom when Job Terminates ===> GJZZP  (Blank for No Notify)


Printer to Route Output to      ===> PRT23_ (Blank for local print)


Current Job Cards:

//GJZZP    JOB (ACCOUNTING INFORMATION)

//*

//*

//*


      Press Enter to process changes, End or Return to exit this function
```

Pressing <help> will let you read the help panels associated with this function. They explain the various fields and the values that could be entered. Rather than get bogged down with details, use the values for each field as they appear on this panel. <FT> to move the cursor from input field to input field.

Press <enter>, and . . .

Scenario Step 2: Defining IBM Job Card



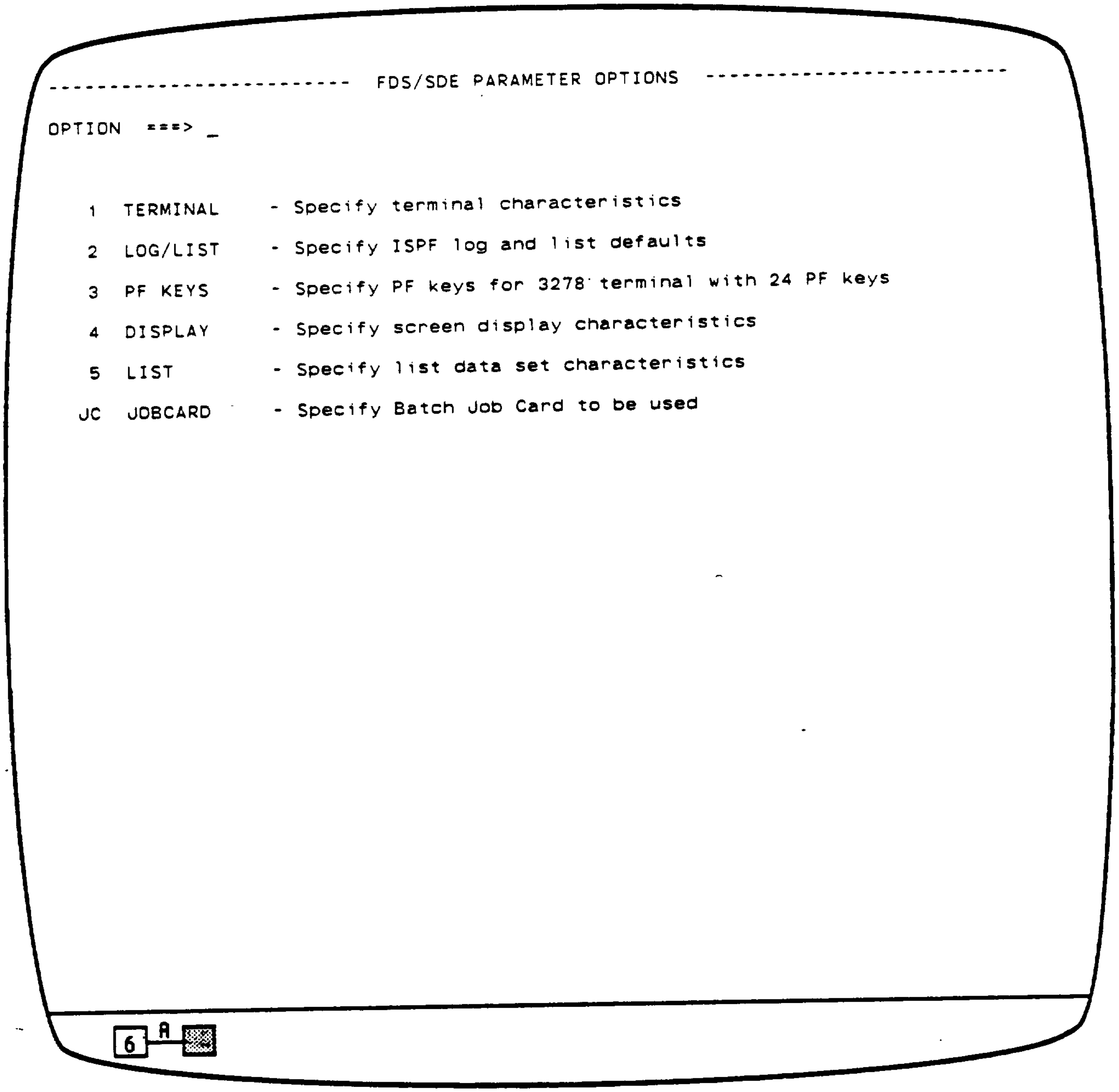
```
-------------------- Enter/Change Batch Job Cards fo      JOB CARDS GENERATED
Command ===> _


Job id          ===> A

Sponsor Code    ===> SPONS

Project Id      ===> TEST

Destination Box ===> CCC


Job Class       ===> A          Message Class ===> A

Message Level   ===> 1          Alloc/Term.   ===> 1

Time (Minutes)  ===> 3


Notify Whom when Job Terminates ===> GJZZP  (Blank for No Notify)


Printer to Route Output to      ===> PRT23  (Blank for local print)


Current Job Cards:

//GJZZPA   JOB (SPONS,TEST,CCC),TIME=0003,

//            MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP

/*ROUTE PRINT PRT23

//*


      Press Enter to process changes, End or Return to exit this function
```

. . . the values typed in each field will be gathered together, formatted and displayed as a valid MVS job card. The short message 'JOB CARDS GENERATED' appears in the upper right corner of the screen.

Silver Spring developers should use FFF instead of CCC, and PRTSS instead of PRT23; your actual sponsor code and project ID will vary; these are fictitious.

When satisfied that the job card is correct, press <end> and . . .

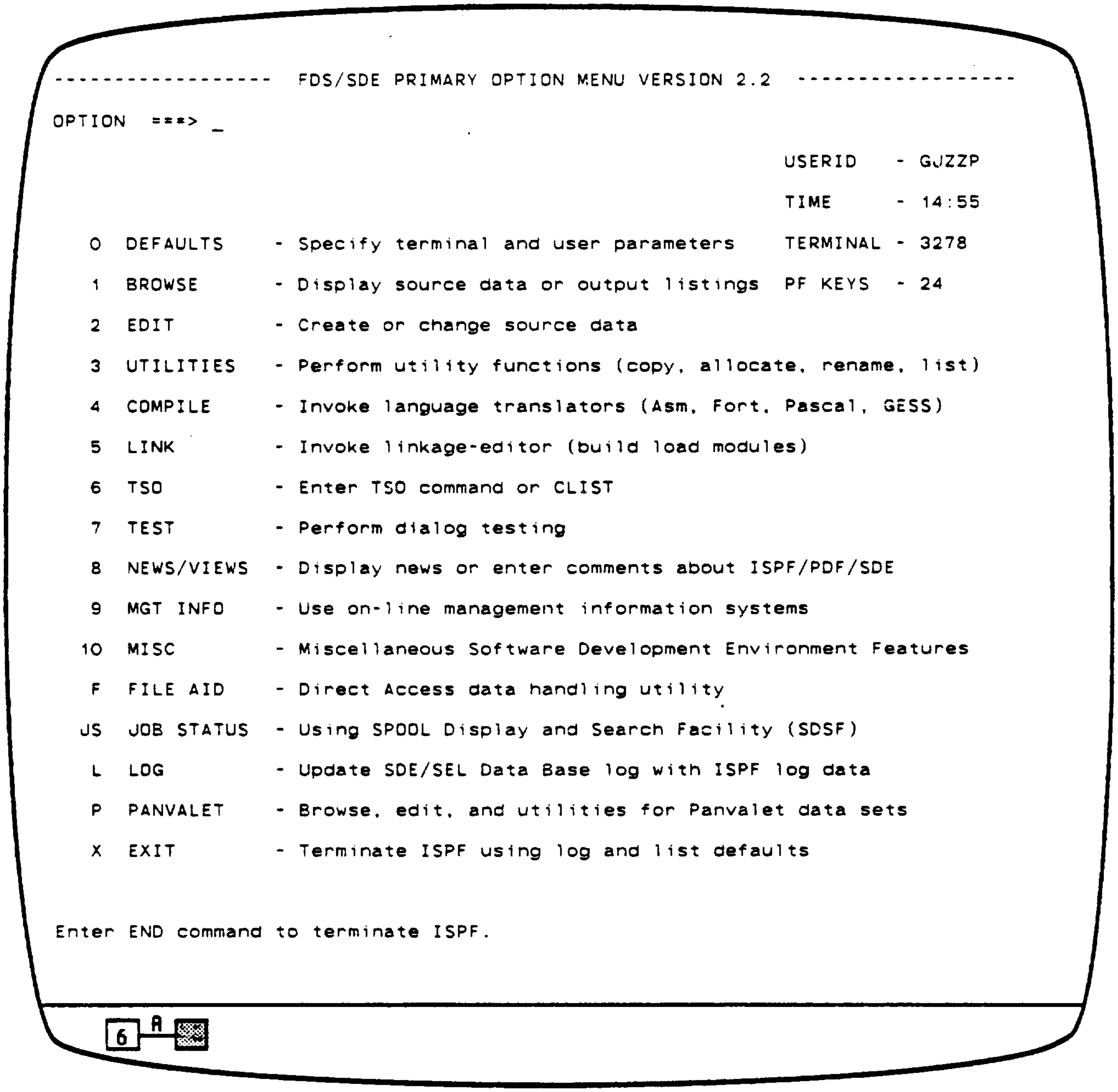
Scenario Step 2: Defining IBM Job Card



```
------------------------- FDS/SDE PARAMETER OPTIONS -------------------------
OPTION  ===> _

   1  TERMINAL    - Specify terminal characteristics
   2  LOG/LIST    - Specify ISPF log and list defaults
   3  PF KEYS     - Specify PF keys for 3278 terminal with 24 PF keys
   4  DISPLAY     - Specify screen display characteristics
   5  LIST        - Specify list data set characteristics
  JC  JOBCARD     - Specify Batch Job Card to be used
```

. . . we will go up one level to the DEFAULTS selection panel. Press <end> again and . . .

Scenario Step 3: Creating/Editing Panvalet member

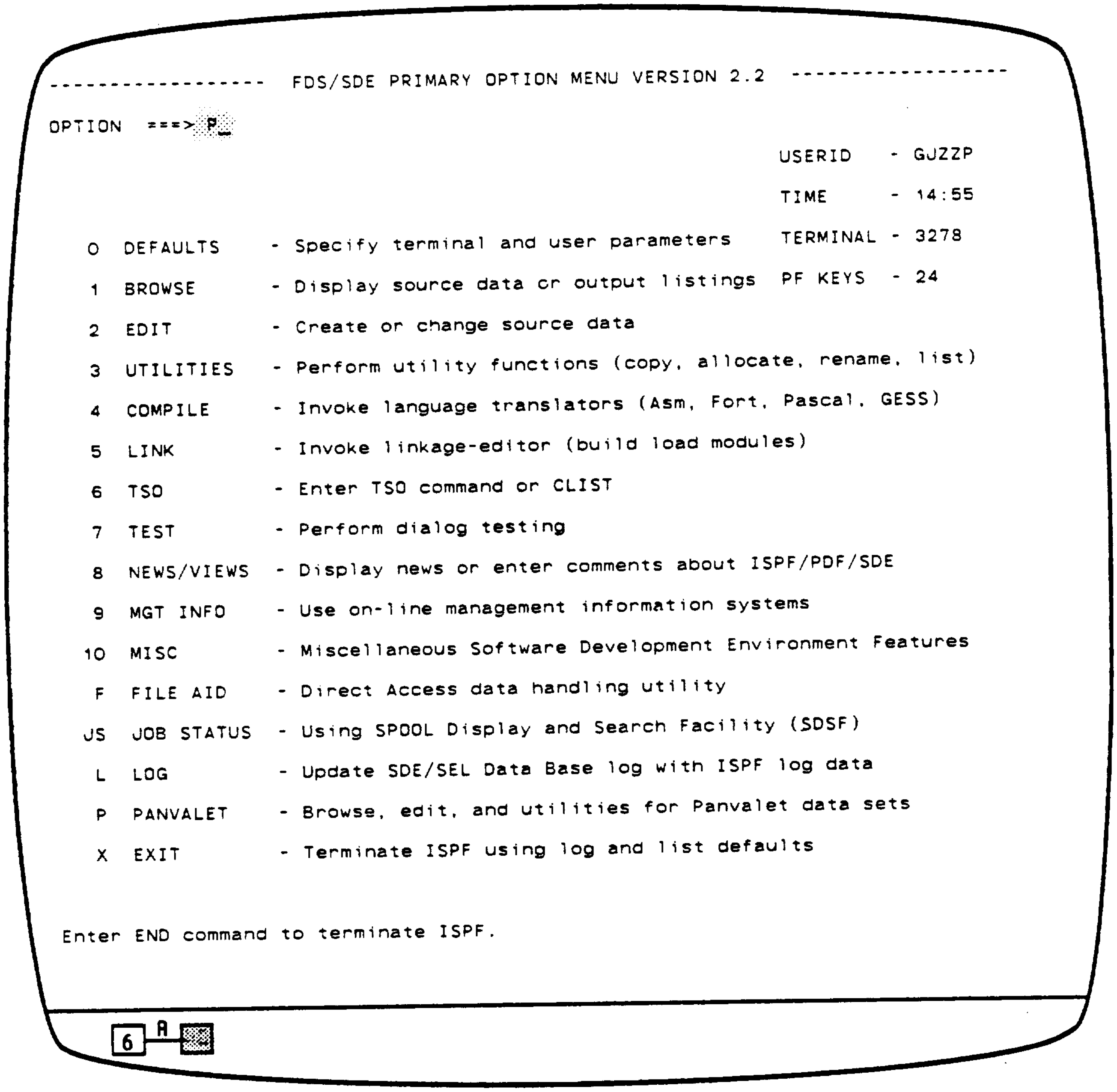


```
------------------  FDS/SDE PRIMARY OPTION MENU VERSION 2.2  ------------------

OPTION  ===> _
                                                            USERID   - GJZZP

                                                            TIME     - 14:55

   0  DEFAULTS     - Specify terminal and user parameters   TERMINAL - 3278

   1  BROWSE       - Display source data or output listings PF KEYS  - 24

   2  EDIT         - Create or change source data

   3  UTILITIES    - Perform utility functions (copy, allocate, rename, list)

   4  COMPILE      - Invoke language translators (Asm, Fort, Pascal, GESS)

   5  LINK         - Invoke linkage-editor (build load modules)

   6  TSO          - Enter TSO command or CLIST

   7  TEST         - Perform dialog testing

   8  NEWS/VIEWS   - Display news or enter comments about ISPF/PDF/SDE

   9  MGT INFO     - Use on-line management information systems

  10  MISC         - Miscellaneous Software Development Environment Features

   F  FILE AID     - Direct Access data handling utility

  JS  JOB STATUS   - Using SPOOL Display and Search Facility (SDSF)

   L  LOG          - Update SDE/SEL Data Base log with ISPF log data

   P  PANVALET     - Browse, edit, and utilities for Panvalet data sets

   X  EXIT         - Terminate ISPF using log and list defaults


Enter END command to terminate ISPF.

   6  A
```

. . . we are back at the primary option panel.

In order to perform step 3 of our scenario, we have to create a FORTRAN main routine in a Panvalet library. To do this, we shculd ...

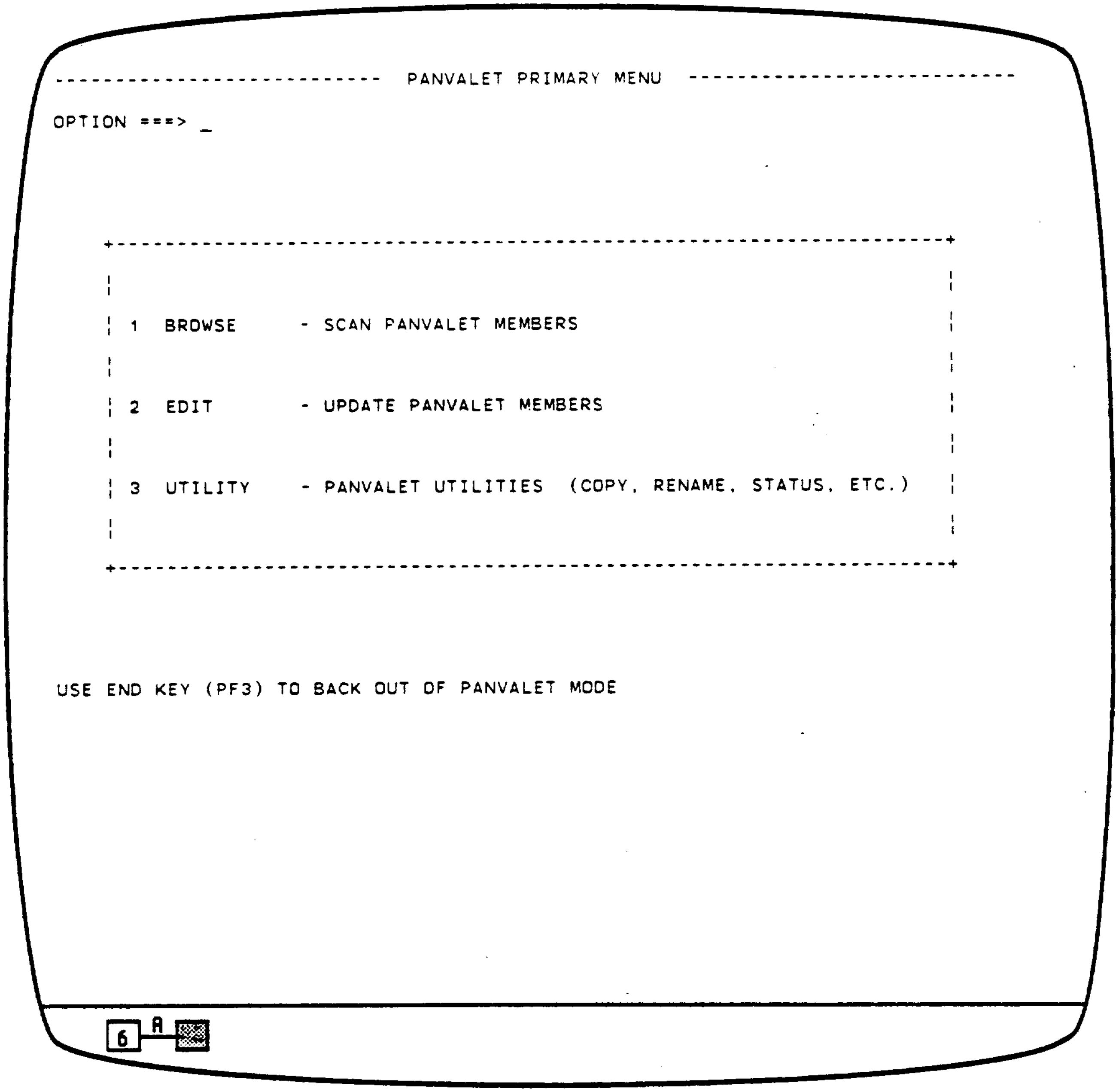
Scenario Step 3: Creating/Editing Panvalet member



```
------------------  FDS/SDE PRIMARY OPTION MENU VERSION 2.2  -----------------
OPTION  ===> P_
                                                         USERID   - GJZZP
                                                         TIME     - 14:55
   0  DEFAULTS     - Specify terminal and user parameters   TERMINAL - 3278
   1  BROWSE       - Display source data or output listings PF KEYS  - 24
   2  EDIT         - Create or change source data
   3  UTILITIES    - Perform utility functions (copy, allocate, rename, list)
   4  COMPILE      - Invoke language translators (Asm, Fort, Pascal, GESS)
   5  LINK         - Invoke linkage-editor (build load modules)
   6  TSO          - Enter TSO command or CLIST
   7  TEST         - Perform dialog testing
   8  NEWS/VIEWS   - Display news or enter comments about ISPF/PDF/SDE
   9  MGT INFO     - Use on-line management information systems
  10  MISC         - Miscellaneous Software Development Environment Features
   F  FILE AID     - Direct Access data handling utility
  JS  JOB STATUS   - Using SPOOL Display and Search Facility (SDSF)
   L  LOG          - Update SDE/SEL Data Base log with ISPF log data
   P  PANVALET     - Browse, edit, and utilities for Panvalet data sets
   X  EXIT         - Terminate ISPF using log and list defaults

 Enter END command to terminate ISPF.

     6 A
```

. . . type option ‘P’ and press the <enter> key, which . . .

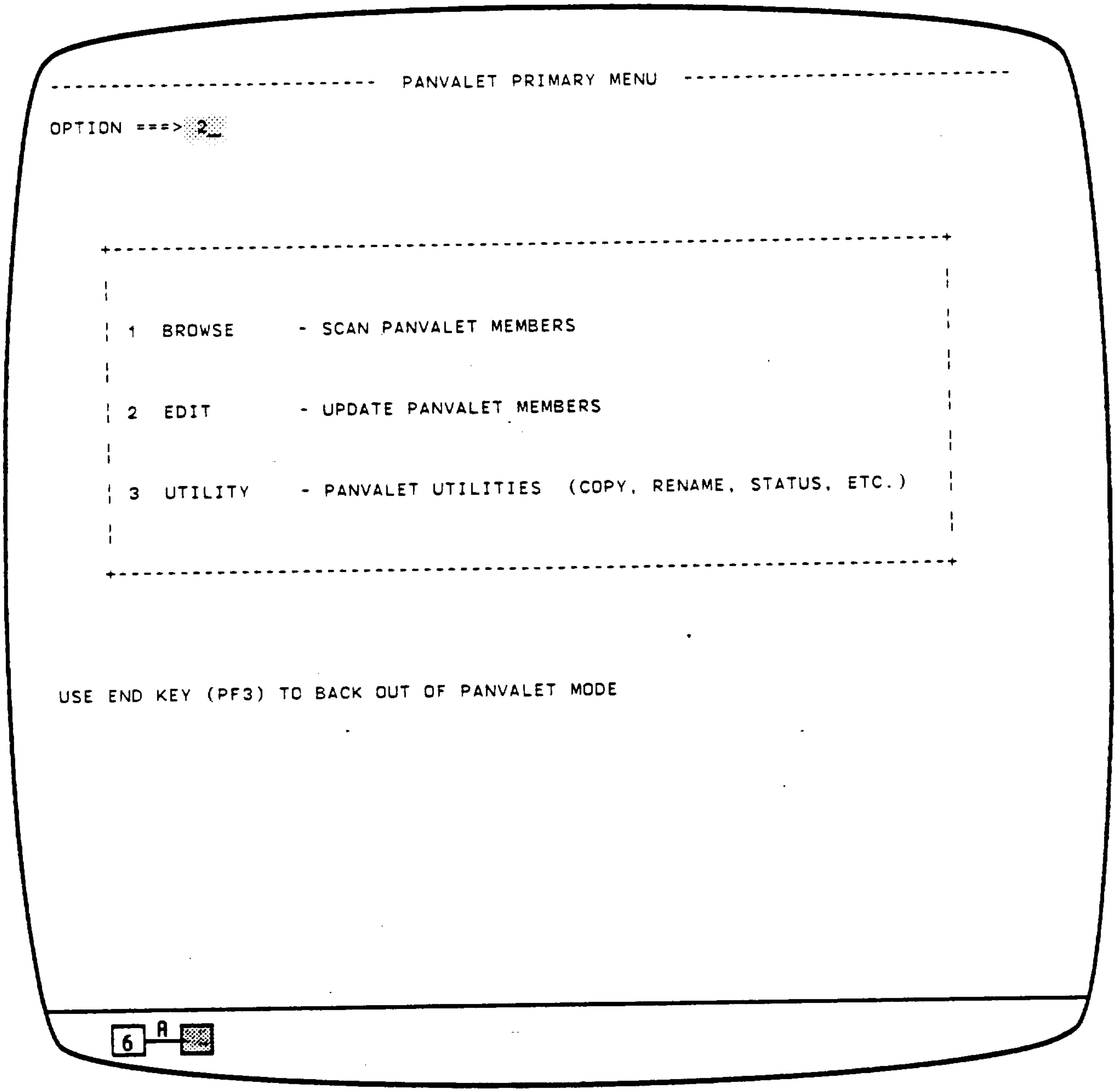
Scenario Step 3: Creating/Editing Panvalet member



```
-------------------------- PANVALET PRIMARY MENU --------------------------
OPTION ===> _



    +------------------------------------------------------------------+
    |                                                                  |
    | 1  BROWSE     - SCAN PANVALET MEMBERS                            |
    |                                                                  |
    | 2  EDIT       - UPDATE PANVALET MEMBERS                          |
    |                                                                  |
    | 3  UTILITY    - PANVALET UTILITIES  (COPY, RENAME, STATUS, ETC.) |
    |                                                                  |
    +------------------------------------------------------------------+


USE END KEY (PF3) TO BACK OUT OF PANVALET MODE
```

. . . displays the Panvalet function's selection panel. Panvalet is a data base management system that specializes in storing source code. It can also store data or anything else that can be grouped into 80- character records ("card-images", for trivia buffs) . Each separate entity is referred to as a 'member' in a Panvalet library. Members can be grouped by using the same first two characters as a prefix in a name.

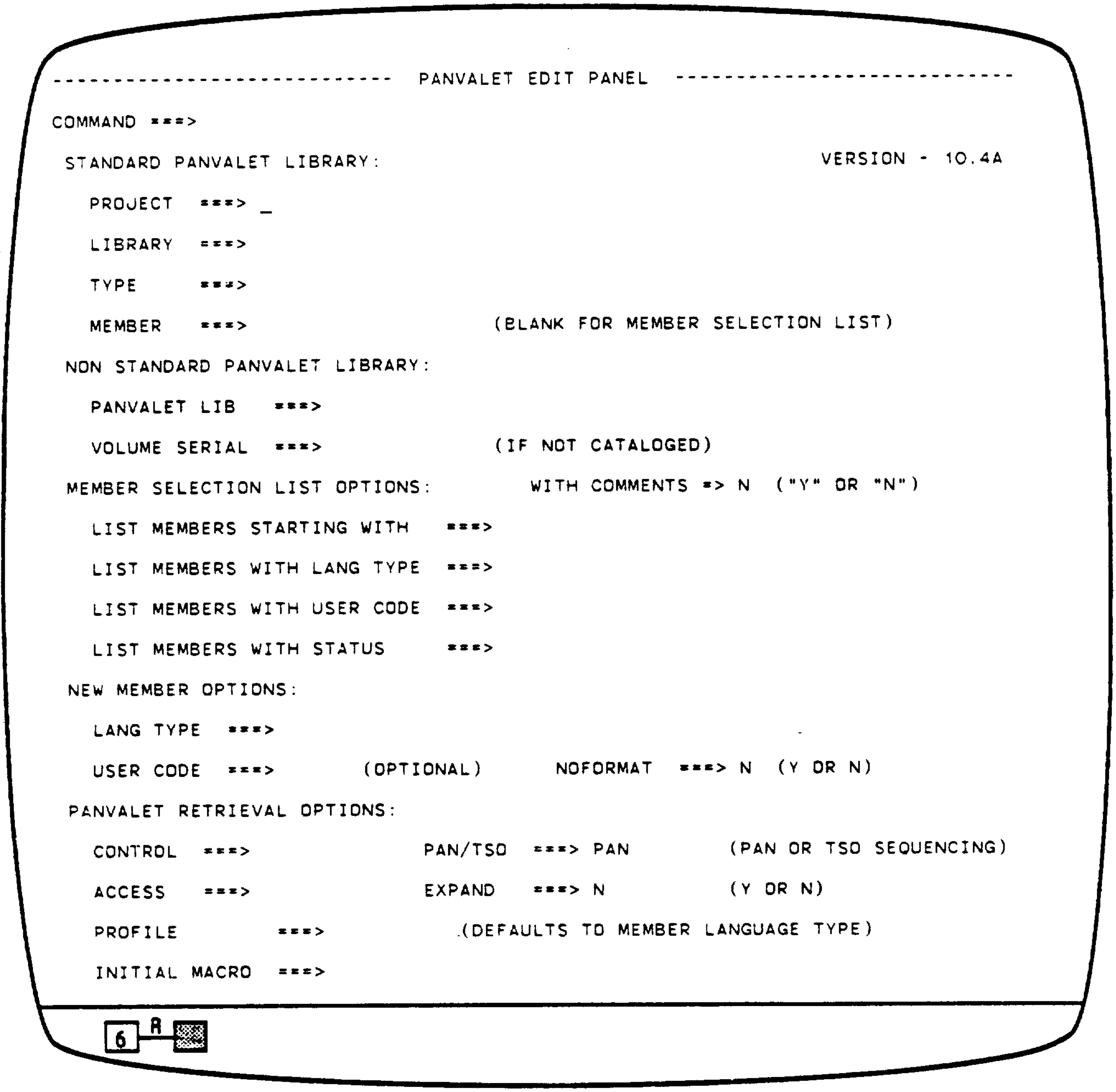
Scenario Step 3: Creating/Editing Panvalet member



```
---------------------------- PANVALET PRIMARY MENU ----------------------------
OPTION ===> 2_



    +-------------------------------------------------------------------------+
    |                                                                         |
    | 1  BROWSE     - SCAN PANVALET MEMBERS                                   |
    |                                                                         |
    | 2  EDIT       - UPDATE PANVALET MEMBERS                                 |
    |                                                                         |
    | 3  UTILITY    - PANVALET UTILITIES  (COPY, RENAME, STATUS, ETC.)        |
    |                                                                         |
    +-------------------------------------------------------------------------+



USE END KEY (PF3) TO BACK OUT OF PANVALET MODE
```

Select option 2, edit, and press <enter> . . .

Scenario Step 3: Creating/Editing Panvalet member

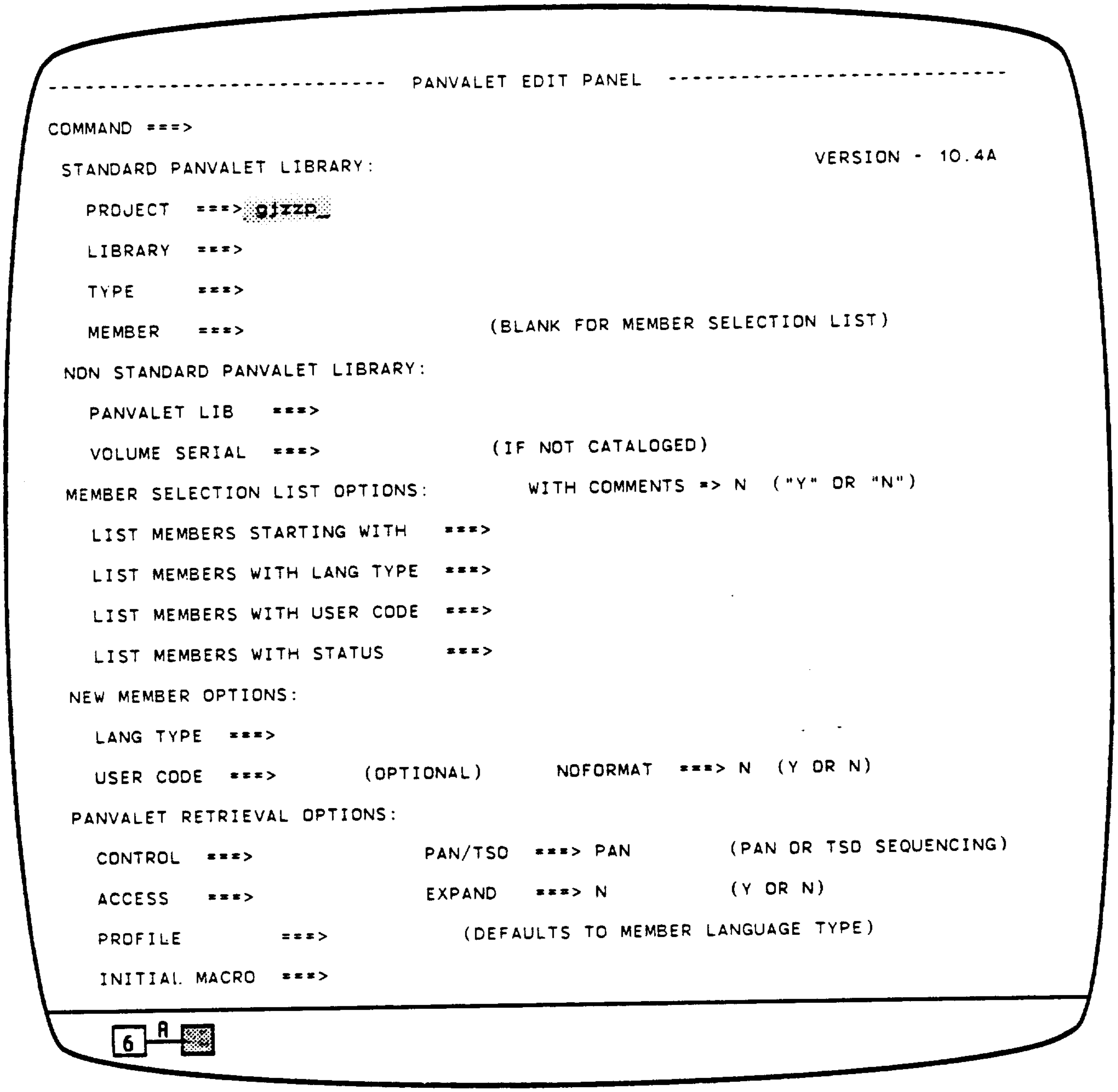


```
--------------------------- PANVALET EDIT PANEL ---------------------------
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                        VERSION - 10.4A
   PROJECT  ===> _
   LIBRARY  ===>
   TYPE     ===>
   MEMBER   ===>                 (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB    ===>
   VOLUME SERIAL   ===>          (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:     WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH    ===>
   LIST MEMBERS WITH LANG TYPE   ===>
   LIST MEMBERS WITH USER CODE   ===>
   LIST MEMBERS WITH STATUS      ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===>
   USER CODE  ===>      (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>             PAN/TSO  ===> PAN      (PAN OR TSO SEQUENCING)
   ACCESS   ===>             EXPAND   ===> N        (Y OR N)
   PROFILE        ===>        (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO  ===>
```

. . . and the Panvalet edit function's parameter entry panel is displayed. This panel is where we 'fill in the blanks' that provide the Panvalet edit function with the information it needs to begin processing.

We are going to create three new members in an existing Panvalet library that is named GJZZP.TEST.PAN. If you do not have a Panvalet library, use option 3.2.2 to create one.

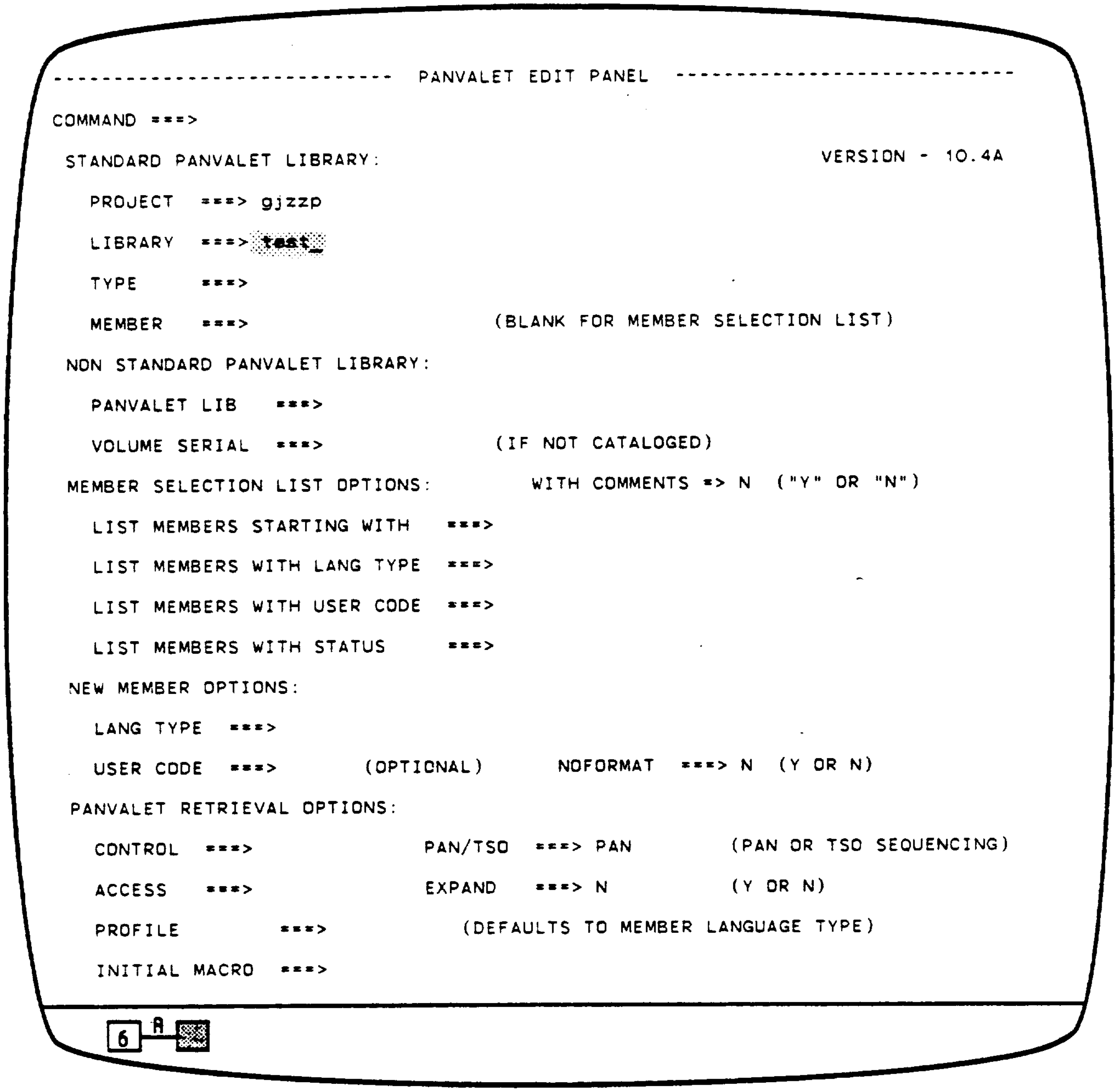
Scenario Step 3: Creating/Editing Panvalet member



```
---------------------------- PANVALET EDIT PANEL ----------------------------
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                   VERSION - 10.4A
   PROJECT  ===> gjzzp_
   LIBRARY  ===>
   TYPE     ===>
   MEMBER   ===>                  (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB   ===>
   VOLUME SERIAL  ===>            (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:       WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH   ===>
   LIST MEMBERS WITH LANG TYPE  ===>
   LIST MEMBERS WITH USER CODE  ===>
   LIST MEMBERS WITH STATUS     ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===>
   USER CODE  ===>       (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>             PAN/TSO  ===> PAN        (PAN OR TSO SEQUENCING)
   ACCESS   ===>             EXPAND   ===> N          (Y OR N)
   PROFILE        ===>          (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO  ===>
```

Grouping all members of the same system or subsystem by prefix allows us to refer to the group by prefix when using some of the SDE functions. Even though this scenario will not use the functions, we'll prefix the component names with 'TS' (for TeSt). Use the <NL> key to move the cursor to the PROJECT name qualifier parameter entry field and type in gjzzp . . .

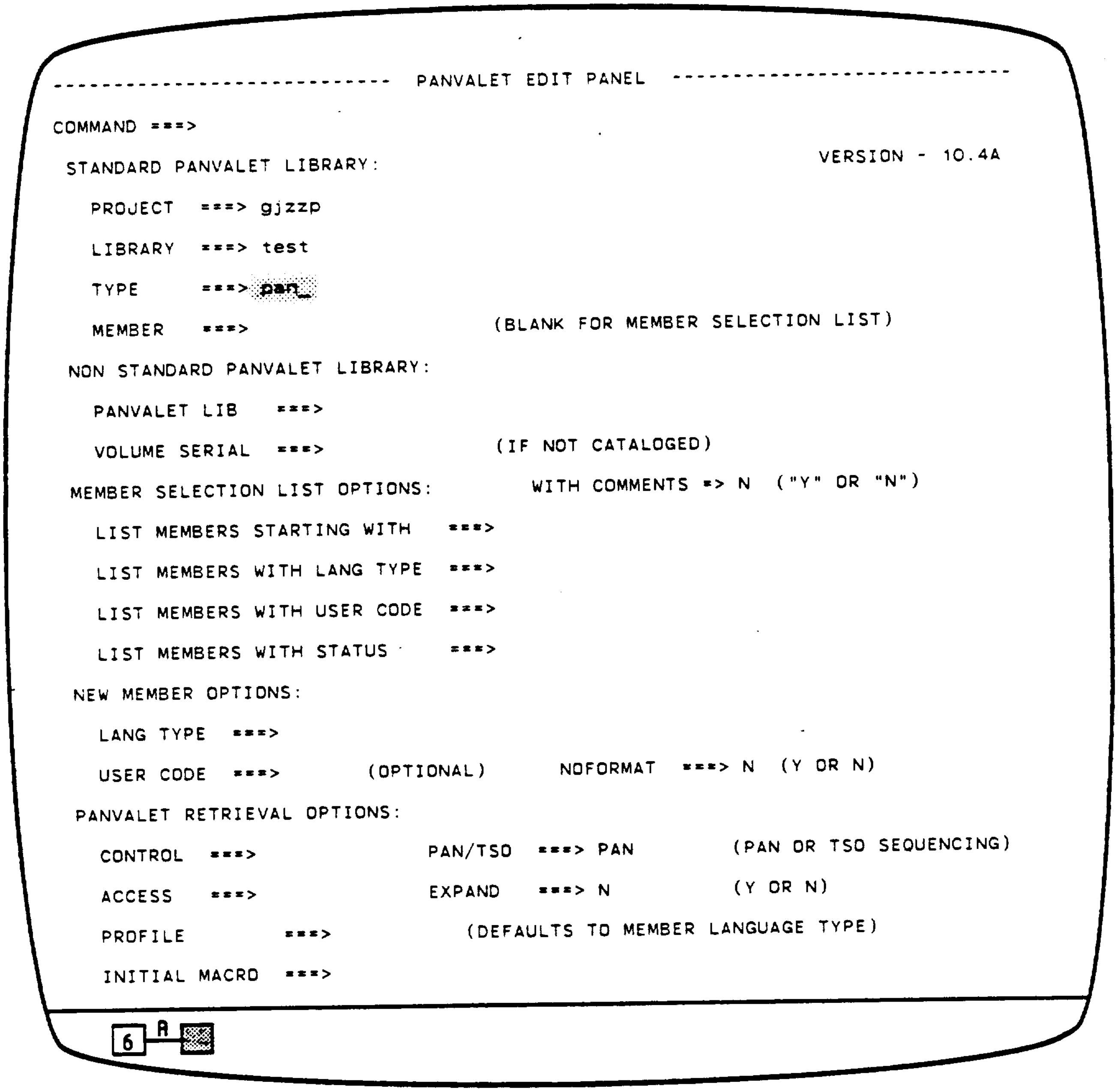
Scenario Step 3: Creating/Editing Panvalet member



```
---------------------------- PANVALET EDIT PANEL ----------------------------
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                     VERSION - 10.4A
   PROJECT  ===> gjzzp
   LIBRARY  ===> test_
   TYPE     ===>
   MEMBER   ===>                    (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB    ===>
   VOLUME SERIAL   ===>             (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:        WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH    ===>
   LIST MEMBERS WITH LANG TYPE   ===>
   LIST MEMBERS WITH USER CODE   ===>
   LIST MEMBERS WITH STATUS      ===>
 NEW MEMBER OPTIONS:
   LANG TYPE   ===>
   USER CODE   ===>       (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL   ===>              PAN/TSO  ===> PAN        (PAN OR TSO SEQUENCING)
   ACCESS    ===>              EXPAND   ===> N          (Y OR N)
   PROFILE         ===>           (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO   ===>
```

. . . Use the <NL> key to move to the LIBRARY parameter entry field and type in test . . .

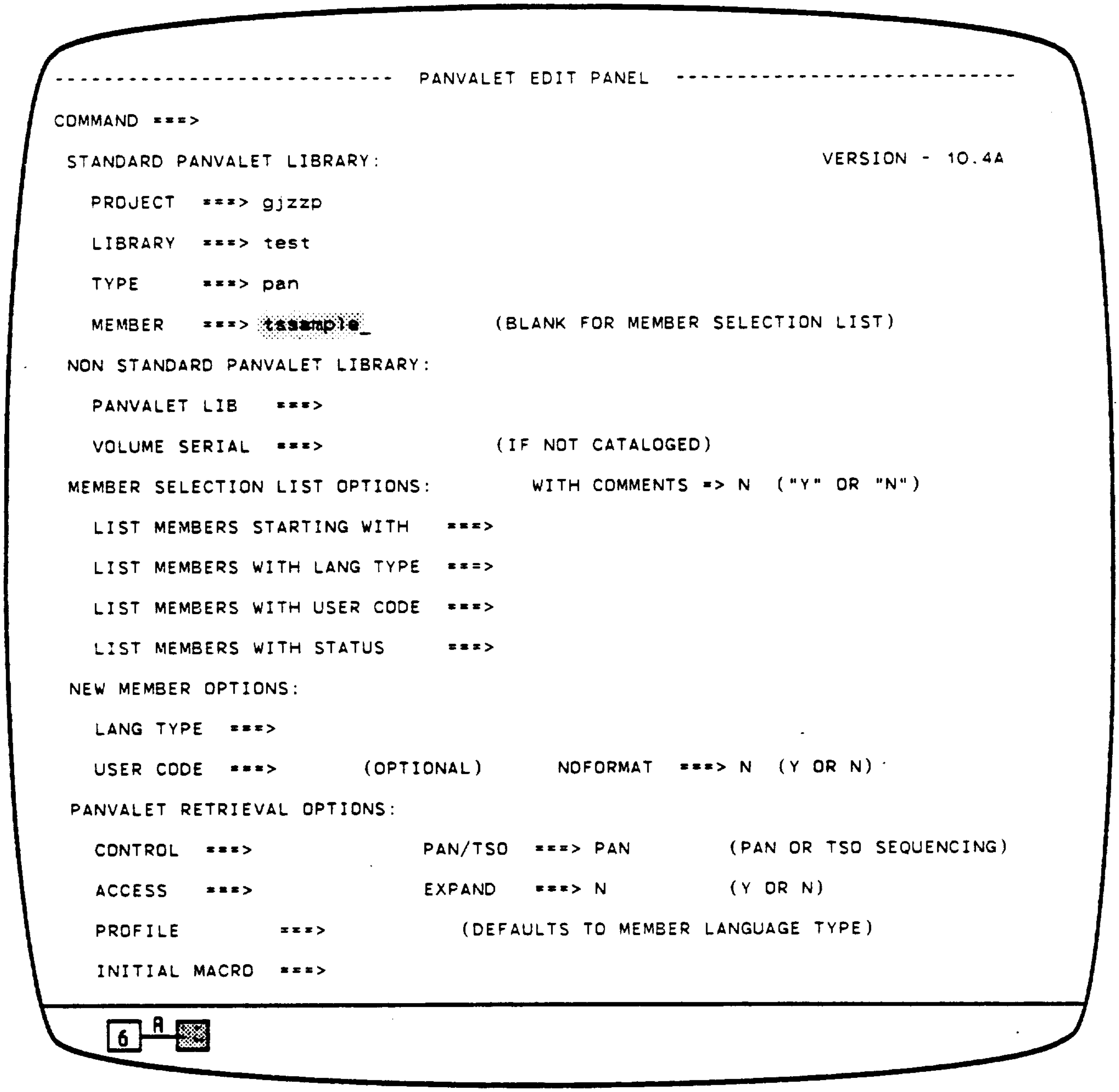
Scenario Step 3: Creating/Editing Panvalet member



```
---------------------------- PANVALET EDIT PANEL ----------------------------
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                          VERSION - 10.4A
   PROJECT  ===> gjzzp
   LIBRARY  ===> test
   TYPE     ===> pan_
   MEMBER   ===>                     (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB   ===>
   VOLUME SERIAL  ===>               (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:         WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH    ===>
   LIST MEMBERS WITH LANG TYPE   ===>
   LIST MEMBERS WITH USER CODE   ===>
   LIST MEMBERS WITH STATUS      ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===>
   USER CODE  ===>       (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>             PAN/TSO  ===> PAN       (PAN OR TSO SEQUENCING)
   ACCESS   ===>             EXPAND   ===> N         (Y OR N)
   PROFILE        ===>          (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO  ===>
```

. . . Use the <NL> key to move to the TYPE parameter entry field and type in pan . . .

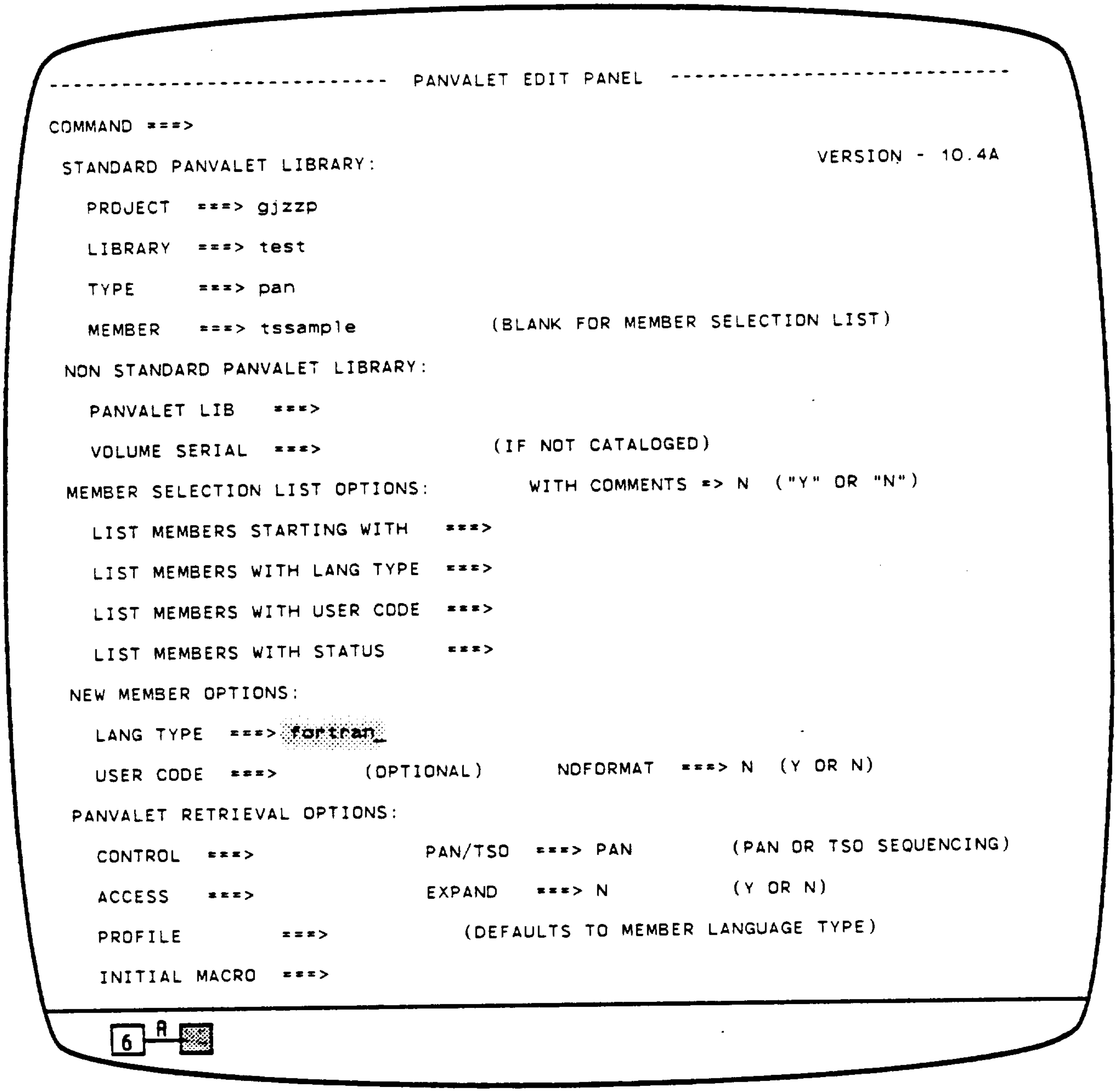
Scenario Step 3: Creating/Editing Panvalet member



```
---------------------------  PANVALET EDIT PANEL  ---------------------------
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                       VERSION - 10.4A
   PROJECT  ===> gjzzp
   LIBRARY  ===> test
   TYPE     ===> pan
   MEMBER   ===> tssample_         (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB   ===>
   VOLUME SERIAL  ===>             (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:        WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH   ===>
   LIST MEMBERS WITH LANG TYPE  ===>
   LIST MEMBERS WITH USER CODE  ===>
   LIST MEMBERS WITH STATUS     ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===>
   USER CODE  ===>       (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>               PAN/TSO  ===> PAN       (PAN OR TSO SEQUENCING)
   ACCESS   ===>               EXPAND   ===> N         (Y OR N)
   PROFILE        ===>           (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO  ===>
```

. . . Press <NL> to move to the MEMBER parameter entry field and type in the prefix and name combination tssample . . .

Scenario Step 3: Creating/Editing Panvalet member

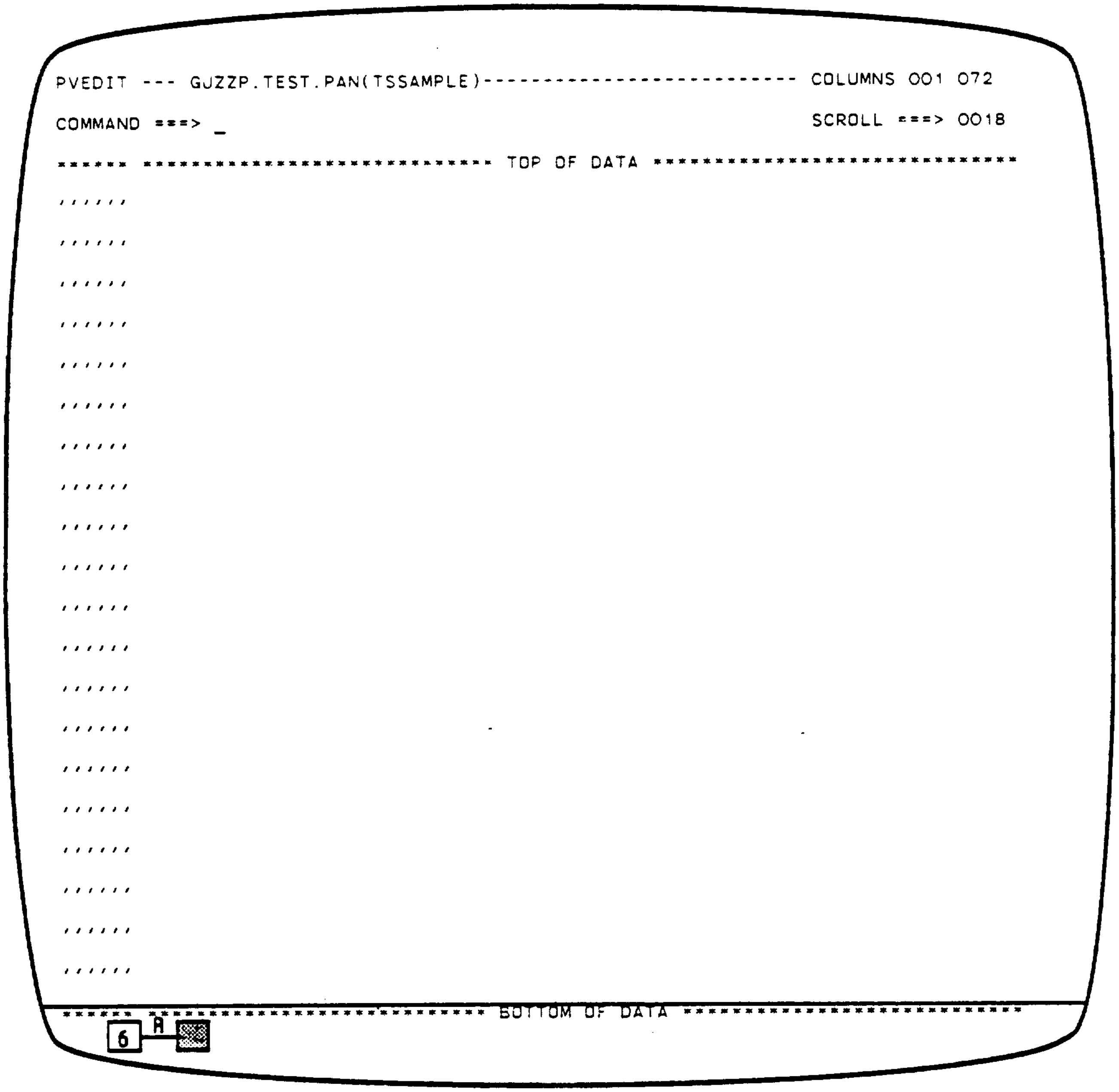


```
---------------------------- PANVALET EDIT PANEL ----------------------------
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                       VERSION - 10.4A
   PROJECT  ===> gjzzp
   LIBRARY  ===> test
   TYPE     ===> pan
   MEMBER   ===> tssample          (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB   ===>
   VOLUME SERIAL  ===>             (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:      WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH   ===>
   LIST MEMBERS WITH LANG TYPE  ===>
   LIST MEMBERS WITH USER CODE  ===>
   LIST MEMBERS WITH STATUS     ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===> fortran_
   USER CODE  ===>       (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>              PAN/TSO  ===> PAN        (PAN OR TSO SEQUENCING)
   ACCESS   ===>              EXPAND   ===> N          (Y OR N)
   PROFILE        ===>           (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO  ===>
```

. . . and finally <NL> seven times to the language type parameter entry field and type in fortran. The language type entry is only required when a new Panvalet library member is to be created.

Now, press the <enter> key . . .

Scenario Step 3: Creating/Editing Panvalet member

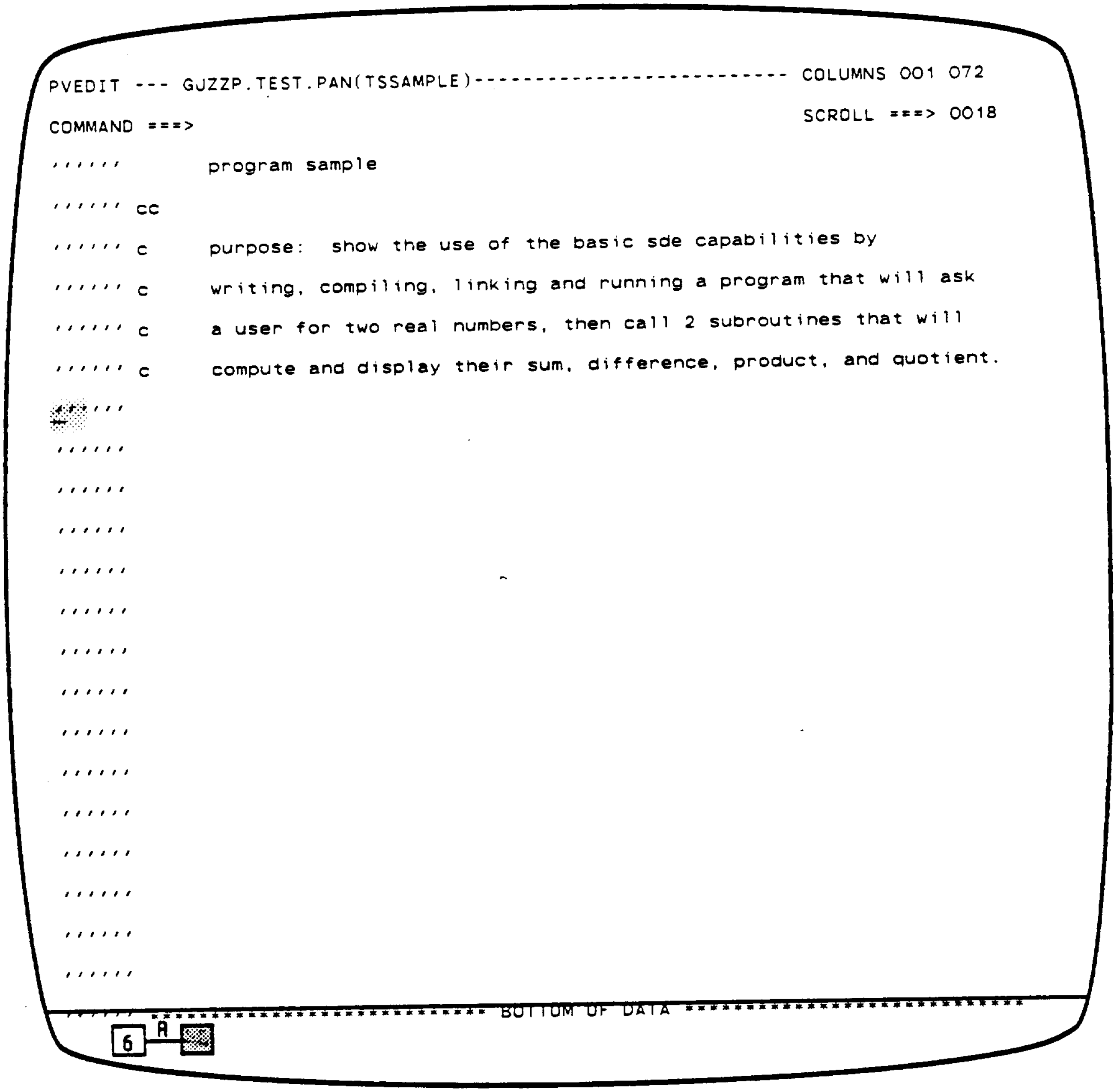


```
PVEDIT --- GJZZP.TEST.PAN(TSSAMPLE)------------------------- COLUMNS 001 072
COMMAND ===> _                                               SCROLL ===> 0018
****** ***************************** TOP OF DATA ******************************
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** ***************************** BOTTOM OF DATA ****************************
```

. . . and the Panvalet edit data entry panel is displayed. The dots on the left indicate new line positions. Now we can begin entering the program into memory.

Press <FT> at the end of each line . .

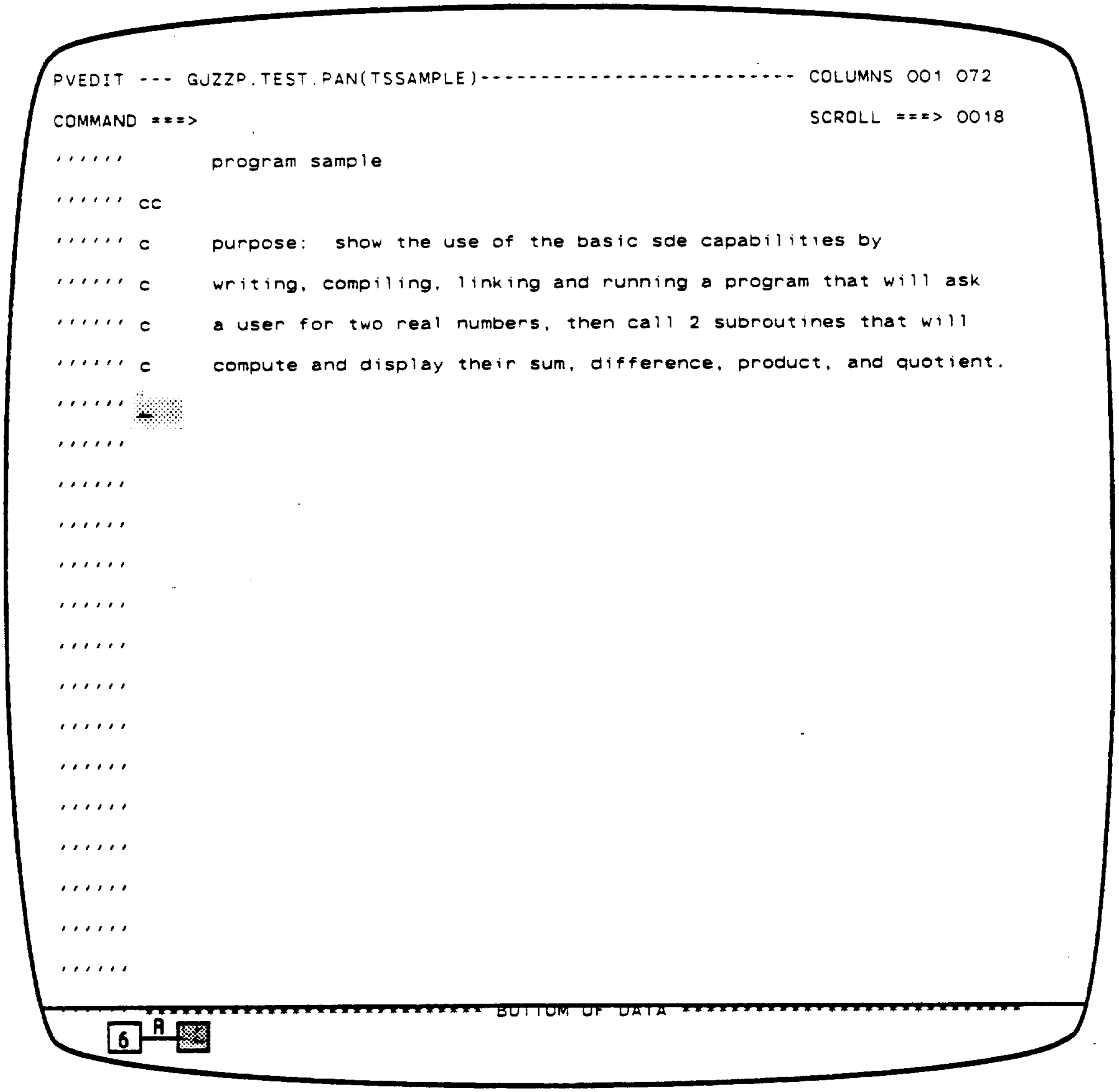
Scenario Step 3: Creating/Editing Panvalet member



```
PVEDIT --- GJZZP.TEST.PAN(TSSAMPLE)------------------------ COLUMNS 001 072
COMMAND ===>                                                SCROLL ===> 0018
''''''      program sample
'''''' cc
'''''' c    purpose:  show the use of the basic sde capabilities by
'''''' c    writing, compiling, linking and running a program that will ask
'''''' c    a user for two real numbers, then call 2 subroutines that will
'''''' c    compute and display their sum, difference, product, and quotient.
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
'''''' ****************************** BOTTOM OF DATA ******************************
```

. . . and the cursor will move to the far left of the next line. Press the <FT> key again . . .

Scenario Step 3: Creating/Editing Panvalet member

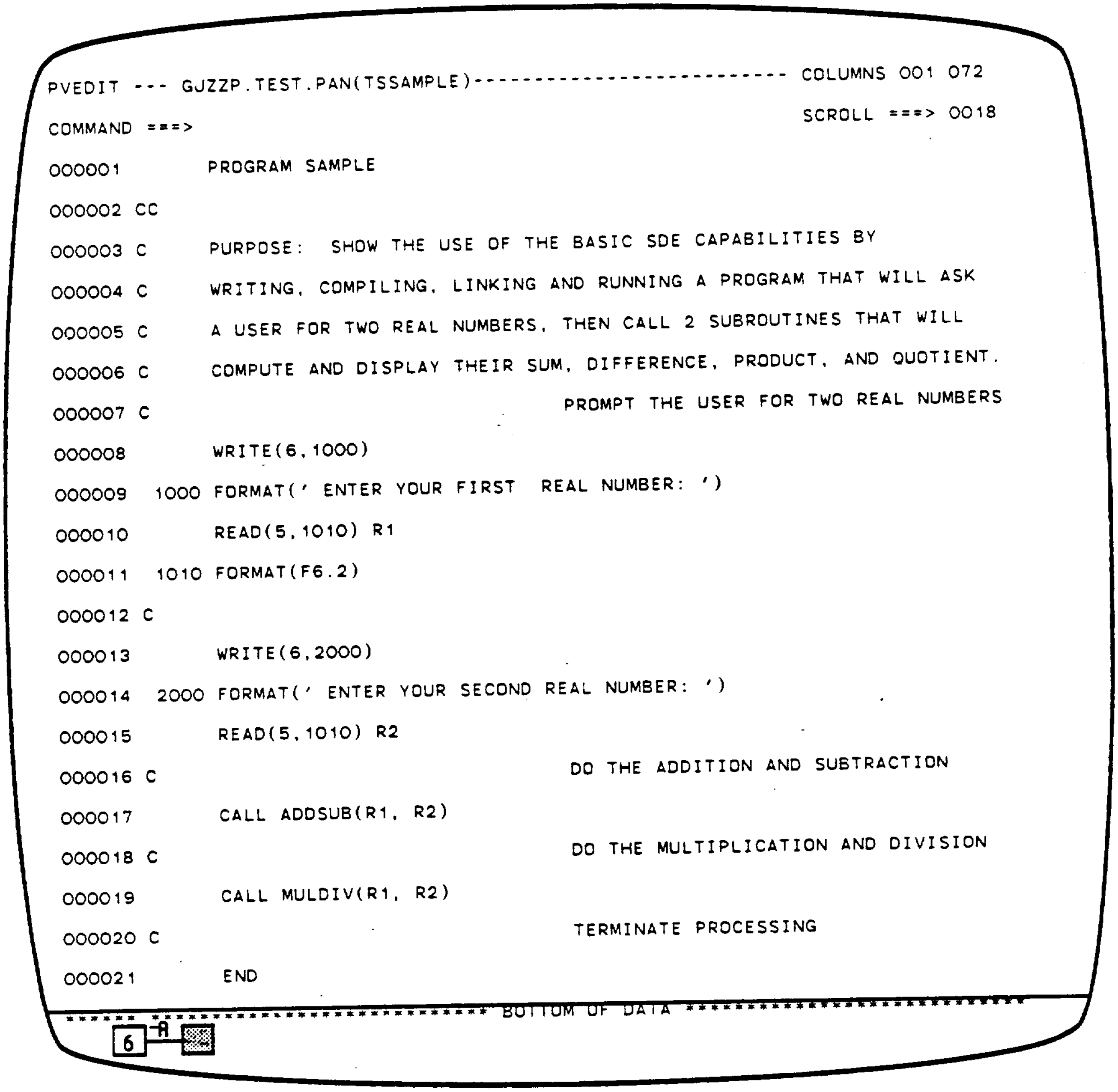


```
PVEDIT --- GJZZP.TEST.PAN(TSSAMPLE)------------------------- COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> 0018
''''''        program sample
'''''' cc
'''''' c      purpose:  show the use of the basic sde capabilities by
'''''' c      writing, compiling, linking and running a program that will ask
'''''' c      a user for two real numbers, then call 2 subroutines that will
'''''' c      compute and display their sum, difference, product, and quotient.
'''''' _
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** ***************************** BOTTOM OF DATA *****************************
```

. . . and the cursor will move to the first column of the data entry area. Continue to type the program into the computer.

When the last line of the program has been typed in, press <enter> and . . .

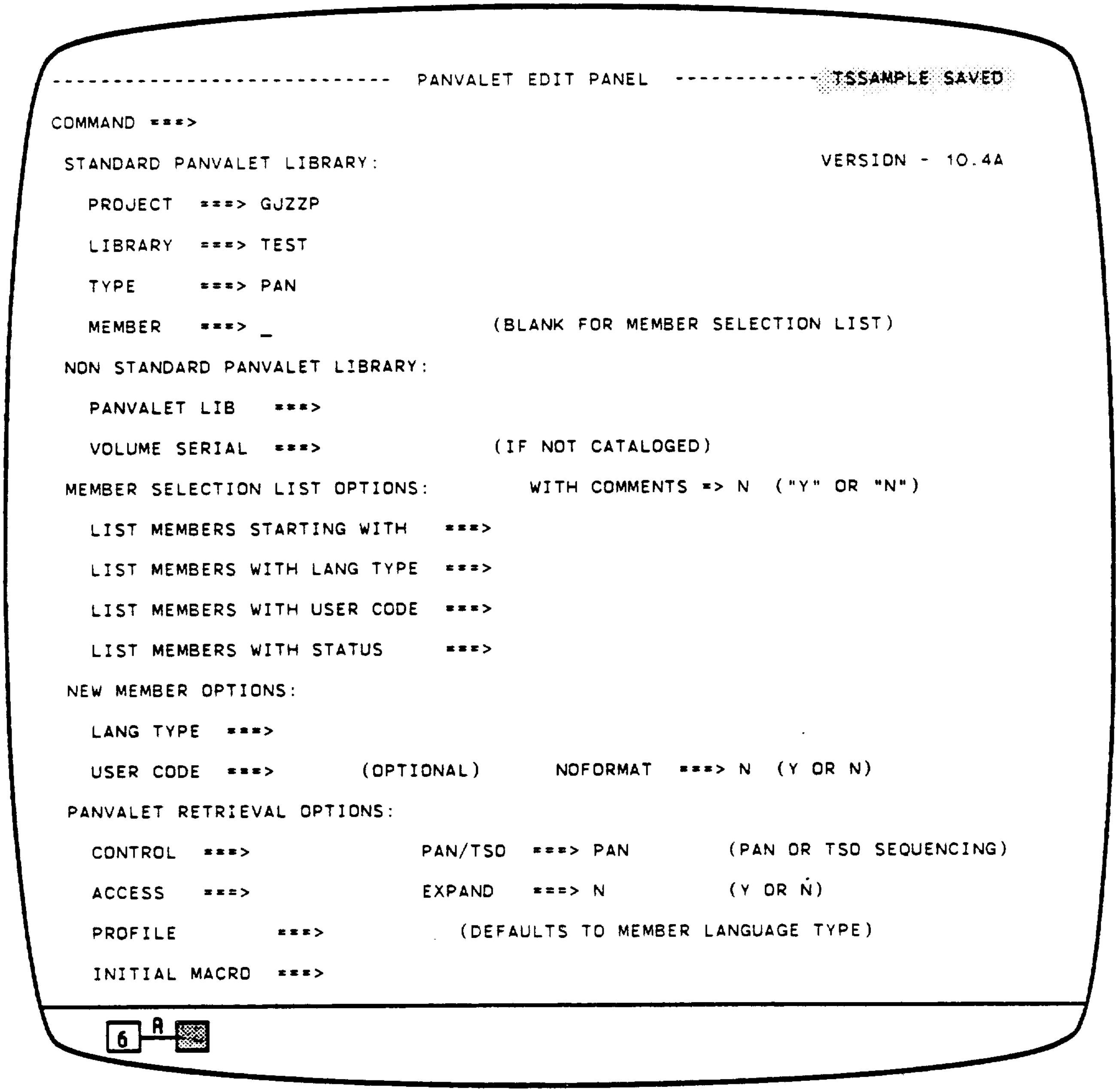
Scenario Step 3: Creating/Editing Panvalet member



```
PVEDIT --- GJZZP.TEST.PAN(TSSAMPLE)------------------------- COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> 0018
000001       PROGRAM SAMPLE
000002 CC
000003 C     PURPOSE:  SHOW THE USE OF THE BASIC SDE CAPABILITIES BY
000004 C     WRITING, COMPILING, LINKING AND RUNNING A PROGRAM THAT WILL ASK
000005 C     A USER FOR TWO REAL NUMBERS, THEN CALL 2 SUBROUTINES THAT WILL
000006 C     COMPUTE AND DISPLAY THEIR SUM, DIFFERENCE, PRODUCT, AND QUOTIENT.
000007 C                               PROMPT THE USER FOR TWO REAL NUMBERS
000008       WRITE(6,1000)
000009  1000 FORMAT(' ENTER YOUR FIRST  REAL NUMBER: ')
000010       READ(5,1010) R1
000011  1010 FORMAT(F6.2)
000012 C
000013       WRITE(6,2000)
000014  2000 FORMAT(' ENTER YOUR SECOND REAL NUMBER: ')
000015       READ(5,1010) R2
000016 C                               DO THE ADDITION AND SUBTRACTION
000017       CALL ADDSUB(R1, R2)
000018 C                               DO THE MULTIPLICATION AND DIVISION
000019       CALL MULDIV(R1, R2)
000020 C                               TERMINATE PROCESSING
000021       END
****** ***************************** BOTTOM OF DATA *****************************
```

. . . we notice two things have happened. All of the text has been converted to upper case, and line numbers have appeared where the dots were. The program must now be saved. Press the <end> key . . .

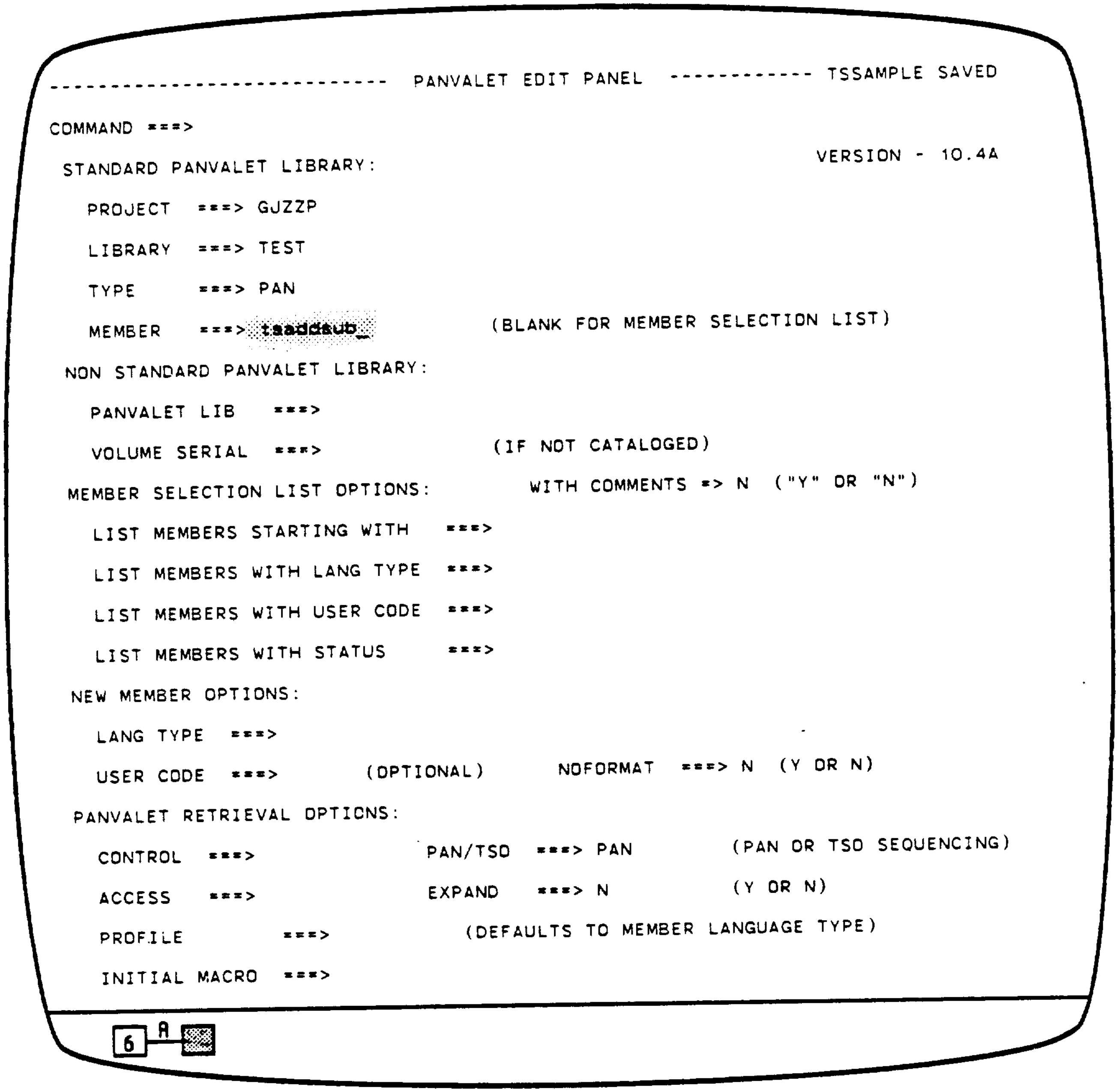
Scenario Step 3: Creating/Editing Panvalet member



```
----------------------------  PANVALET EDIT PANEL  ------------TSSAMPLE SAVED
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                      VERSION - 10.4A
   PROJECT   ===> GJZZP
   LIBRARY   ===> TEST
   TYPE      ===> PAN
   MEMBER    ===> _                 (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB   ===>
   VOLUME SERIAL  ===>              (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:        WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH   ===>
   LIST MEMBERS WITH LANG TYPE  ===>
   LIST MEMBERS WITH USER CODE  ===>
   LIST MEMBERS WITH STATUS     ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===>
   USER CODE  ===>       (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>                PAN/TSO   ===> PAN        (PAN OR TSO SEQUENCING)
   ACCESS   ===>                EXPAND    ===> N          (Y OR N)
   PROFILE        ===>            (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO  ===>
```

. . . and the Panvalet edit parameter entry panel will be redisplayed. Note the message 'TSSAMPLE SAVED' in the upper right corner of the screen. This is an example of a short message.

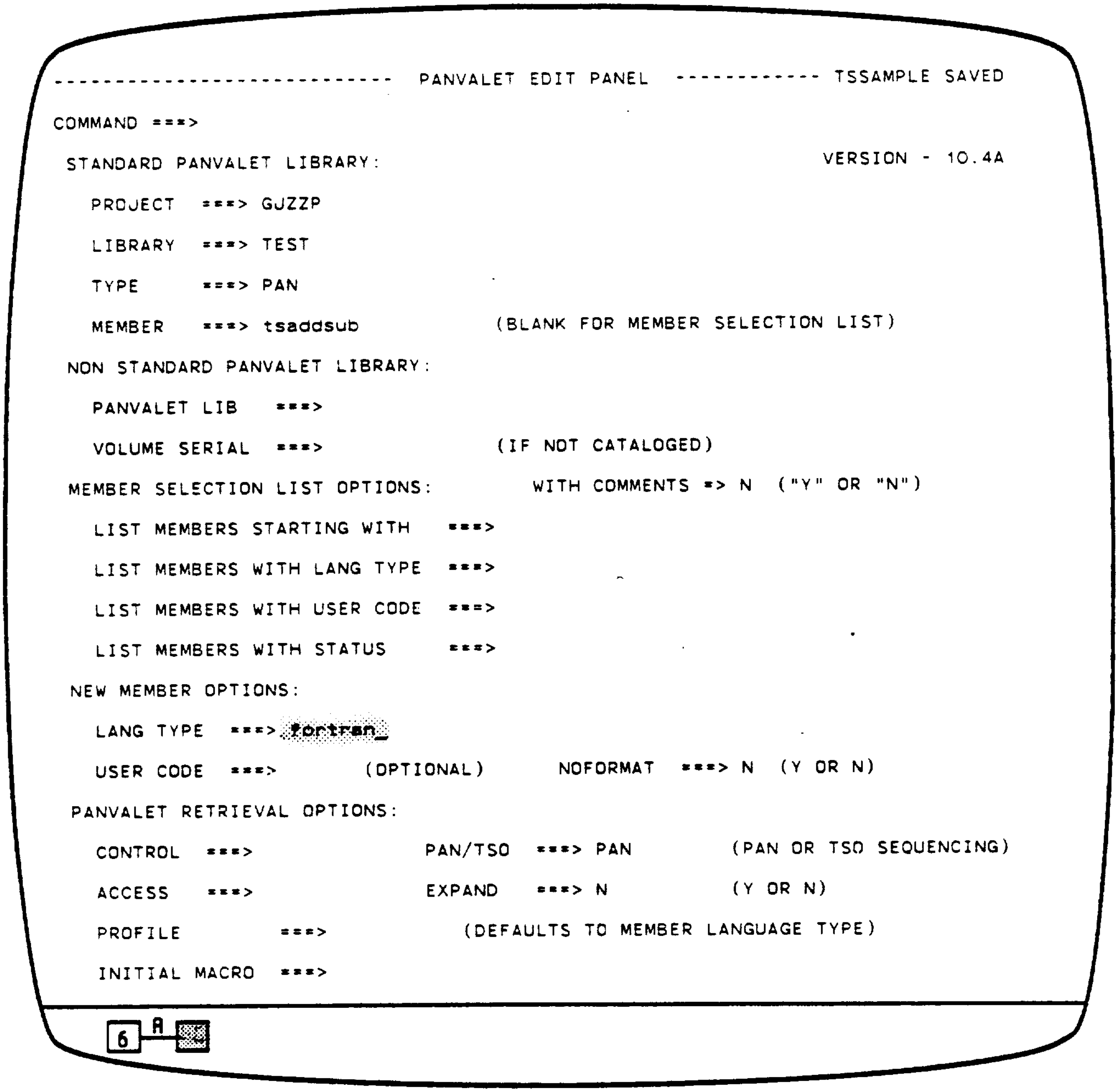
Scenario Step 3: Creating/Editing Panvalet member



```
--------------------------- PANVALET EDIT PANEL ------------ TSSAMPLE SAVED
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                   VERSION - 10.4A
   PROJECT   ===> GJZZP
   LIBRARY   ===> TEST
   TYPE      ===> PAN
   MEMBER    ===> tsaddsub_          (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB   ===>
   VOLUME SERIAL  ===>               (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:         WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH   ===>
   LIST MEMBERS WITH LANG TYPE  ===>
   LIST MEMBERS WITH USER CODE  ===>
   LIST MEMBERS WITH STATUS     ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===>
   USER CODE  ===>       (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>               PAN/TSO  ===> PAN       (PAN OR TSO SEQUENCING)
   ACCESS   ===>               EXPAND   ===> N         (Y OR N)
   PROFILE        ===>           (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO  ===>
```

Next we want to create the two subroutines that will be called by our main routine. Type in the prefix/name combination tsaddsub, . . .

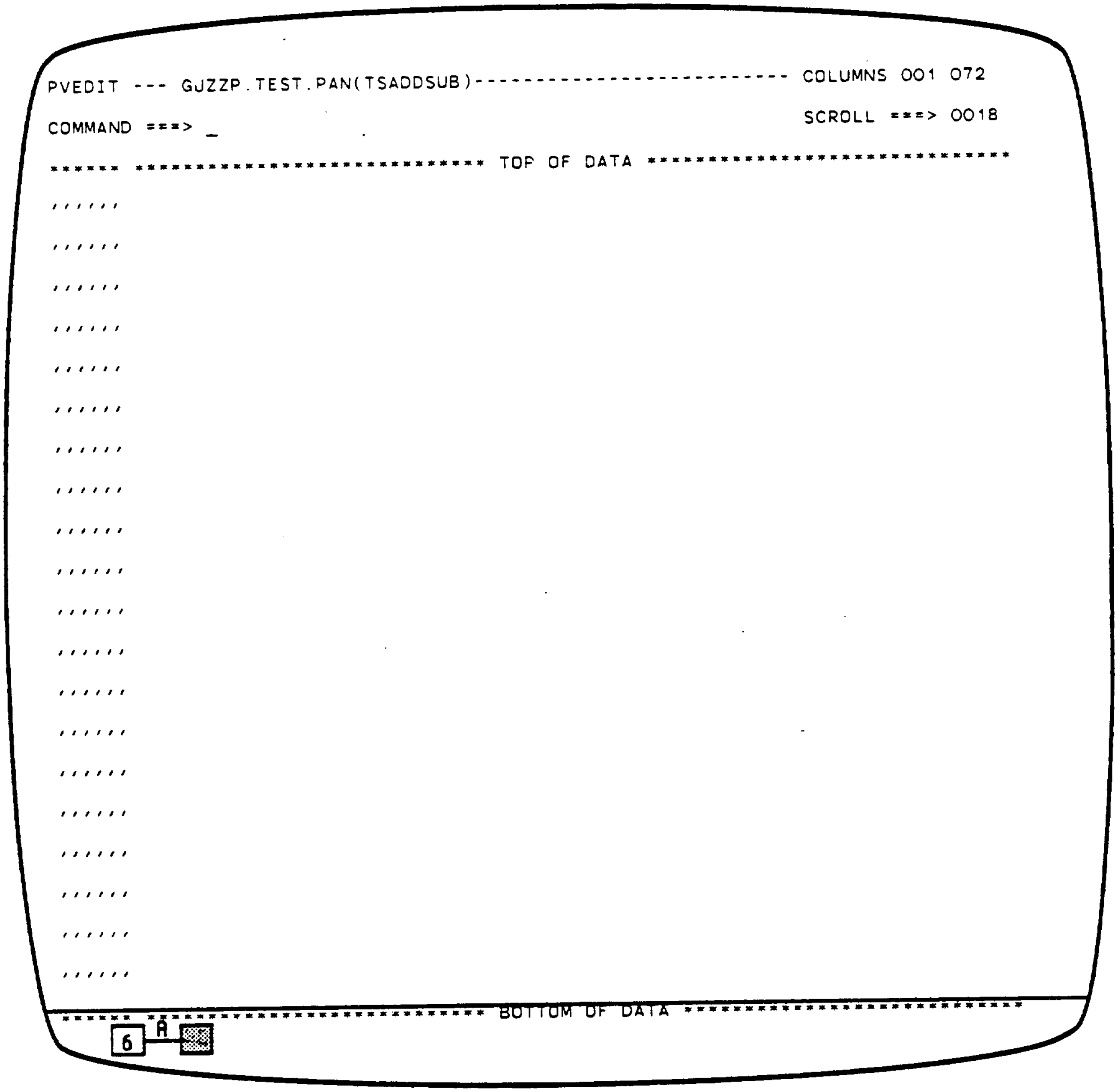
Scenario Step 3: Creating/Editing Panvalet member



```
---------------------------  PANVALET EDIT PANEL  ------------ TSSAMPLE SAVED
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                   VERSION - 10.4A
   PROJECT  ===> GJZZP
   LIBRARY  ===> TEST
   TYPE     ===> PAN
   MEMBER   ===> tsaddsub          (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB   ===>
   VOLUME SERIAL  ===>             (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:       WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH   ===>
   LIST MEMBERS WITH LANG TYPE  ===>
   LIST MEMBERS WITH USER CODE  ===>
   LIST MEMBERS WITH STATUS     ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===> fortran_
   USER CODE  ===>        (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>              PAN/TSO  ===> PAN        (PAN OR TSO SEQUENCING)
   ACCESS   ===>              EXPAND   ===> N          (Y OR N)
   PROFILE        ===>           (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO  ===>
```

. . . <NL> seven times to the language type field; type in fortran; and press <enter> . . .

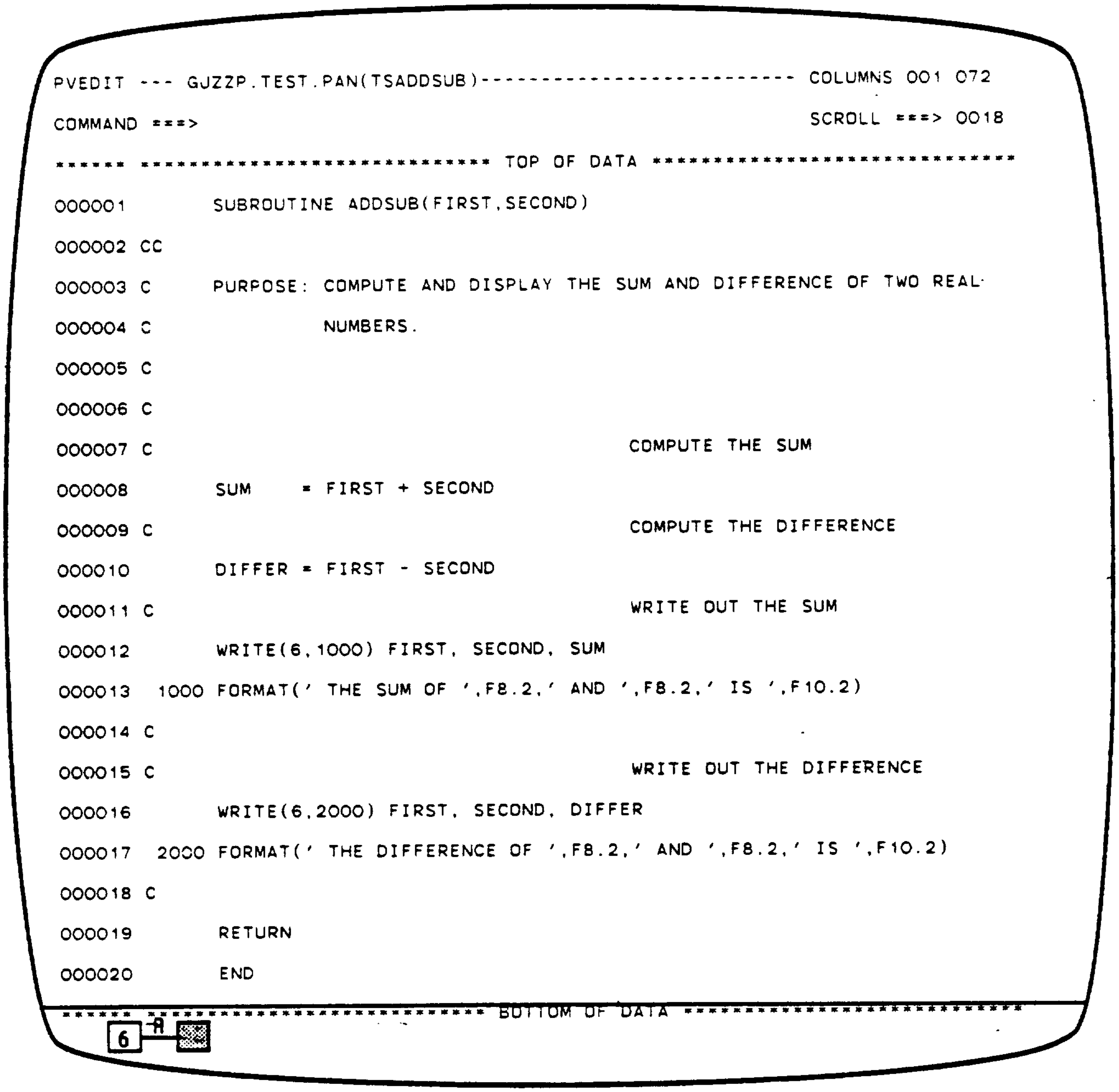
Scenario Step 3: Creating/Editing Panvalet member



```
PVEDIT --- GJZZP.TEST.PAN(TSADDSUB)------------------------- COLUMNS 001 072
COMMAND ===> _                                               SCROLL ===> 0018
****** ***************************** TOP OF DATA ******************************
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** **************************** BOTTOM OF DATA ****************************
```

. . . and we are again in the Panvalet edit data entry mode. Type in the subroutine just like we did the main routine; press <enter> . . .

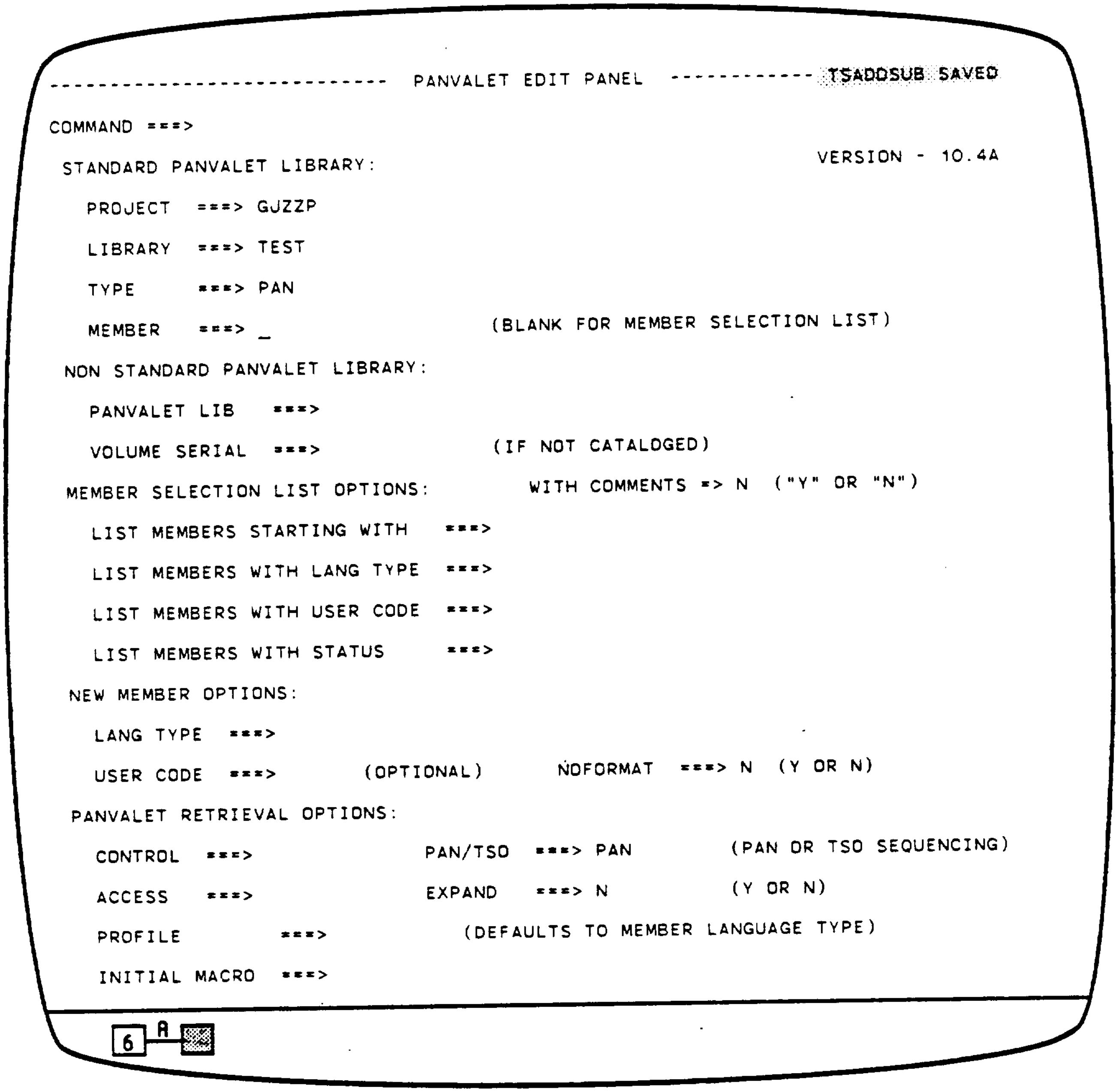
Scenario Step 3: Creating/Editing Panvalet member



```
PVEDIT --- GJZZP.TEST.PAN(TSADDSUB)------------------------- COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> 0018
****** **************************** TOP OF DATA *****************************
      SUBROUTINE ADDSUB(FIRST,SECOND)
CC
C     PURPOSE: COMPUTE AND DISPLAY THE SUM AND DIFFERENCE OF TWO REAL·
C              NUMBERS.
C
C
C                                  COMPUTE THE SUM
      SUM    = FIRST + SECOND
C                                  COMPUTE THE DIFFERENCE
      DIFFER = FIRST - SECOND
C                                  WRITE OUT THE SUM
      WRITE(6,1000) FIRST, SECOND, SUM
 1000 FORMAT(' THE SUM OF ',F8.2,' AND ',F8.2,' IS ',F10.2)
C
C                                  WRITE OUT THE DIFFERENCE
      WRITE(6,2000) FIRST, SECOND, DIFFER
 2000 FORMAT(' THE DIFFERENCE OF ',F8.2,' AND ',F8.2,' IS ',F10.2)
C
      RETURN
      END
****** **************************** BOTTOM OF DATA ***************************
```

. . . and we have our first subroutine finished. Press <end> . . .

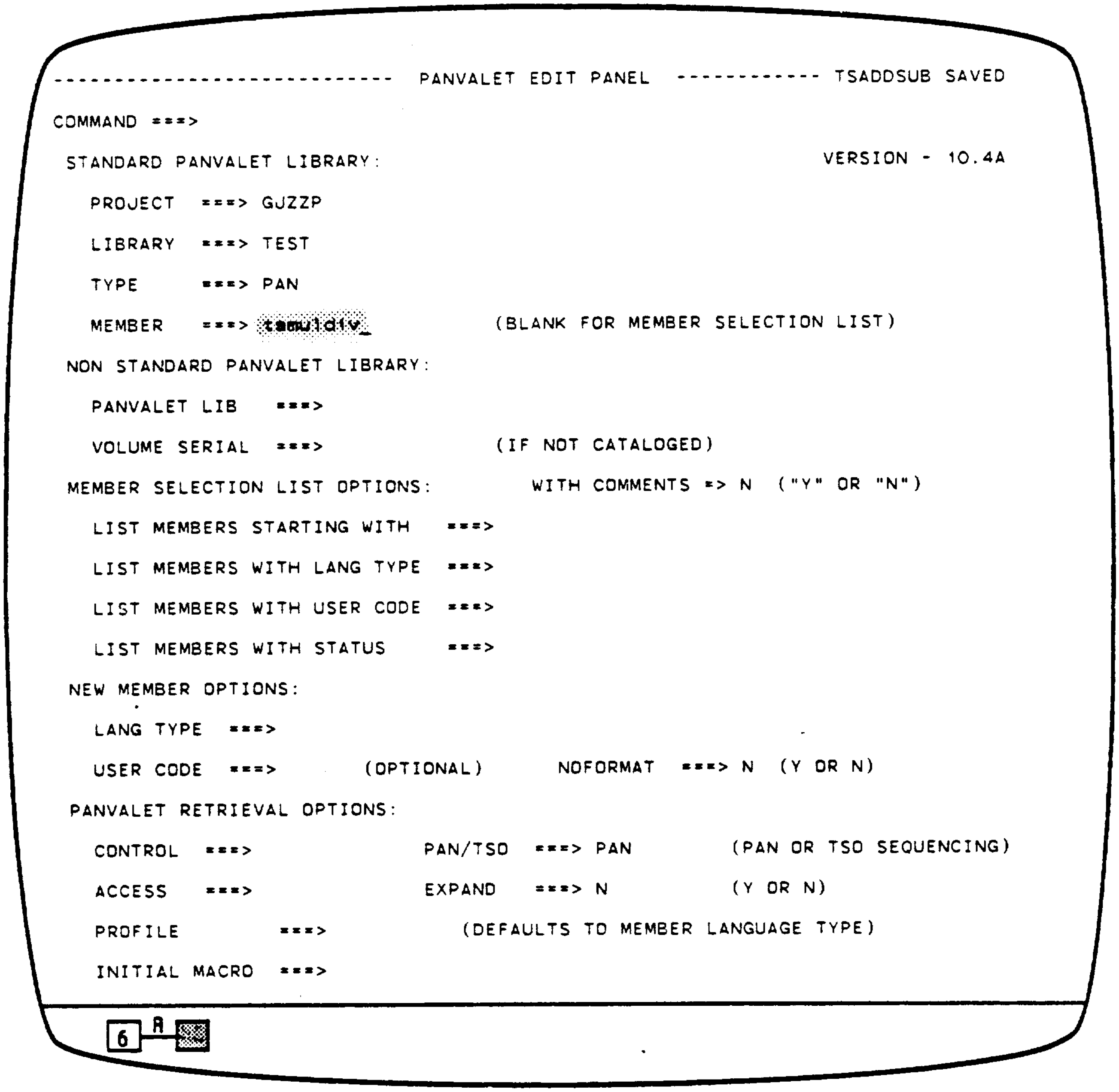
Scenario Step 3: Creating/Editing Panvalet member



```
--------------------------- PANVALET EDIT PANEL ------------ TSADDSUB SAVED
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                    VERSION - 10.4A
   PROJECT  ===> GJZZP
   LIBRARY  ===> TEST
   TYPE     ===> PAN
   MEMBER   ===> _                 (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB   ===>
   VOLUME SERIAL  ===>             (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:       WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH   ===>
   LIST MEMBERS WITH LANG TYPE  ===>
   LIST MEMBERS WITH USER CODE  ===>
   LIST MEMBERS WITH STATUS     ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===>
   USER CODE  ===>       (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>               PAN/TSO  ===> PAN       (PAN OR TSO SEQUENCING)
   ACCESS   ===>               EXPAND   ===> N         (Y OR N)
   PROFILE        ===>            (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO  ===>
   6 A ▨
```

. . . and we are back in the Panvalet parameter entry panel. Note the 'TSADDSUB SAVED' message in the upper right of the panel.

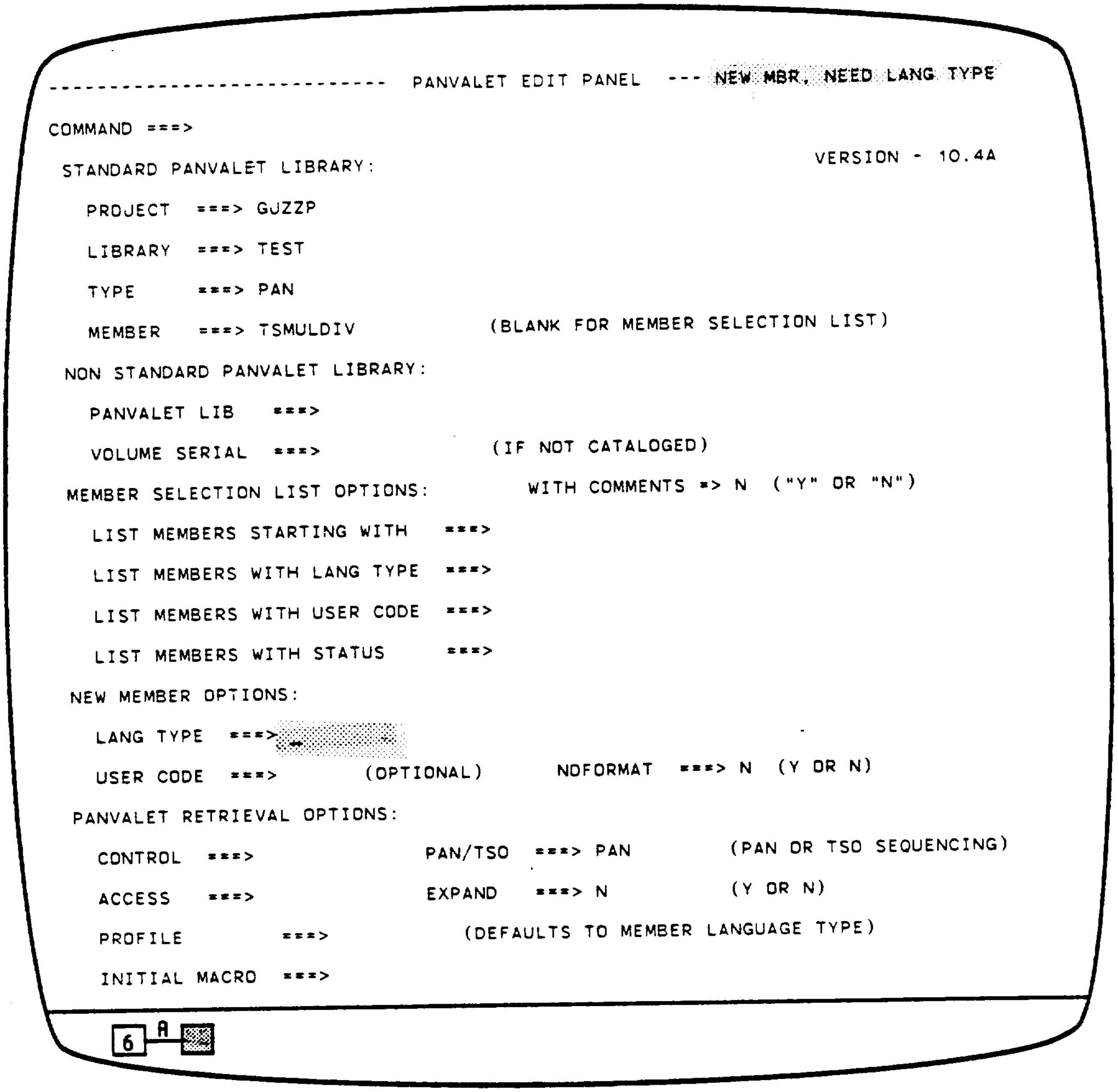
Scenario Step 3: Creating/Editing Panvalet member



```
---------------------------- PANVALET EDIT PANEL  ------------ TSADDSUB SAVED
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                   VERSION - 10.4A
   PROJECT   ===> GJZZP
   LIBRARY   ===> TEST
   TYPE      ===> PAN
   MEMBER    ===> tsmuldiv_          (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB   ===>
   VOLUME SERIAL  ===>               (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:         WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH   ===>
   LIST MEMBERS WITH LANG TYPE  ===>
   LIST MEMBERS WITH USER CODE  ===>
   LIST MEMBERS WITH STATUS     ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===>
   USER CODE  ===>       (OPTIONAL)       NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>             PAN/TSO  ===> PAN        (PAN OR TSO SEQUENCING)
   ACCESS   ===>             EXPAND   ===> N          (Y OR N)
   PROFILE        ===>           (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO  ===>
```

Type in the prefix/name combination of the last subroutine, tsmuldiv, press <enter>, and . . .

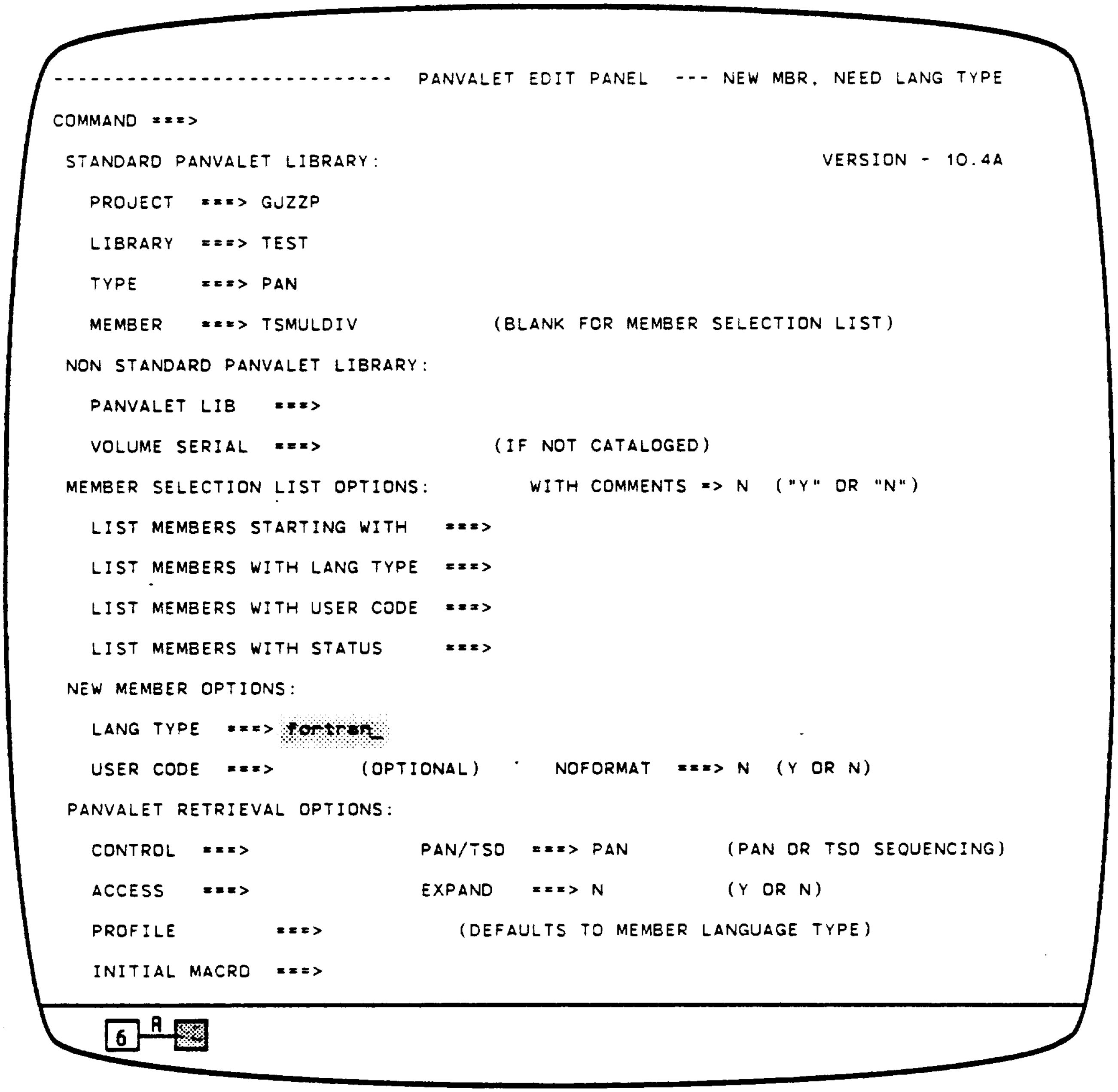
Scenario Step 3: Creating/Editing Panvalet member



```
---------------------------  PANVALET EDIT PANEL  --- NEW MBR, NEED LANG TYPE
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                    VERSION - 10.4A
   PROJECT  ===> GJZZP
   LIBRARY  ===> TEST
   TYPE     ===> PAN
   MEMBER   ===> TSMULDIV          (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB   ===>
   VOLUME SERIAL  ===>             (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:       WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH     ===>
   LIST MEMBERS WITH LANG TYPE    ===>
   LIST MEMBERS WITH USER CODE    ===>
   LIST MEMBERS WITH STATUS       ===>
 NEW MEMBER OPTIONS:
   LANG TYPE  ===>
   USER CODE  ===>        (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL  ===>               PAN/TSO  ===> PAN      (PAN OR TSO SEQUENCING)
   ACCESS   ===>               EXPAND   ===> N        (Y OR N)
   PROFILE       ===>            (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO ===>
```

. . . OOPS! the console beeps and we have a message in the short message area. It's a new member and we forgot to specify the type. Notice the cursor is automatically positioned at the data entry field that is required.

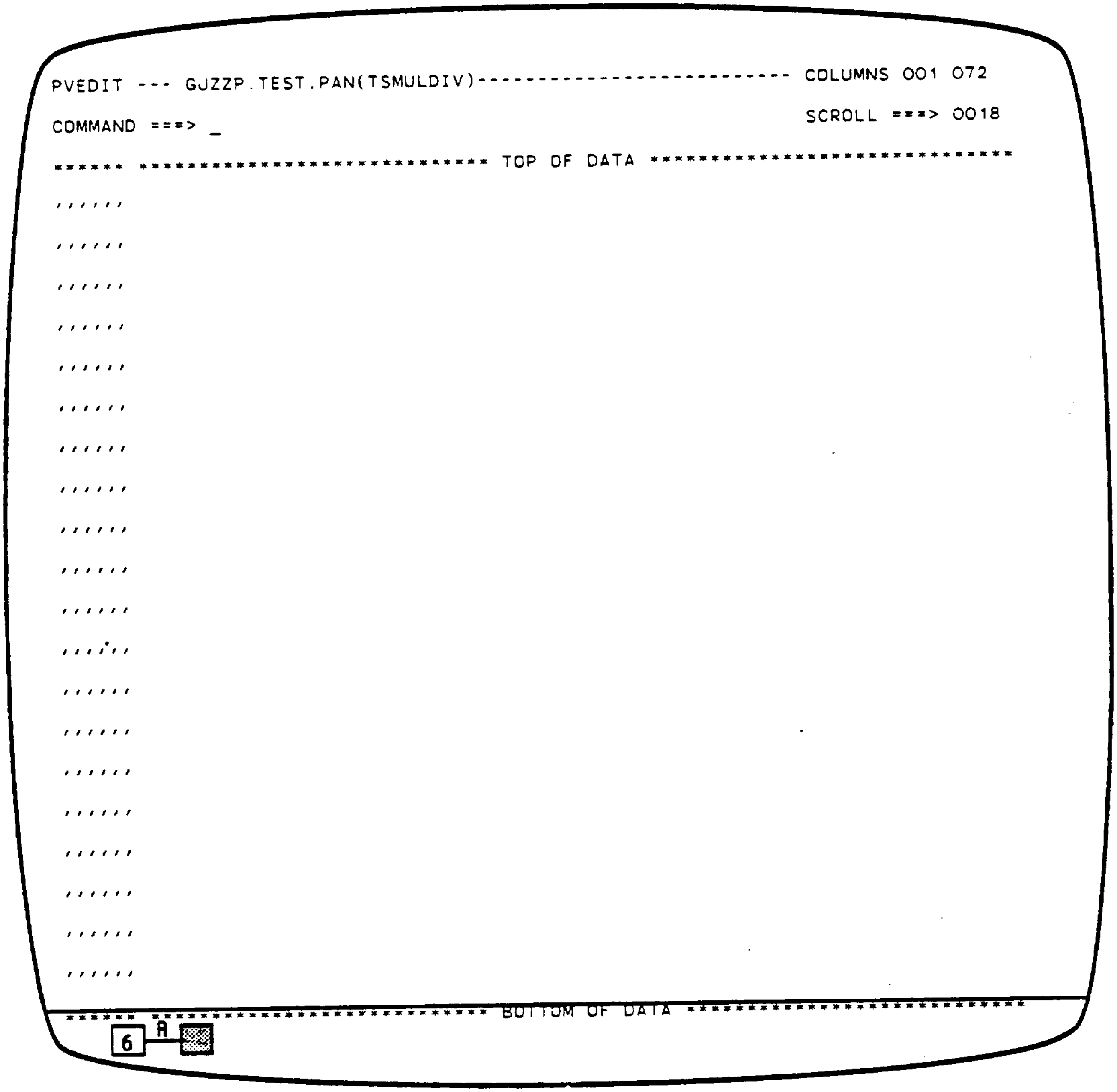
Scenario Step 3: Creating/Editing Panvalet member



```
---------------------------  PANVALET EDIT PANEL  --- NEW MBR, NEED LANG TYPE
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                        VERSION - 10.4A
   PROJECT   ===> GJZZP
   LIBRARY   ===> TEST
   TYPE      ===> PAN
   MEMBER    ===> TSMULDIV           (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB    ===>
   VOLUME SERIAL   ===>              (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:          WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH    ===>
   LIST MEMBERS WITH LANG TYPE   ===>
   LIST MEMBERS WITH USER CODE   ===>
   LIST MEMBERS WITH STATUS      ===>
 NEW MEMBER OPTIONS:
   LANG TYPE   ===> fortran_
   USER CODE   ===>        (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL   ===>              PAN/TSO  ===> PAN        (PAN OR TSO SEQUENCING)
   ACCESS    ===>              EXPAND   ===> N          (Y OR N)
   PROFILE         ===>           (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO   ===>
```

Type in fortran; press <enter> again . . .

Scenario Step 3: Creating/Editing Panvalet member

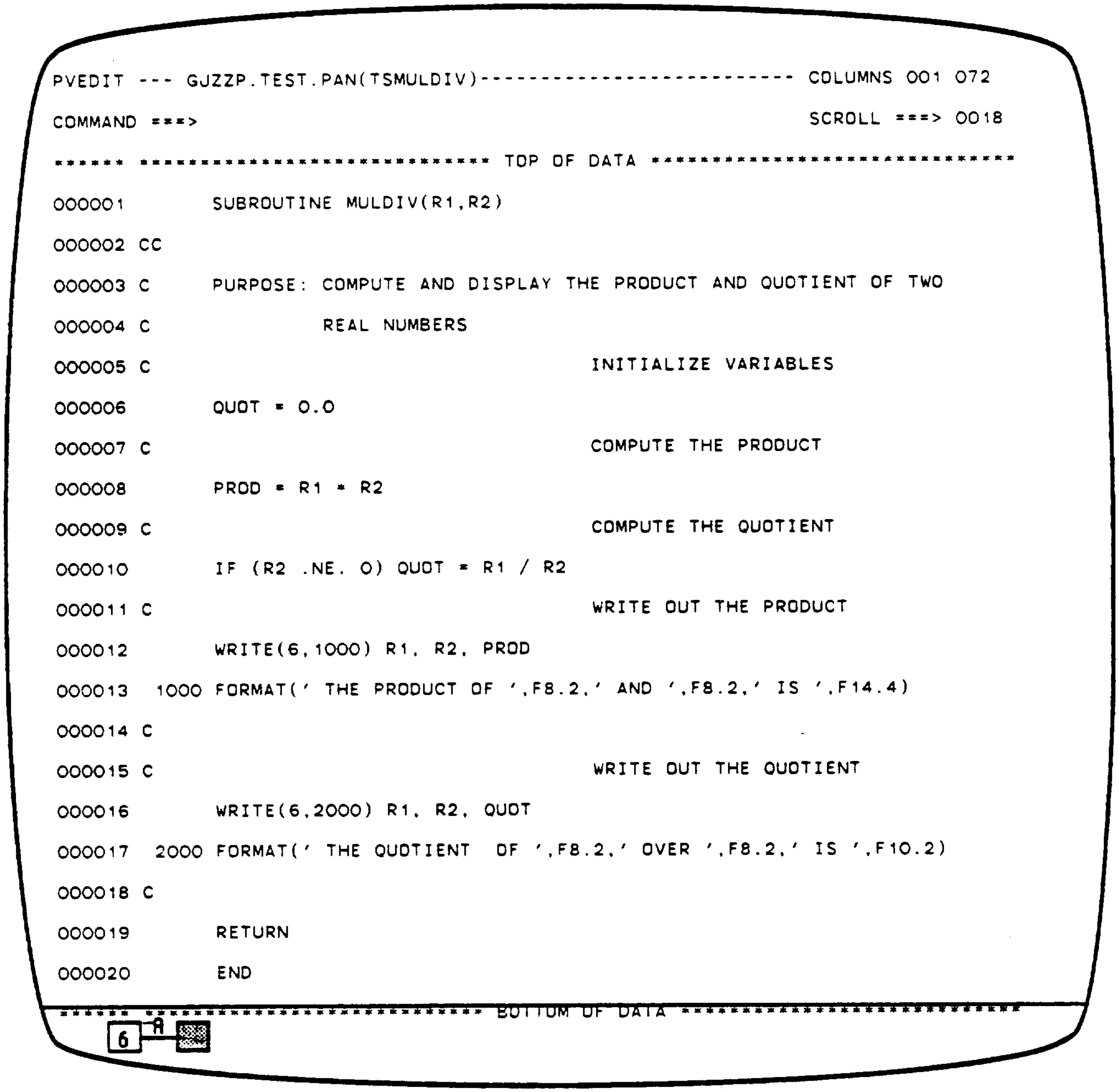


```
PVEDIT --- GJZZP.TEST.PAN(TSMULDIV)------------------------ COLUMNS 001 072
COMMAND ===> _                                               SCROLL ===> 0018
****** ***************************** TOP OF DATA ******************************
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** **************************** BOTTOM OF DATA ****************************
   6 A ▨
```

. . . and we are at the Panvalet edit data entry panel. By now you should have no problem typing in the last subroutine.

Remember that help is available by typing 'help' in the 'COMMAND===>' field or by pressing <help>. The edit help panels explain all of the bells and whistles available to the edit user.

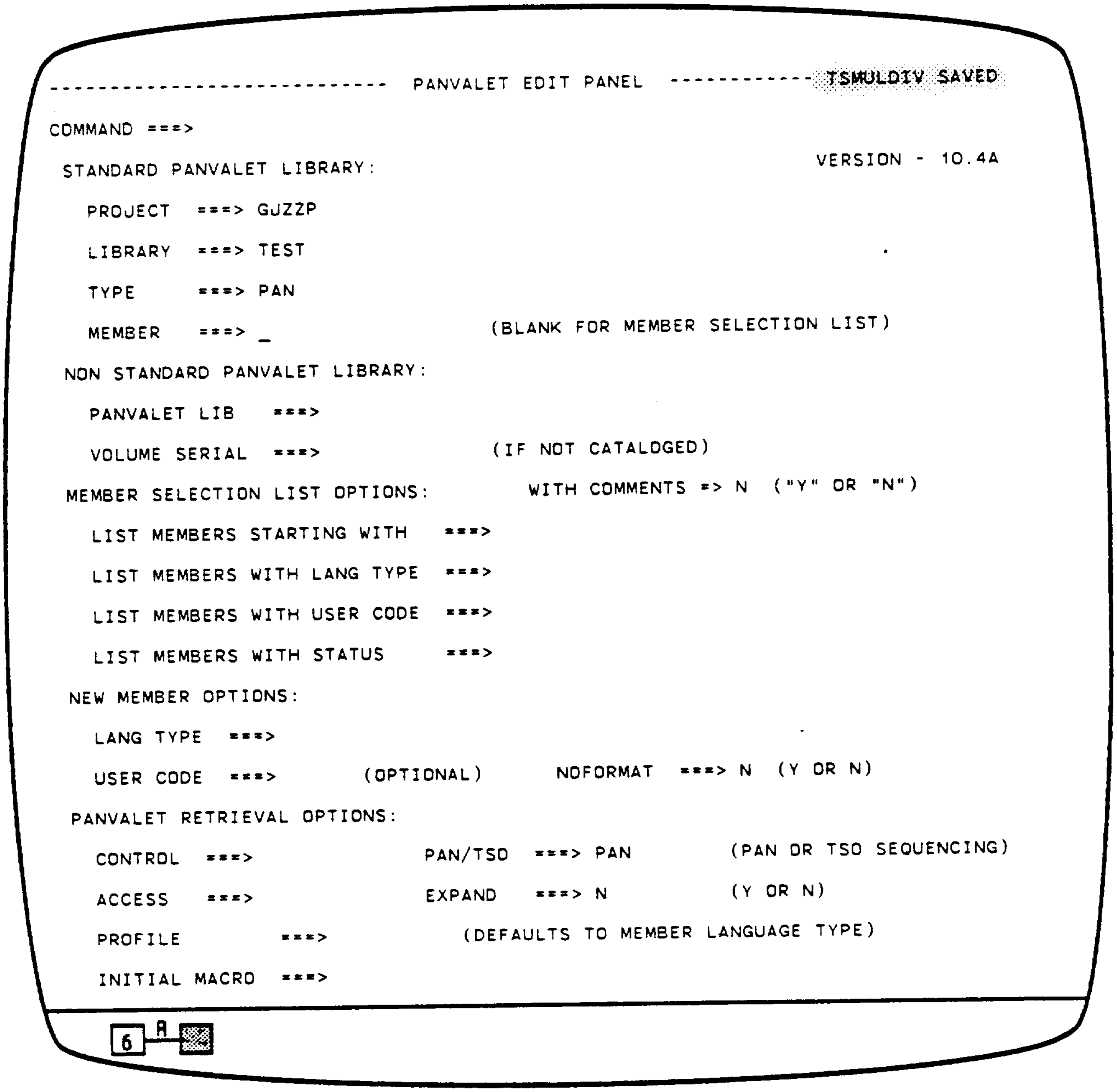
Scenario Step 3: Creating/Editing Panvalet member



```
PVEDIT --- GJZZP.TEST.PAN(TSMULDIV)------------------------- COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> 0018
****** ***************************** TOP OF DATA *****************************
      SUBROUTINE MULDIV(R1,R2)
CC
C     PURPOSE: COMPUTE AND DISPLAY THE PRODUCT AND QUOTIENT OF TWO
C              REAL NUMBERS
C                                    INITIALIZE VARIABLES
      QUOT = 0.0
C                                    COMPUTE THE PRODUCT
      PROD = R1 * R2
C                                    COMPUTE THE QUOTIENT
      IF (R2 .NE. 0) QUOT = R1 / R2
C                                    WRITE OUT THE PRODUCT
      WRITE(6,1000) R1, R2, PROD
 1000 FORMAT(' THE PRODUCT OF ',F8.2,' AND ',F8.2,' IS ',F14.4)
C
C                                    WRITE OUT THE QUOTIENT
      WRITE(6,2000) R1, R2, QUOT
 2000 FORMAT(' THE QUOTIENT  OF ',F8.2,' OVER ',F8.2,' IS ',F10.2)
C
      RETURN
      END
****** **************************** BOTTOM OF DATA ****************************
```

Once again, this is what the display looks like when it is typed in and the <enter> key is pressed. Press <end> to return to the edit parameter entry panel . . .

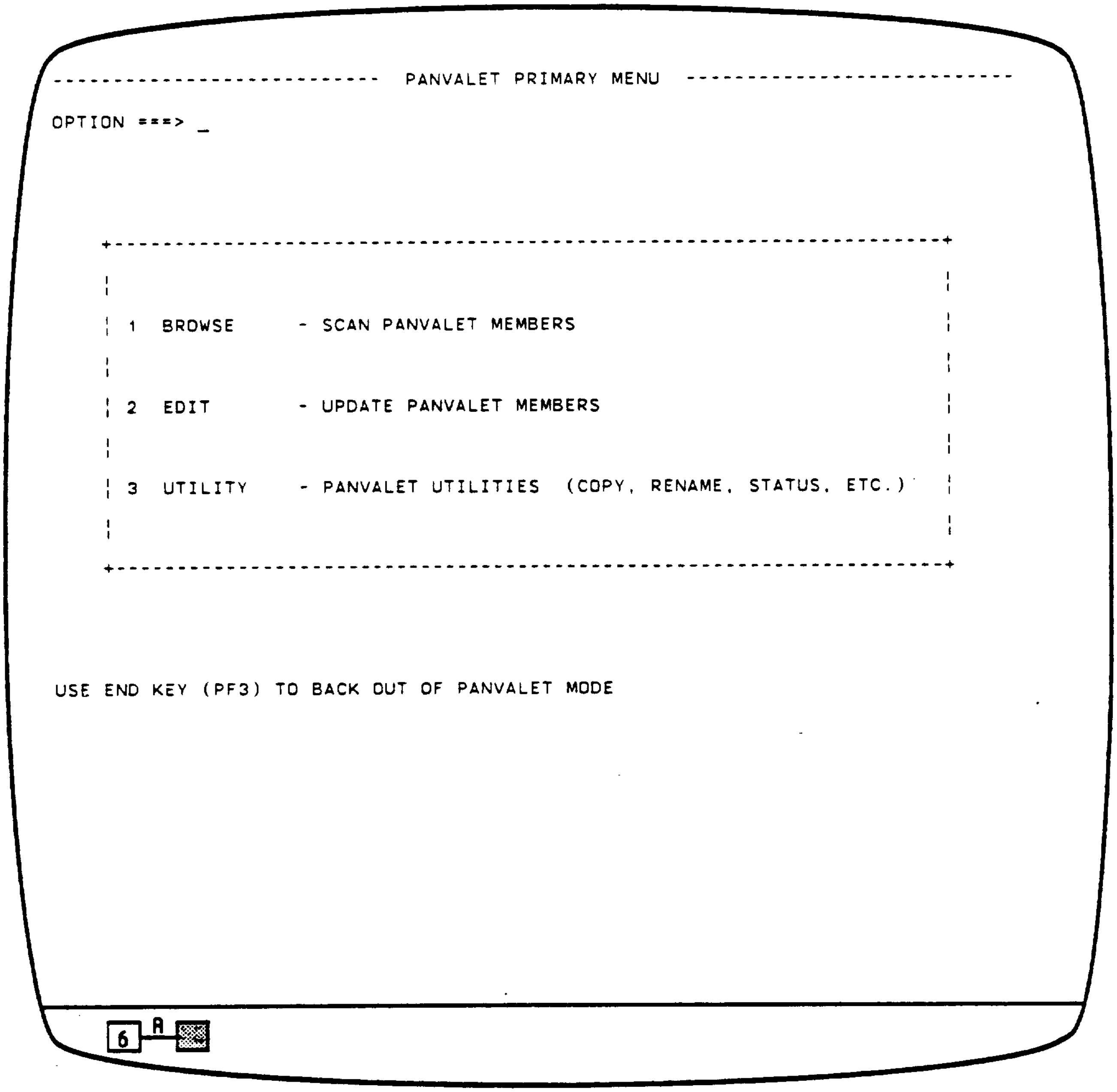
Scenario Step 3: Creating/Editing Panvalet member



```
---------------------------- PANVALET EDIT PANEL ------------TSMULDIV SAVED
COMMAND ===>
 STANDARD PANVALET LIBRARY:                                   VERSION - 10.4A
   PROJECT   ===> GJZZP
   LIBRARY   ===> TEST
   TYPE      ===> PAN
   MEMBER    ===> _                 (BLANK FOR MEMBER SELECTION LIST)
 NON STANDARD PANVALET LIBRARY:
   PANVALET LIB    ===>
   VOLUME SERIAL   ===>             (IF NOT CATALOGED)
 MEMBER SELECTION LIST OPTIONS:        WITH COMMENTS => N  ("Y" OR "N")
   LIST MEMBERS STARTING WITH       ===>
   LIST MEMBERS WITH LANG TYPE      ===>
   LIST MEMBERS WITH USER CODE      ===>
   LIST MEMBERS WITH STATUS         ===>
 NEW MEMBER OPTIONS:
   LANG TYPE   ===>
   USER CODE   ===>       (OPTIONAL)      NOFORMAT  ===> N  (Y OR N)
 PANVALET RETRIEVAL OPTIONS:
   CONTROL   ===>               PAN/TSO   ===> PAN        (PAN OR TSO SEQUENCING)
   ACCESS    ===>               EXPAND    ===> N          (Y OR N)
   PROFILE         ===>           (DEFAULTS TO MEMBER LANGUAGE TYPE)
   INITIAL MACRO   ===>
```

Note the short message 'TSMULDIV SAVED' and press <end> to go up one level . . .

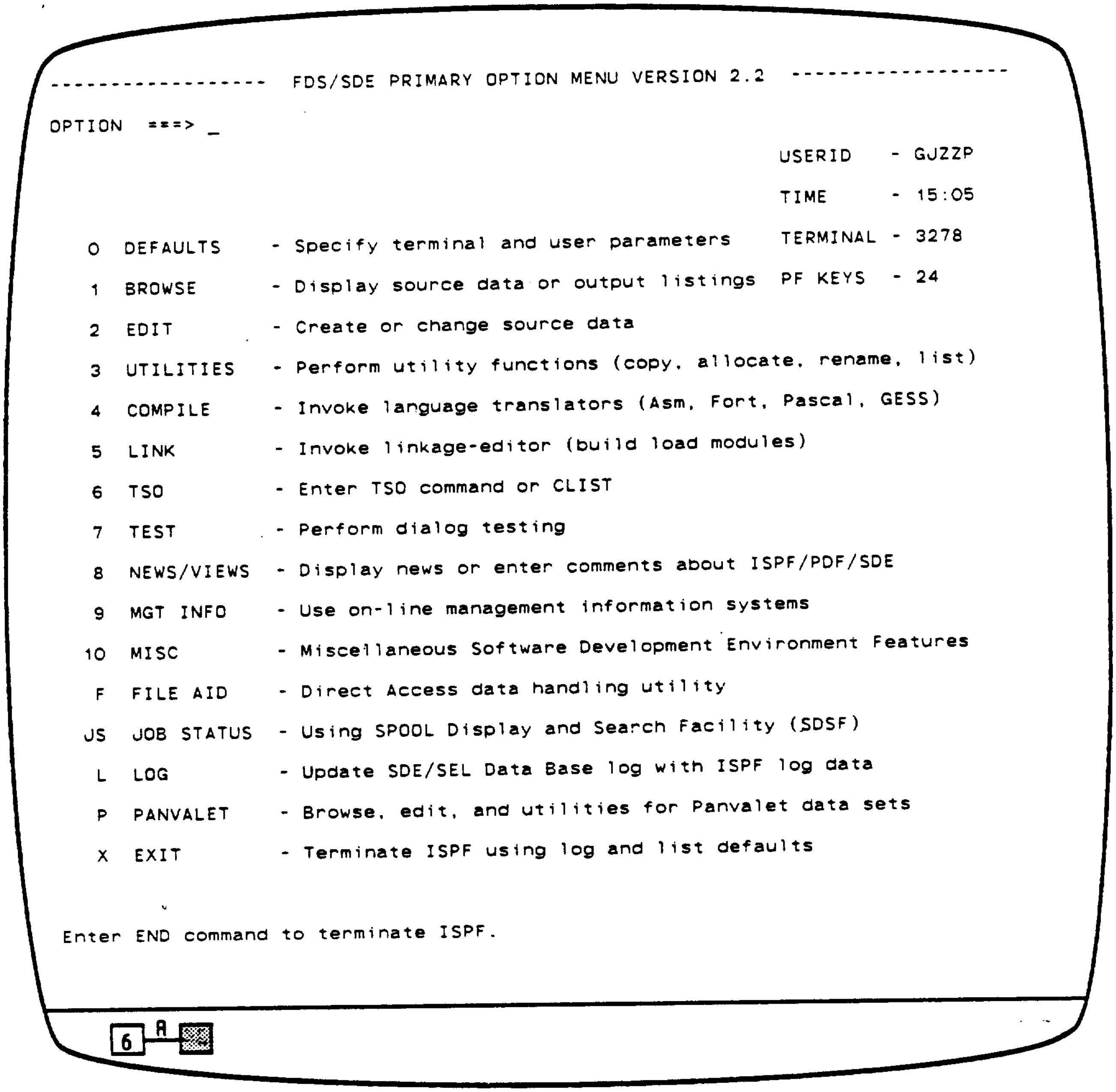
Scenario Step 3: Creating/Editing Panvalet member



```
--------------------------- PANVALET PRIMARY MENU ---------------------------
OPTION ===> _



    +-----------------------------------------------------------------------+
    |                                                                       |
    | 1  BROWSE     - SCAN PANVALET MEMBERS                                 |
    |                                                                       |
    | 2  EDIT       - UPDATE PANVALET MEMBERS                               |
    |                                                                       |
    | 3  UTILITY    - PANVALET UTILITIES  (COPY, RENAME, STATUS, ETC.)      |
    |                                                                       |
    +-----------------------------------------------------------------------+



USE END KEY (PF3) TO BACK OUT OF PANVALET MODE
```

. . . to the Panvalet selection panel. Press <end> to go up one more level . . .

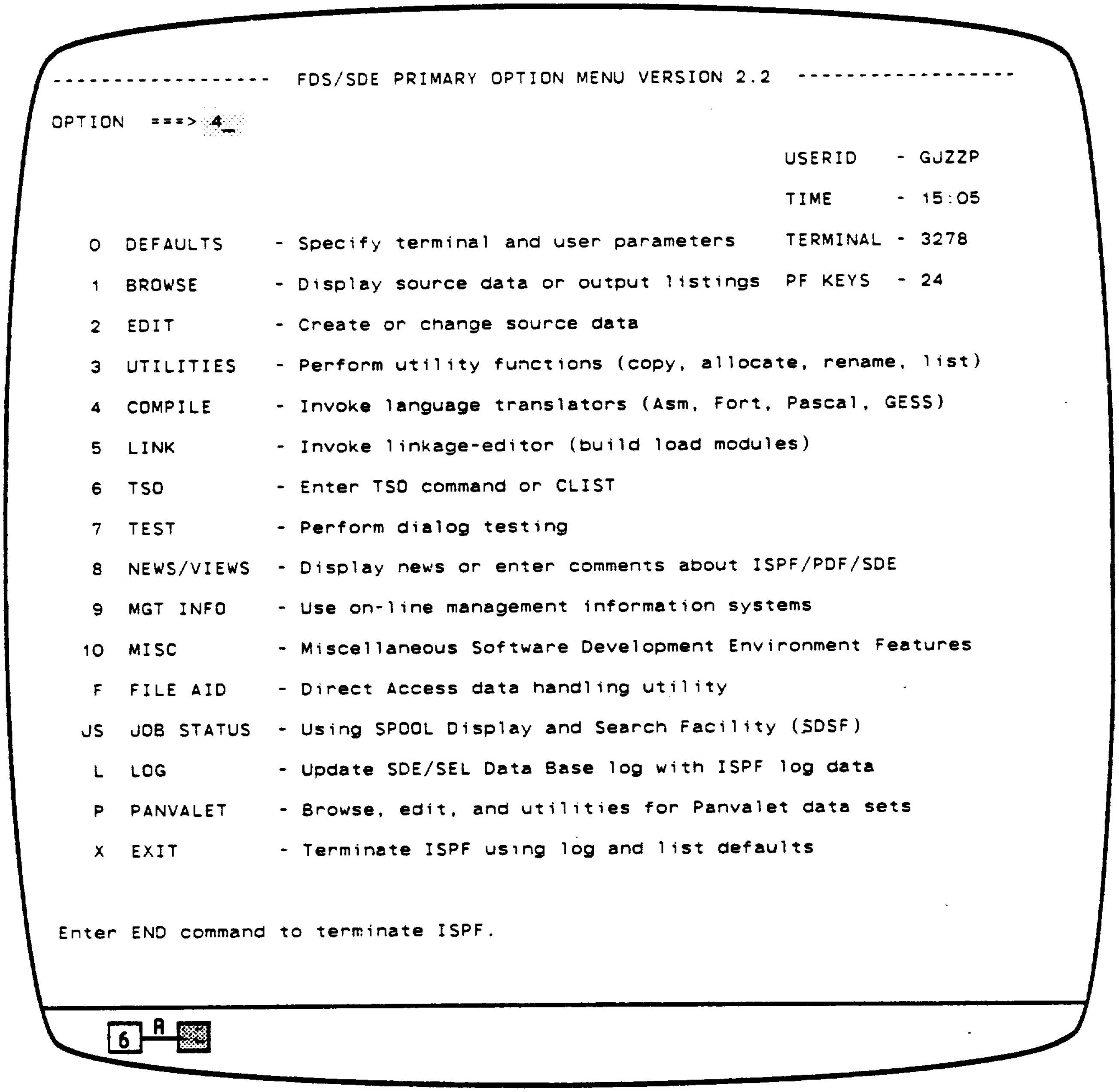
Scenario Step 3: Creating/Editing Panvalet member



```
------------------  FDS/SDE PRIMARY OPTION MENU VERSION 2.2  ------------------
OPTION  ===> _
                                                       USERID   - GJZZP
                                                       TIME     - 15:05
   0  DEFAULTS    - Specify terminal and user parameters  TERMINAL - 3278
   1  BROWSE      - Display source data or output listings PF KEYS  - 24
   2  EDIT        - Create or change source data
   3  UTILITIES   - Perform utility functions (copy, allocate, rename, list)
   4  COMPILE     - Invoke language translators (Asm, Fort, Pascal, GESS)
   5  LINK        - Invoke linkage-editor (build load modules)
   6  TSO         - Enter TSO command or CLIST
   7  TEST        - Perform dialog testing
   8  NEWS/VIEWS  - Display news or enter comments about ISPF/PDF/SDE
   9  MGT INFO    - Use on-line management information systems
  10  MISC        - Miscellaneous Software Development Environment Features
   F  FILE AID    - Direct Access data handling utility
  JS  JOB STATUS  - Using SPOOL Display and Search Facility (SDSF)
   L  LOG         - Update SDE/SEL Data Base log with ISPF log data
   P  PANVALET    - Browse, edit, and utilities for Panvalet data sets
   X  EXIT        - Terminate ISPF using log and list defaults

 Enter END command to terminate ISPF.
```

. . . to the primary option panel. We have just completed step 3 of our scenario.

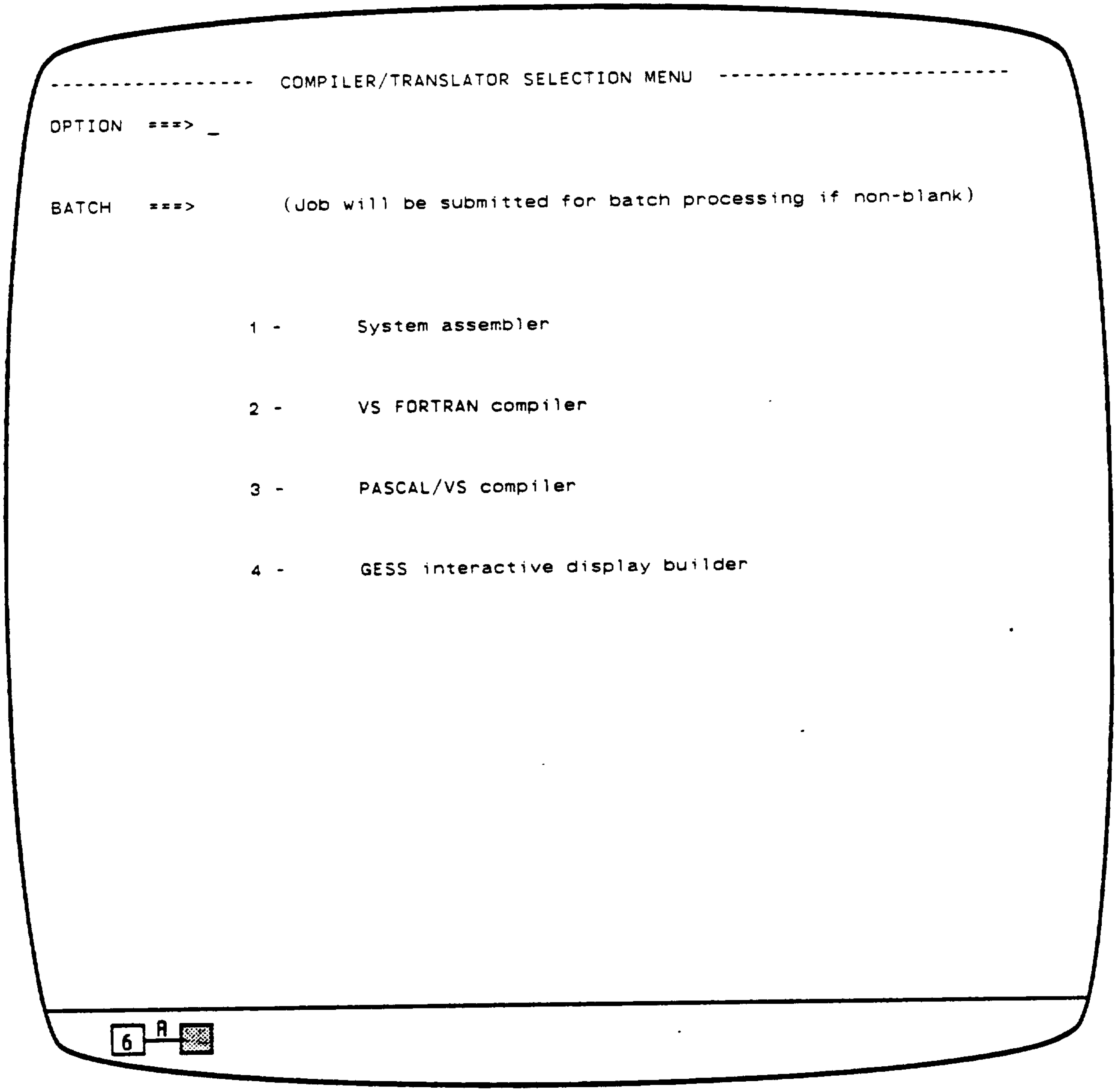
Scenario Step 4: Compiling program and subroutines



```
------------------ FDS/SDE PRIMARY OPTION MENU VERSION 2.2 -----------------

OPTION  ===> 4_
                                                          USERID   - GJZZP

                                                          TIME     - 15:05

   0  DEFAULTS    - Specify terminal and user parameters  TERMINAL - 3278

   1  BROWSE      - Display source data or output listings PF KEYS  - 24

   2  EDIT        - Create or change source data

   3  UTILITIES   - Perform utility functions (copy, allocate, rename, list)

   4  COMPILE     - Invoke language translators (Asm, Fort, Pascal, GESS)

   5  LINK        - Invoke linkage-editor (build load modules)

   6  TSO         - Enter TSO command or CLIST

   7  TEST        - Perform dialog testing

   8  NEWS/VIEWS  - Display news or enter comments about ISPF/PDF/SDE

   9  MGT INFO    - Use on-line management information systems

  10  MISC        - Miscellaneous Software Development Environment Features

   F  FILE AID    - Direct Access data handling utility

  JS  JOB STATUS  - Using SPOOL Display and Search Facility (SDSF)

   L  LOG         - Update SDE/SEL Data Base log with ISPF log data

   P  PANVALET    - Browse, edit, and utilities for Panvalet data sets

   X  EXIT        - Terminate ISPF using log and list defaults


Enter END command to terminate ISPF.
```

Step 4 requires that we compile these three routines. To do this, type 4, then <enter> . . .

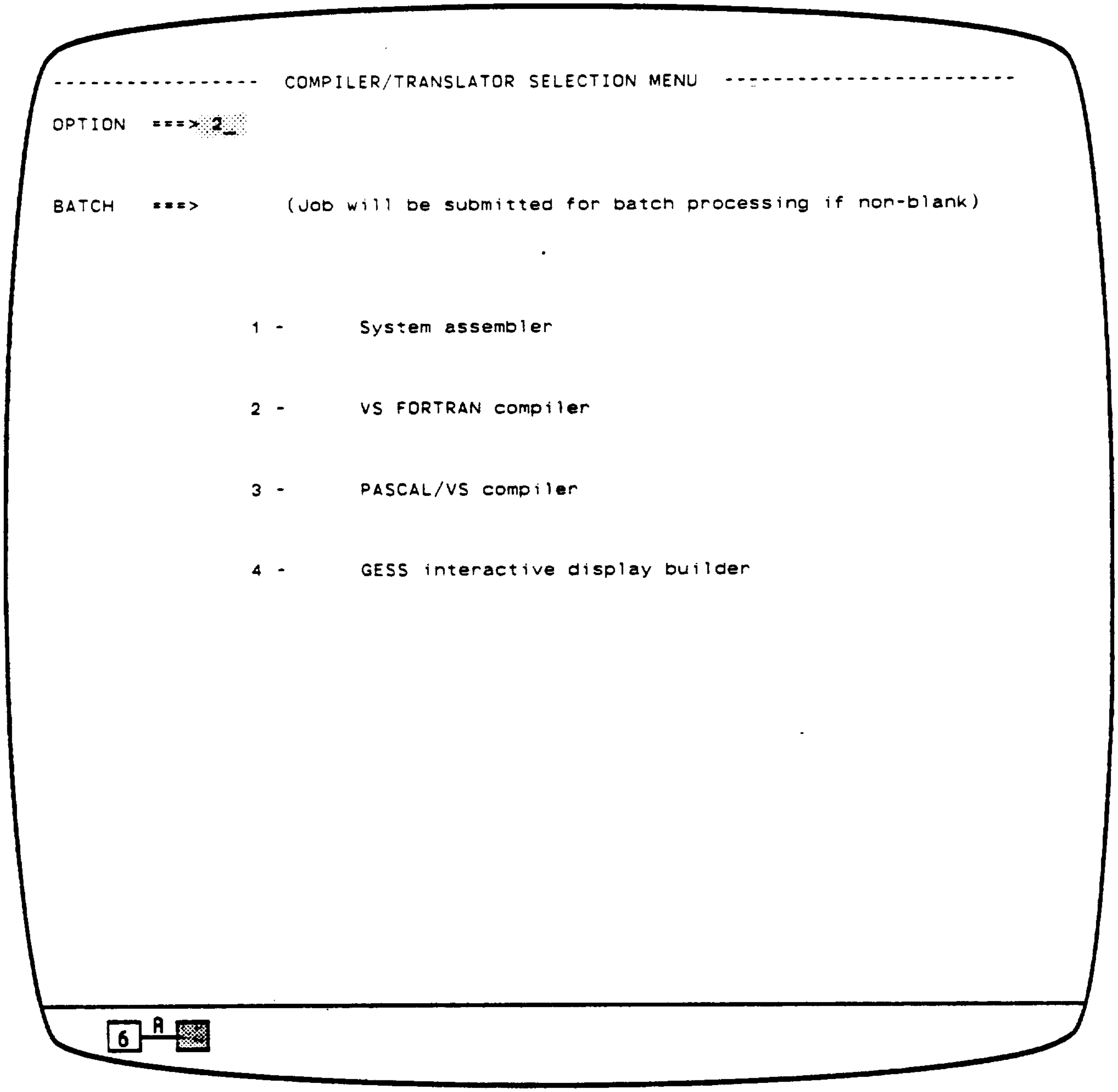
Scenario Step 4: Compiling program and subroutines



```
----------------  COMPILER/TRANSLATOR SELECTION MENU  -----------------------
OPTION  ===> _


BATCH   ===>        (Job will be submitted for batch processing if non-blank)




                 1 -      System assembler


                 2 -      VS FORTRAN compiler


                 3 -      PASCAL/VS compiler


                 4 -      GESS interactive display builder
```

. . . and we arrive at the compiler/translator function's selection panel. We wish to perform a FORTRAN compilation, so . . .

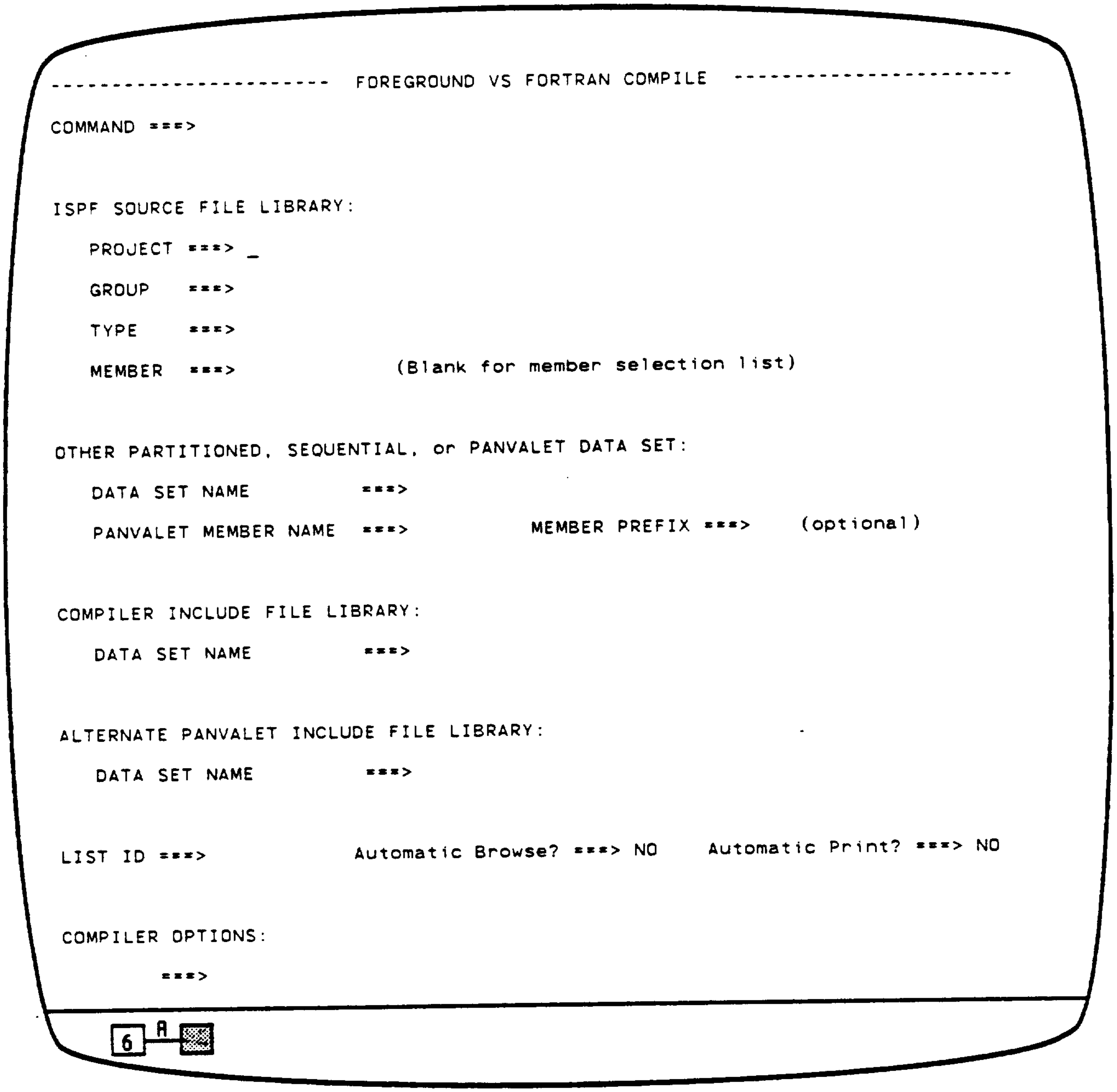
Scenario Step 4: Compiling program and subroutines



```
-----------------  COMPILER/TRANSLATOR SELECTION MENU  -----------------------
OPTION   ===> 2_

BATCH    ===>      (Job will be submitted for batch processing if non-blank)


                1 -      System assembler

                2 -      VS FORTRAN compiler

                3 -      PASCAL/VS compiler

                4 -      GESS interactive display builder
```

. . . type 2, then press <enter>, and . . .

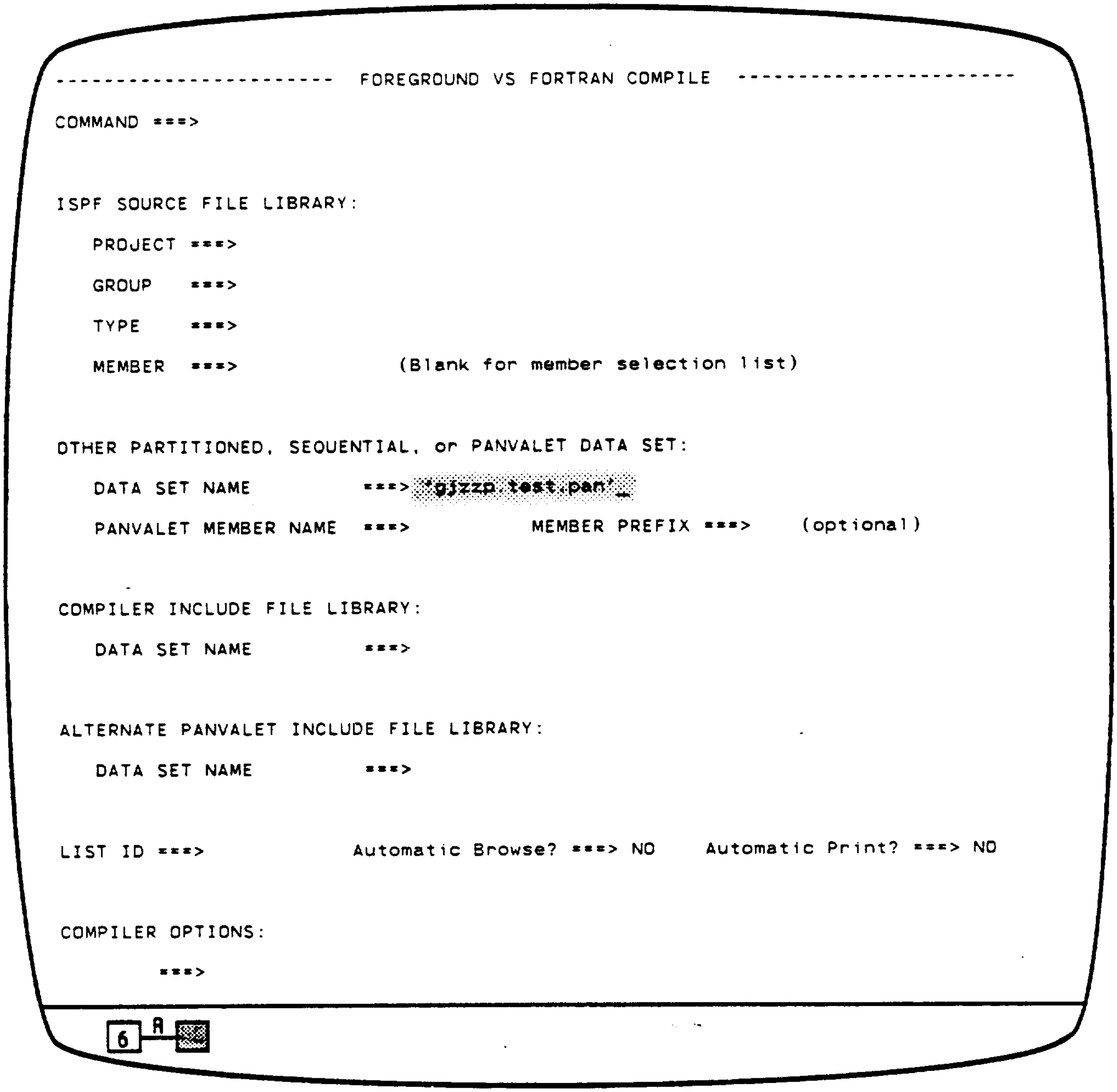
Scenario Step 4: Compiling program and subroutines



```
----------------------- FOREGROUND VS FORTRAN COMPILE -----------------------
COMMAND ===>


ISPF SOURCE FILE LIBRARY:
   PROJECT ===> _
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>                (Blank for member selection list)


OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
   DATA SET NAME         ===>
   PANVALET MEMBER NAME  ===>          MEMBER PREFIX ===>    (optional)


COMPILER INCLUDE FILE LIBRARY:
   DATA SET NAME         ===>


ALTERNATE PANVALET INCLUDE FILE LIBRARY:
   DATA SET NAME         ===>


LIST ID ===>            Automatic Browse? ===> NO    Automatic Print? ===> NO


COMPILER OPTIONS:
      ===>
```

. . . the foreground VS FORTRAN compile parameter entry panel is displayed. <NL> four times to the DATA SET NAME parameter entry field . . .

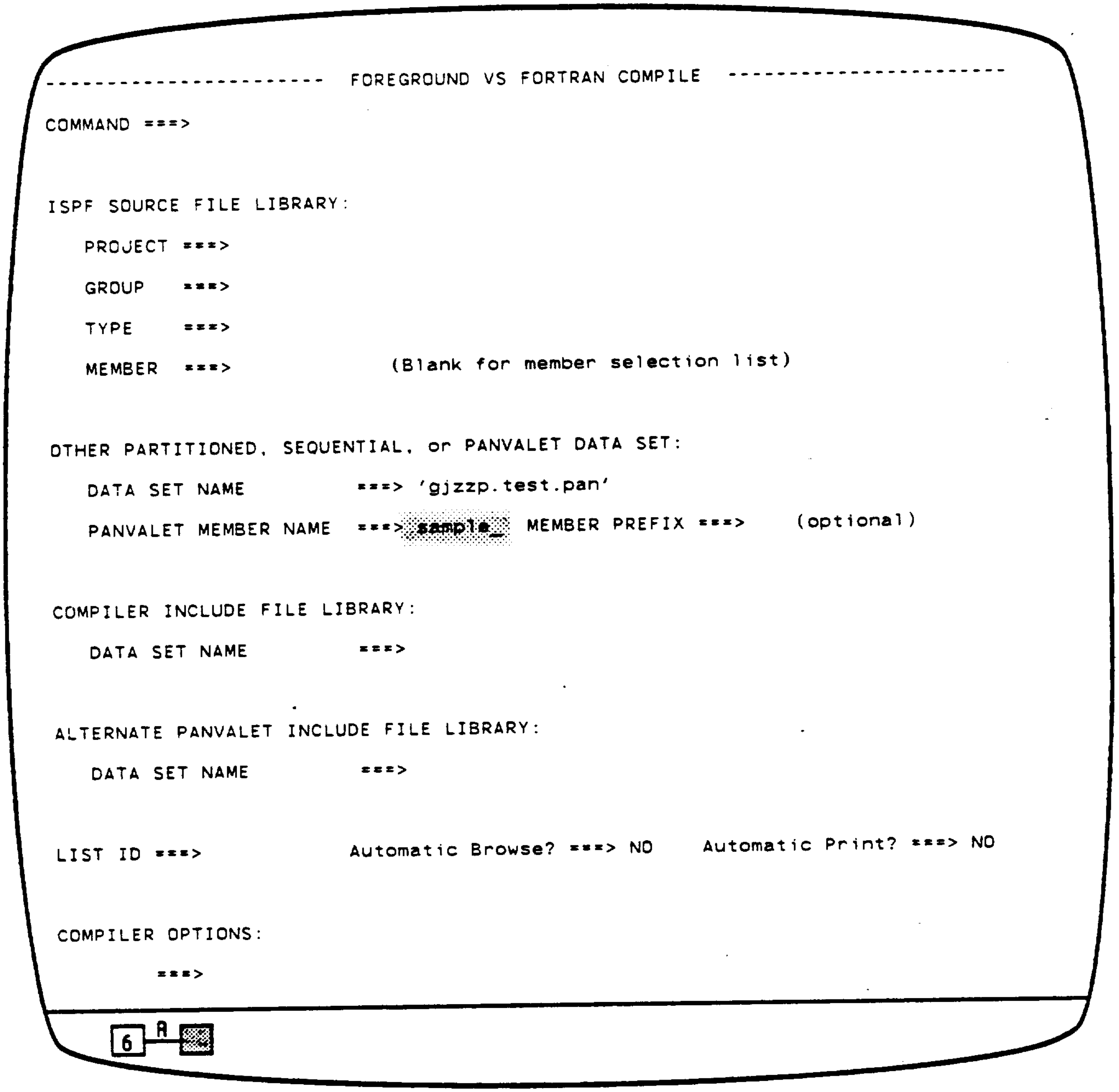
Scenario Step 4: Compiling program and subroutines



```
----------------------- FOREGROUND VS FORTRAN COMPILE -----------------------
COMMAND ===>


ISPF SOURCE FILE LIBRARY:
   PROJECT ===>
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>             (Blank for member selection list)


OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
   DATA SET NAME          ===> 'gjzzp.test.pan'_
   PANVALET MEMBER NAME   ===>          MEMBER PREFIX ===>    (optional)


COMPILER INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>


ALTERNATE PANVALET INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>


LIST ID ===>            Automatic Browse? ===> NO    Automatic Print? ===> NO


COMPILER OPTIONS:
        ===>
```

. . . and type in the name of the Panvalet library (even though our TSO userid is GJZZP, we are explicitly defining our data set by using quotes). <FT> once to the member name field . . .

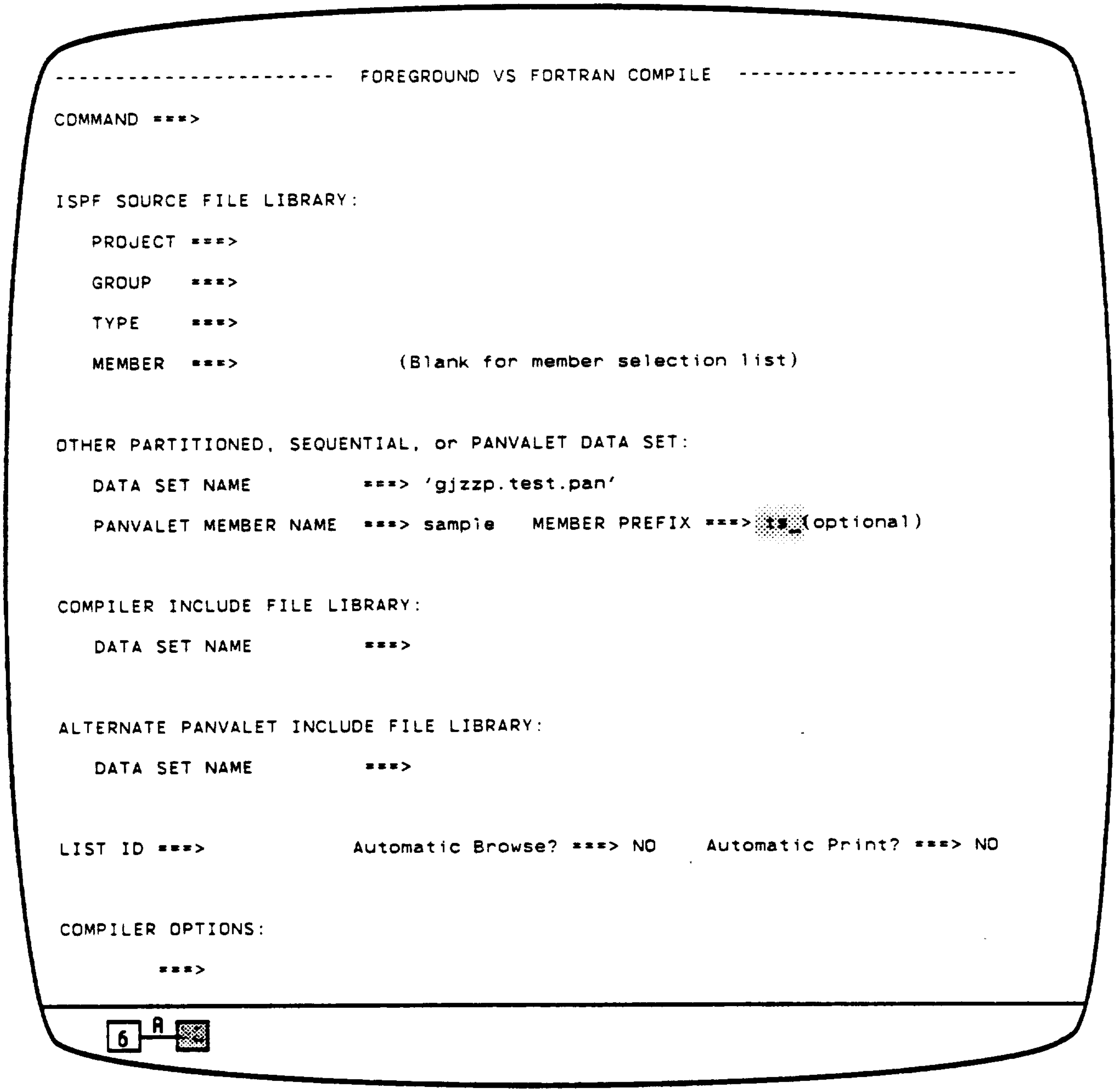
Scenario Step 4: Compiling program and subroutines



```
----------------------- FOREGROUND VS FORTRAN COMPILE -----------------------
COMMAND ===>


ISPF SOURCE FILE LIBRARY:
   PROJECT ===>
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>               (Blank for member selection list)


OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
   DATA SET NAME          ===> 'gjzzp.test.pan'
   PANVALET MEMBER NAME   ===> sample_ MEMBER PREFIX ===>    (optional)


COMPILER INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>


ALTERNATE PANVALET INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>


LIST ID ===>             Automatic Browse? ===> NO    Automatic Print? ===> NO


COMPILER OPTIONS:
       ===>
```

. . . and type in the Panvalet member name, SAMPLE. <FT> once more to the member prefix field . . .

Scenario Step 4: Compiling program and subroutines

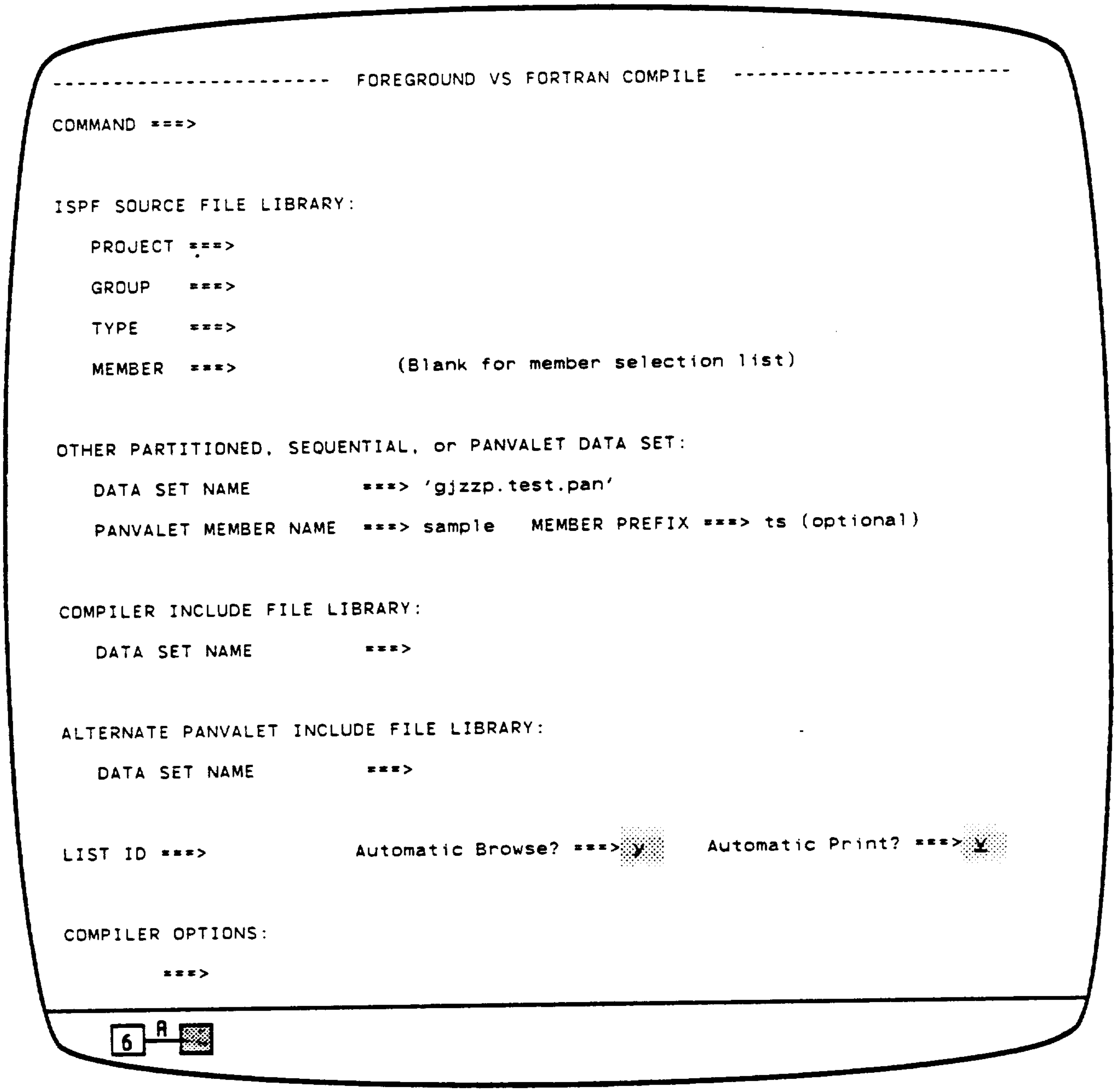


```
----------------------  FOREGROUND VS FORTRAN COMPILE  ----------------------
COMMAND ===>


ISPF SOURCE FILE LIBRARY:
   PROJECT ===>
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>             (Blank for member selection list)


OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
   DATA SET NAME         ===> 'gjzzp.test.pan'
   PANVALET MEMBER NAME  ===> sample   MEMBER PREFIX ===> ts_(optional)


COMPILER INCLUDE FILE LIBRARY:
   DATA SET NAME         ===>


ALTERNATE PANVALET INCLUDE FILE LIBRARY:
   DATA SET NAME         ===>


LIST ID ===>            Automatic Browse? ===> NO    Automatic Print? ===> NO


COMPILER OPTIONS:
        ===>
```

. . . and type in the Panvalet member prefix, TS. The INCLUDE FILES are not required, <help> will explain them if you wish to know about them. The listing ID will default to the prefix and member name.

Note that the automatic browse and automatic print answers are both 'NO'. We should change this for a first compile. <FT> to these fields . . .

Scenario Step 4: Compiling program and subroutines

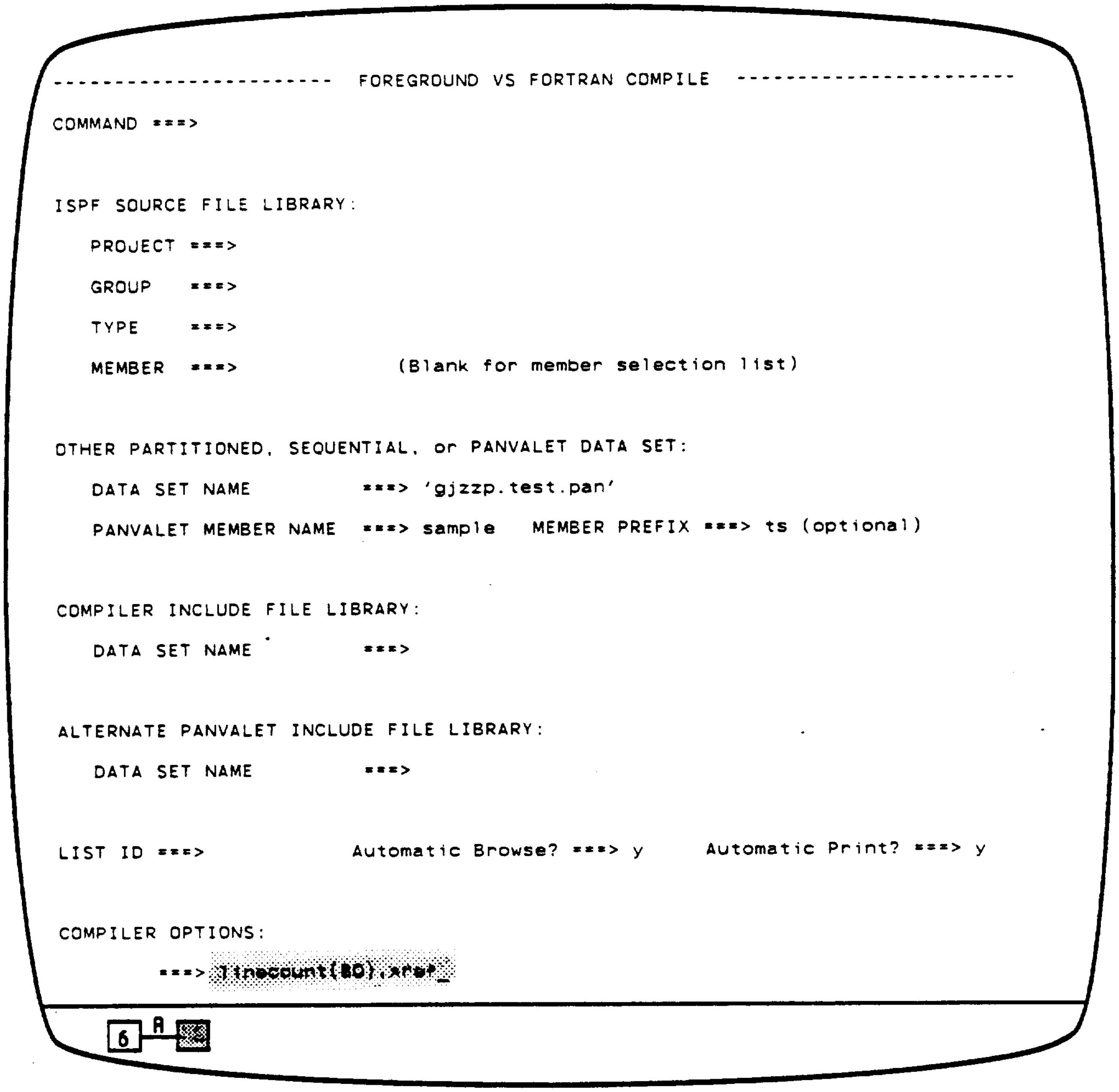


```
----------------------  FOREGROUND VS FORTRAN COMPILE  ----------------------
COMMAND ===>


ISPF SOURCE FILE LIBRARY:
   PROJECT ===>
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>              (Blank for member selection list)


OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
   DATA SET NAME         ===> 'gjzzp.test.pan'
   PANVALET MEMBER NAME  ===> sample   MEMBER PREFIX ===> ts (optional)


COMPILER INCLUDE FILE LIBRARY:
   DATA SET NAME         ===>


ALTERNATE PANVALET INCLUDE FILE LIBRARY:
   DATA SET NAME         ===>


LIST ID ===>            Automatic Browse? ===> y    Automatic Print? ===> Y


COMPILER OPTIONS:
       ===>
```

. . . and change them both to 'y'. SDE will accept y, ye, or yes in most cases; blank defaults to 'NO'. The entries can be in upper or lower case.

Now <FT> to the compiler options field . . .

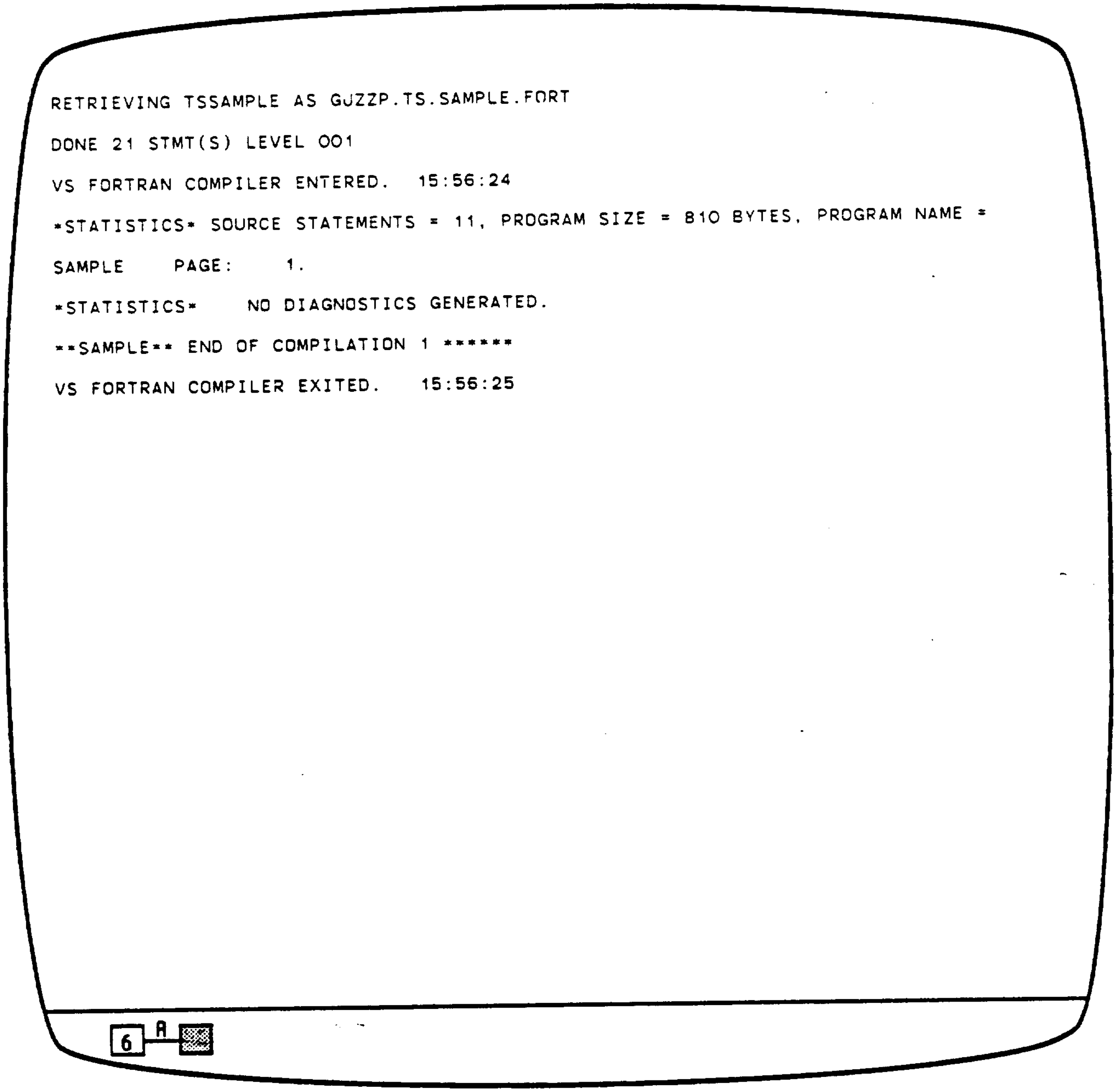
Scenario Step 4: Compiling program and subroutines



```
----------------------  FOREGROUND VS FORTRAN COMPILE  ----------------------
COMMAND ===>


ISPF SOURCE FILE LIBRARY:

   PROJECT ===>

   GROUP   ===>

   TYPE    ===>

   MEMBER  ===>              (Blank for member selection list)


OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:

   DATA SET NAME         ===> 'gjzzp.test.pan'

   PANVALET MEMBER NAME  ===> sample   MEMBER PREFIX ===> ts (optional)


COMPILER INCLUDE FILE LIBRARY:

   DATA SET NAME         ===>


ALTERNATE PANVALET INCLUDE FILE LIBRARY:

   DATA SET NAME         ===>


LIST ID ===>            Automatic Browse? ===> y    Automatic Print? ===> y


COMPILER OPTIONS:

        ===> linecount(80),xref_
```

. . . and type in a couple of compiler options. The object module output by the compiler will be named 'GJZZP.TS.SAMPLE.OBJ'. Press <enter> and the compile will begin. The screen will clear . . .

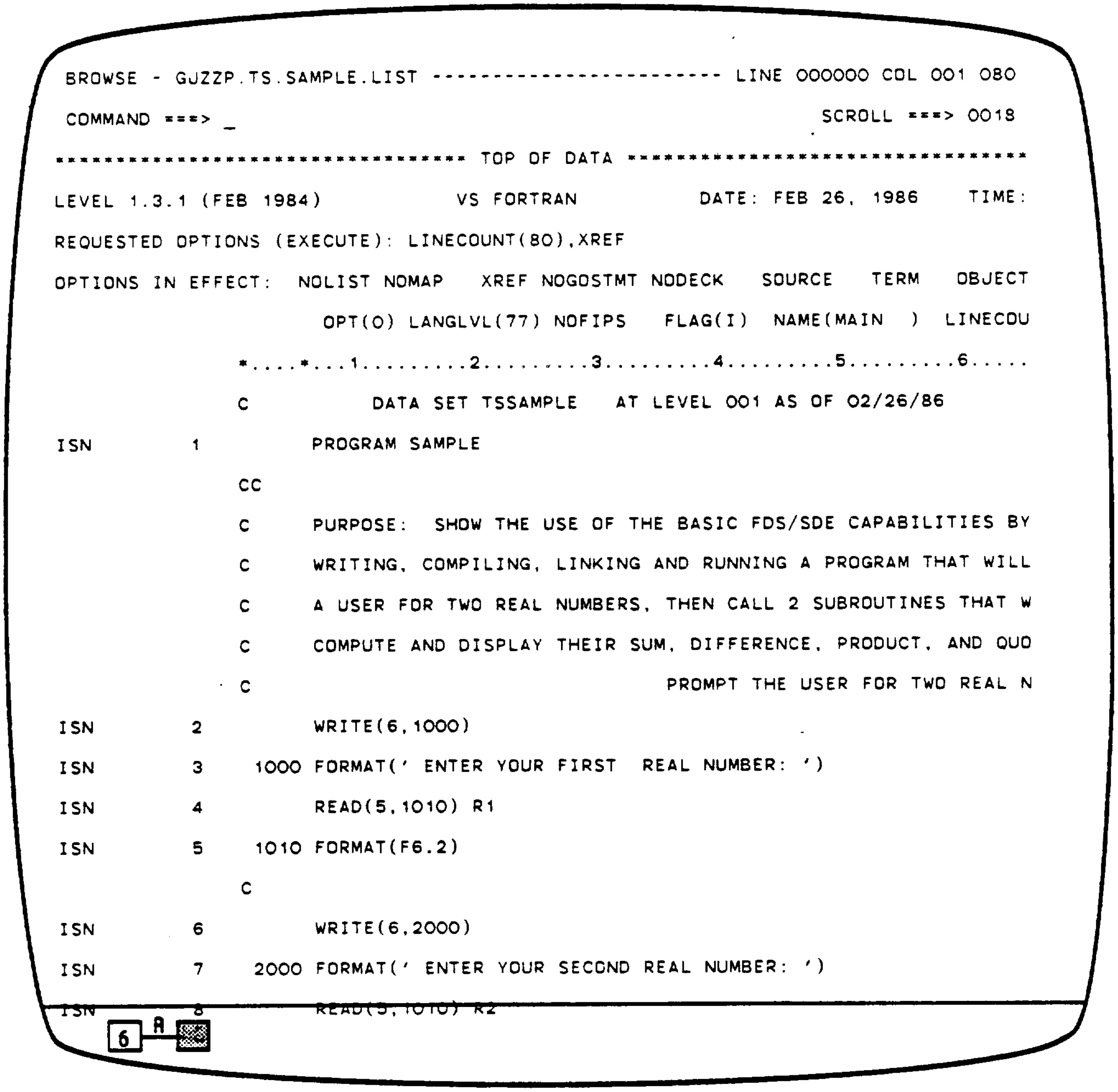
Scenario Step 4: Compiling program and subroutines



```
RETRIEVING TSSAMPLE AS GJZZP.TS.SAMPLE.FORT
DONE 21 STMT(S) LEVEL 001
VS FORTRAN COMPILER ENTERED.  15:56:24
*STATISTICS* SOURCE STATEMENTS = 11, PROGRAM SIZE = 810 BYTES, PROGRAM NAME =
SAMPLE    PAGE:     1.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
**SAMPLE** END OF COMPILATION 1 ******
VS FORTRAN COMPILER EXITED.   15:56:25
```

and these messages will be displayed as the compile progresses. A few seconds after the VS FORTRAN COMPILER EXITED message is displayed . . .

Scenario Step 4: Compiling program and subroutines

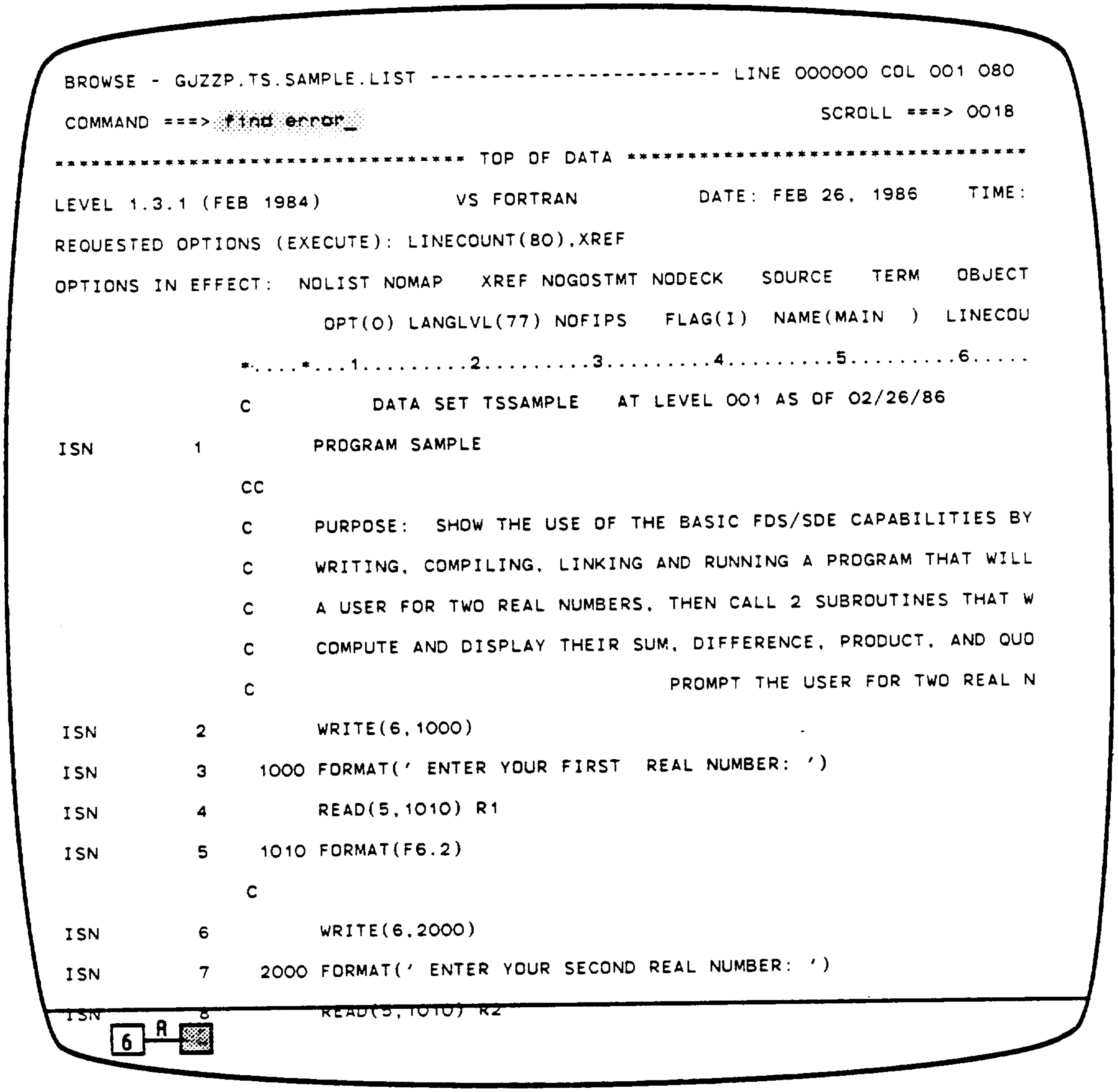


```
 BROWSE - GJZZP.TS.SAMPLE.LIST ------------------------ LINE 000000 COL 001 080
 COMMAND ===> _                                                  SCROLL ===> 0018
********************************* TOP OF DATA *********************************
LEVEL 1.3.1 (FEB 1984)           VS FORTRAN          DATE: FEB 26, 1986    TIME:
REQUESTED OPTIONS (EXECUTE): LINECOUNT(80),XREF
OPTIONS IN EFFECT:  NOLIST NOMAP   XREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOU
                *....*...1.........2.........3.........4.........5.........6.....
                C          DATA SET TSSAMPLE   AT LEVEL 001 AS OF 02/26/86
ISN        1          PROGRAM SAMPLE
                CC
                C     PURPOSE:  SHOW THE USE OF THE BASIC FDS/SDE CAPABILITIES BY
                C     WRITING, COMPILING, LINKING AND RUNNING A PROGRAM THAT WILL
                C     A USER FOR TWO REAL NUMBERS, THEN CALL 2 SUBROUTINES THAT W
                C     COMPUTE AND DISPLAY THEIR SUM, DIFFERENCE, PRODUCT, AND QUO
                C                              PROMPT THE USER FOR TWO REAL N
ISN        2          WRITE(6,1000)
ISN        3     1000 FORMAT(' ENTER YOUR FIRST  REAL NUMBER: ')
ISN        4          READ(5,1010) R1
ISN        5     1010 FORMAT(F6.2)
                C
ISN        6          WRITE(6,2000)
ISN        7     2000 FORMAT(' ENTER YOUR SECOND REAL NUMBER: ')
ISN        8          READ(5,1010) R2
```

. . . this display will be seen. It is a "browse" of the listing file. Note the listing file name has defaulted to the prefix and member as anticipated.

If you wish you can use <up>, <down>, <left>, or <right> to scroll the display window around the listing file. The listing file is 132 characters wide.

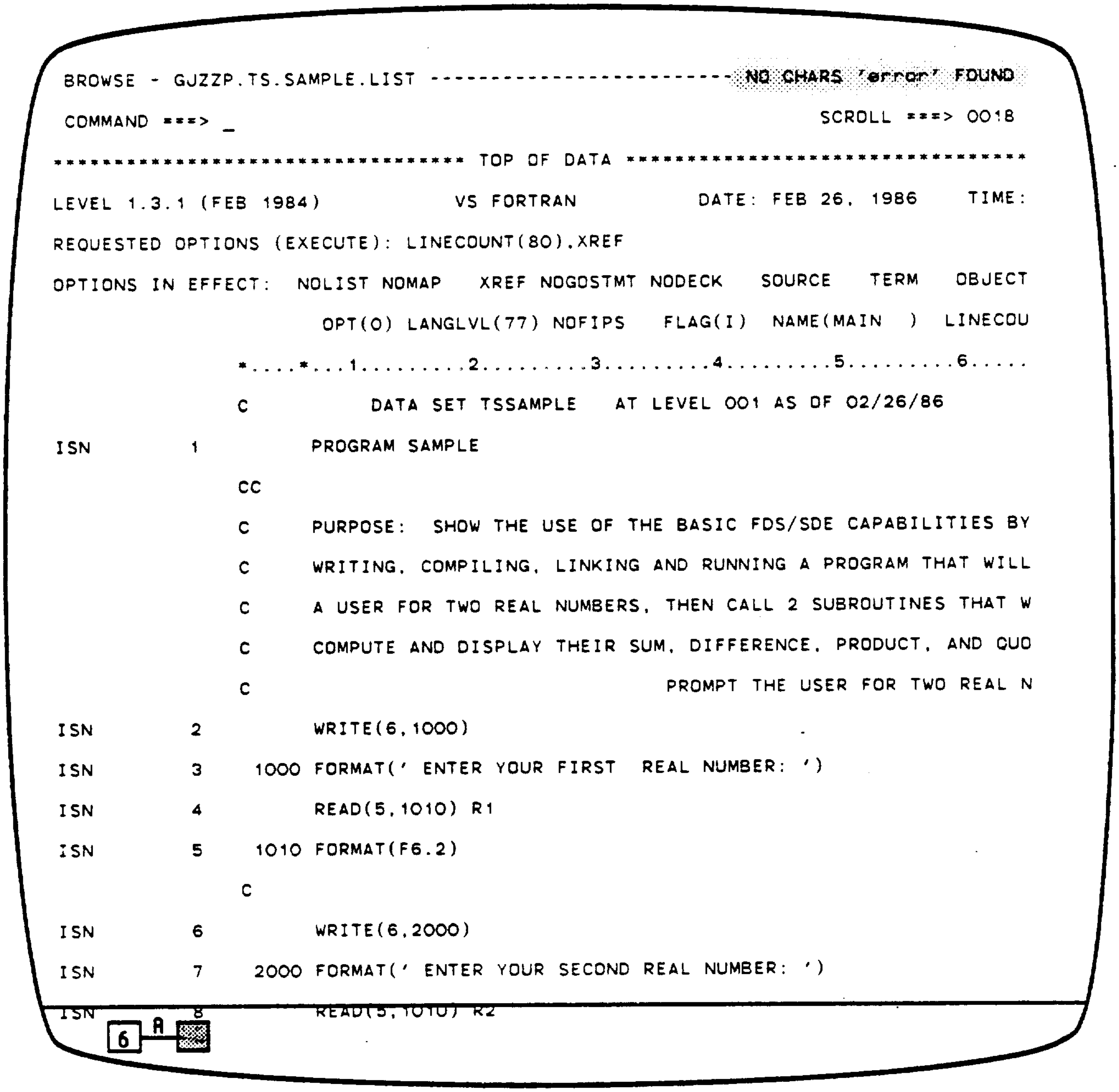
Scenario Step 4: Compiling program and subroutines



```
BROWSE - GJZZP.TS.SAMPLE.LIST ----------------------- LINE 000000 COL 001 080
COMMAND ===> find error_                                      SCROLL ===> 0018
********************************* TOP OF DATA *********************************
LEVEL 1.3.1 (FEB 1984)          VS FORTRAN          DATE: FEB 26, 1986    TIME:
REQUESTED OPTIONS (EXECUTE): LINECOUNT(80),XREF
OPTIONS IN EFFECT:  NOLIST NOMAP   XREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT
                     OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOU
                *....*...1.........2.........3.........4.........5.........6.....
                C          DATA SET TSSAMPLE   AT LEVEL 001 AS OF 02/26/86
ISN       1            PROGRAM SAMPLE
                CC
                C      PURPOSE:  SHOW THE USE OF THE BASIC FDS/SDE CAPABILITIES BY
                C      WRITING, COMPILING, LINKING AND RUNNING A PROGRAM THAT WILL
                C      A USER FOR TWO REAL NUMBERS, THEN CALL 2 SUBROUTINES THAT W
                C      COMPUTE AND DISPLAY THEIR SUM, DIFFERENCE, PRODUCT, AND QUO
                C                                 PROMPT THE USER FOR TWO REAL N
ISN       2            WRITE(6,1000)
ISN       3       1000 FORMAT(' ENTER YOUR FIRST  REAL NUMBER: ')
ISN       4            READ(5,1010) R1
ISN       5       1010 FORMAT(F6.2)
                C
ISN       6            WRITE(6,2000)
ISN       7       2000 FORMAT(' ENTER YOUR SECOND REAL NUMBER: ')
ISN       8            READ(5,1010) R2
```

Type 'find error' in the command line field of the display, press <enter> . . .

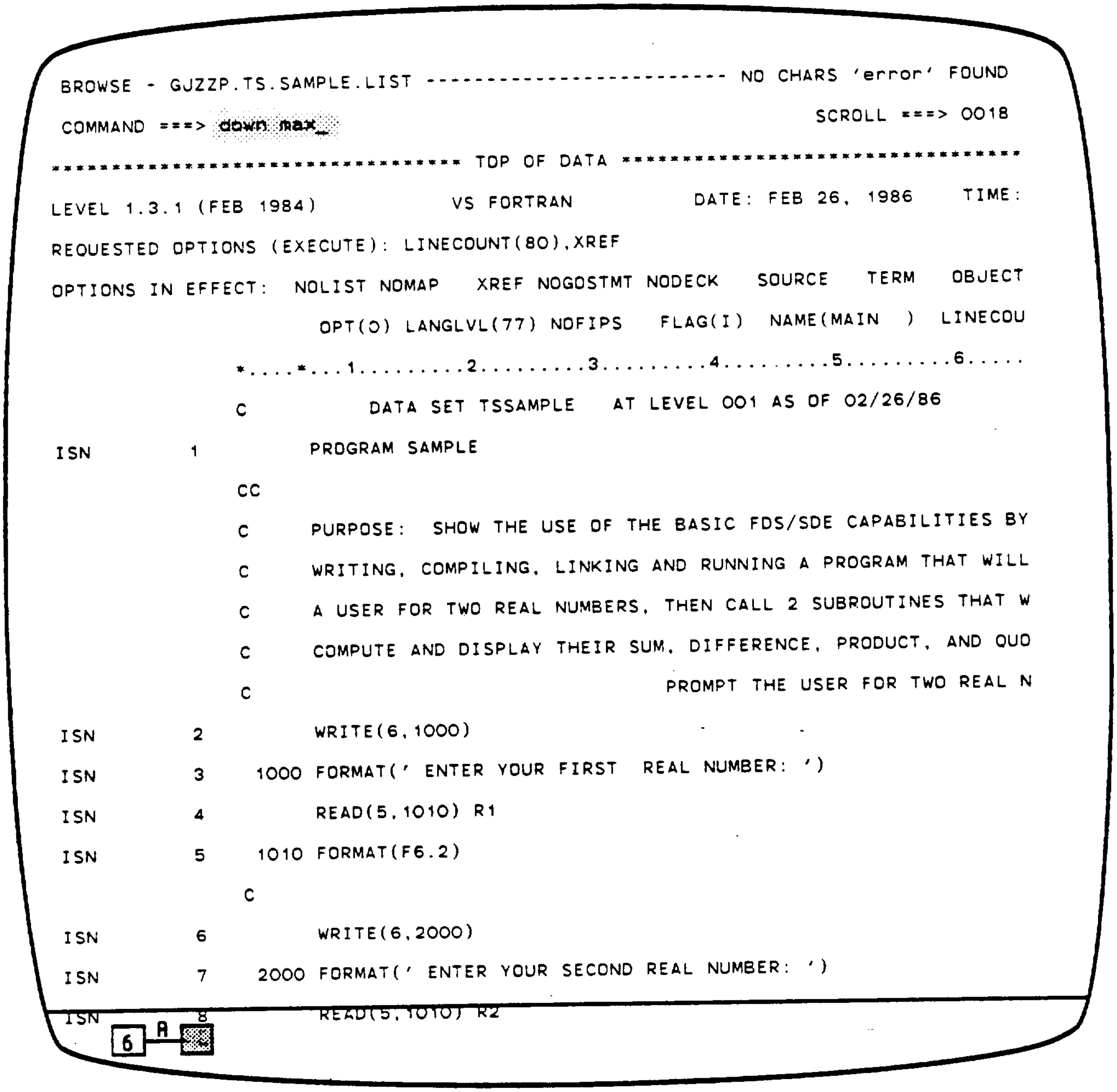
Scenario Step 4: Compiling program and subroutines



```
 BROWSE - GJZZP.TS.SAMPLE.LIST ------------------------- NO CHARS 'error' FOUND
 COMMAND ===> _                                                 SCROLL ===> 0018
********************************** TOP OF DATA **********************************
LEVEL 1.3.1 (FEB 1984)          VS FORTRAN          DATE: FEB 26, 1986     TIME:
REQUESTED OPTIONS (EXECUTE): LINECOUNT(80),XREF
OPTIONS IN EFFECT:  NOLIST NOMAP   XREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOU
               *....*...1.........2.........3.........4.........5.........6.....
               C          DATA SET TSSAMPLE   AT LEVEL 001 AS OF 02/26/86
ISN        1         PROGRAM SAMPLE
               CC
               C     PURPOSE:  SHOW THE USE OF THE BASIC FDS/SDE CAPABILITIES BY
               C     WRITING, COMPILING, LINKING AND RUNNING A PROGRAM THAT WILL
               C     A USER FOR TWO REAL NUMBERS, THEN CALL 2 SUBROUTINES THAT W
               C     COMPUTE AND DISPLAY THEIR SUM, DIFFERENCE, PRODUCT, AND QUO
               C                             PROMPT THE USER FOR TWO REAL N
ISN        2         WRITE(6,1000)
ISN        3   1000 FORMAT(' ENTER YOUR FIRST  REAL NUMBER: ')
ISN        4         READ(5,1010) R1
ISN        5   1010 FORMAT(F6.2)
               C
ISN        6         WRITE(6,2000)
ISN        7   2000 FORMAT(' ENTER YOUR SECOND REAL NUMBER: ')
ISN        8         READ(5,1010) R2
```

. . . and the message NO CHARS 'error' FOUND will appear in the upper right of the display. If the word "error" was in the file, as in an error message, the cursor would be positioned at that location in the file. This lets us see the lines that are wrong without scanning through the entire listing.

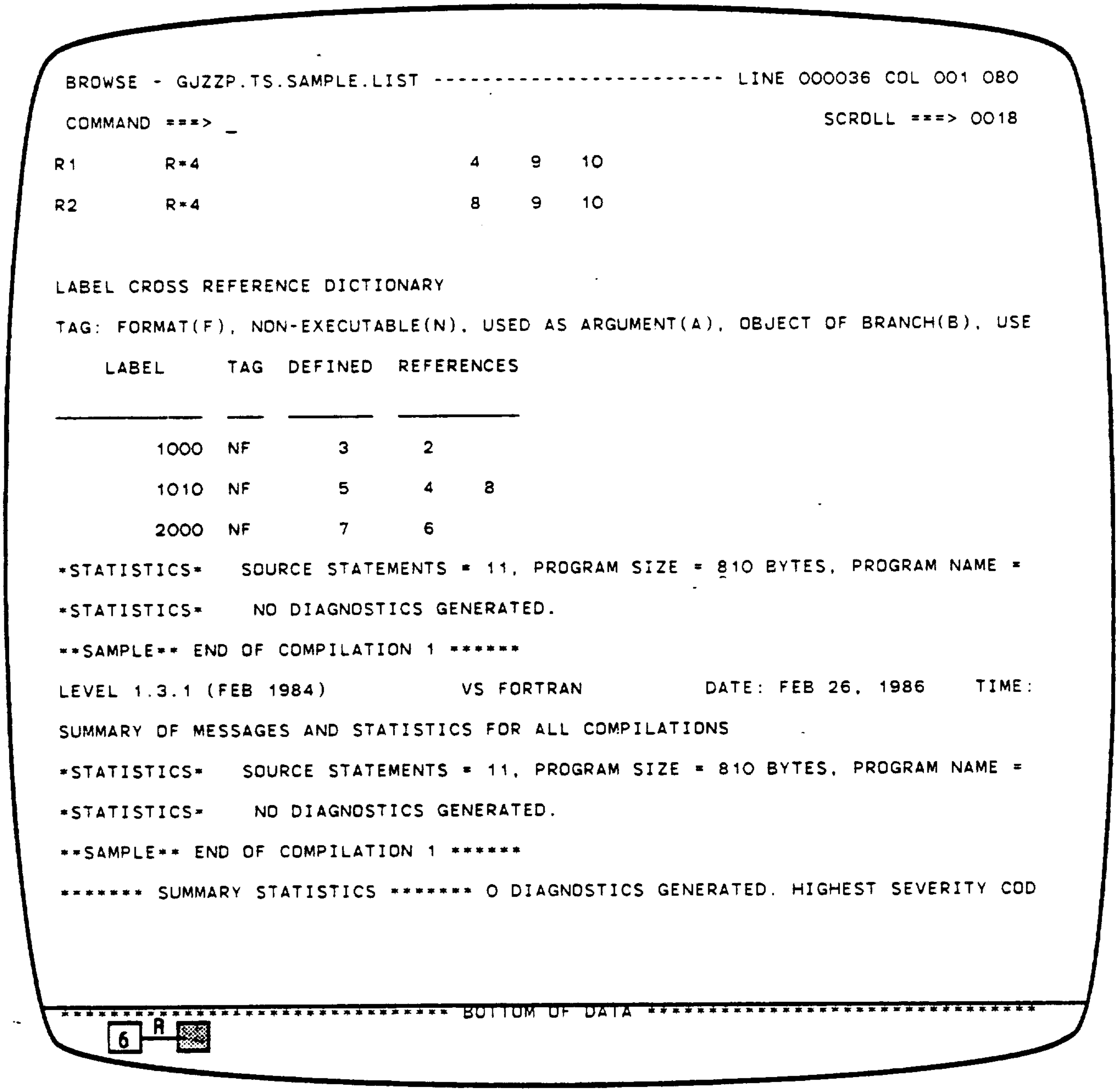
Scenario Step 4: Compiling program and subroutines



```
BROWSE - GJZZP.TS.SAMPLE.LIST ------------------------- NO CHARS 'error' FOUND
COMMAND ===> down max_                                     SCROLL ===> 0018
********************************* TOP OF DATA ********************************
LEVEL 1.3.1 (FEB 1984)         VS FORTRAN         DATE: FEB 26, 1986    TIME:
REQUESTED OPTIONS (EXECUTE): LINECOUNT(80),XREF
OPTIONS IN EFFECT:  NOLIST NOMAP   XREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT
                        OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN    )  LINECOU
                    *....*...1.........2.........3.........4.........5.........6.....
                    C          DATA SET TSSAMPLE   AT LEVEL 001 AS OF 02/26/86
ISN       1               PROGRAM SAMPLE
                    CC
                    C     PURPOSE:  SHOW THE USE OF THE BASIC FDS/SDE CAPABILITIES BY
                    C     WRITING, COMPILING, LINKING AND RUNNING A PROGRAM THAT WILL
                    C     A USER FOR TWO REAL NUMBERS, THEN CALL 2 SUBROUTINES THAT W
                    C     COMPUTE AND DISPLAY THEIR SUM, DIFFERENCE, PRODUCT, AND QUO
                    C                                 PROMPT THE USER FOR TWO REAL N
ISN       2               WRITE(6,1000)
ISN       3         1000 FORMAT(' ENTER YOUR FIRST  REAL NUMBER: ')
ISN       4               READ(5,1010) R1
ISN       5         1010 FORMAT(F6.2)
                    C
ISN       6               WRITE(6,2000)
ISN       7         2000 FORMAT(' ENTER YOUR SECOND REAL NUMBER: ')
ISN       8               READ(5,1010) R2
```

Type 'down max' in the command line field and press <enter> . . .

Scenario Step 4: Compiling program and subroutines

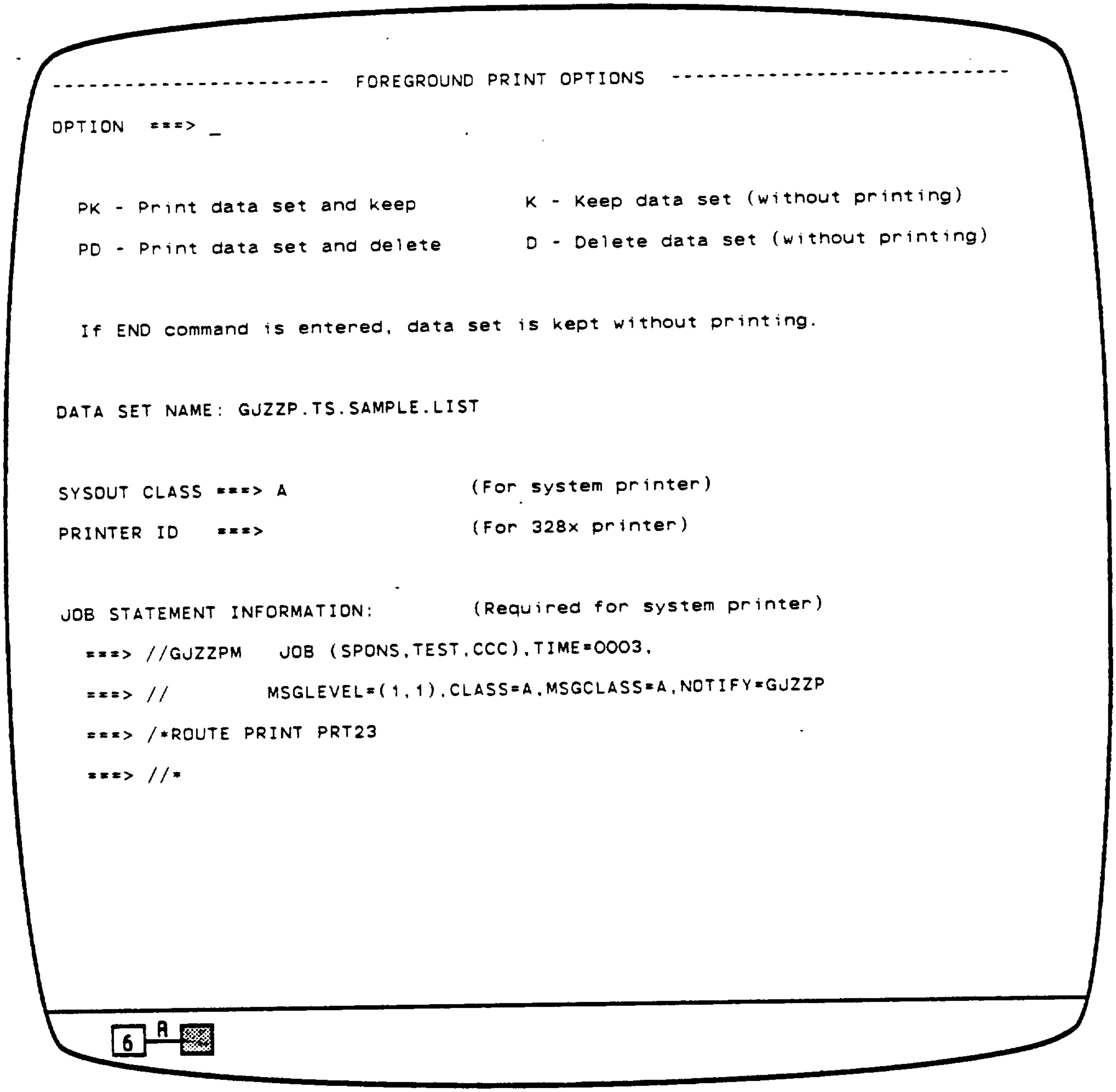


```
 BROWSE - GJZZP.TS.SAMPLE.LIST ----------------------- LINE 000036 COL 001 080
 COMMAND ===> _                                                SCROLL ===> 0018
R1         R*4                         4     9    10
R2         R*4                         8     9    10

LABEL CROSS REFERENCE DICTIONARY
TAG: FORMAT(F), NON-EXECUTABLE(N), USED AS ARGUMENT(A), OBJECT OF BRANCH(B), USE
    LABEL      TAG  DEFINED  REFERENCES
____________  ___  _______  __________
        1000  NF         3       2
        1010  NF         5       4    8
        2000  NF         7       6
*STATISTICS*    SOURCE STATEMENTS = 11, PROGRAM SIZE = 810 BYTES, PROGRAM NAME =
*STATISTICS*     NO DIAGNOSTICS GENERATED.
**SAMPLE** END OF COMPILATION 1 ******
LEVEL 1.3.1 (FEB 1984)           VS FORTRAN           DATE: FEB 26, 1986    TIME:
SUMMARY OF MESSAGES AND STATISTICS FOR ALL COMPILATIONS
*STATISTICS*    SOURCE STATEMENTS = 11, PROGRAM SIZE = 810 BYTES, PROGRAM NAME =
*STATISTICS*     NO DIAGNOSTICS GENERATED.
**SAMPLE** END OF COMPILATION 1 ******
******* SUMMARY STATISTICS ******* 0 DIAGNOSTICS GENERATED. HIGHEST SEVERITY COD

******************************** BOTTOM OF DATA ********************************
```

. . . and we will skip to the bottom of the listing field. In this way we can review the processing statistics, in case they flashed by when we weren't looking.

When you are finished scrolling around the listing file, press <end> . . .

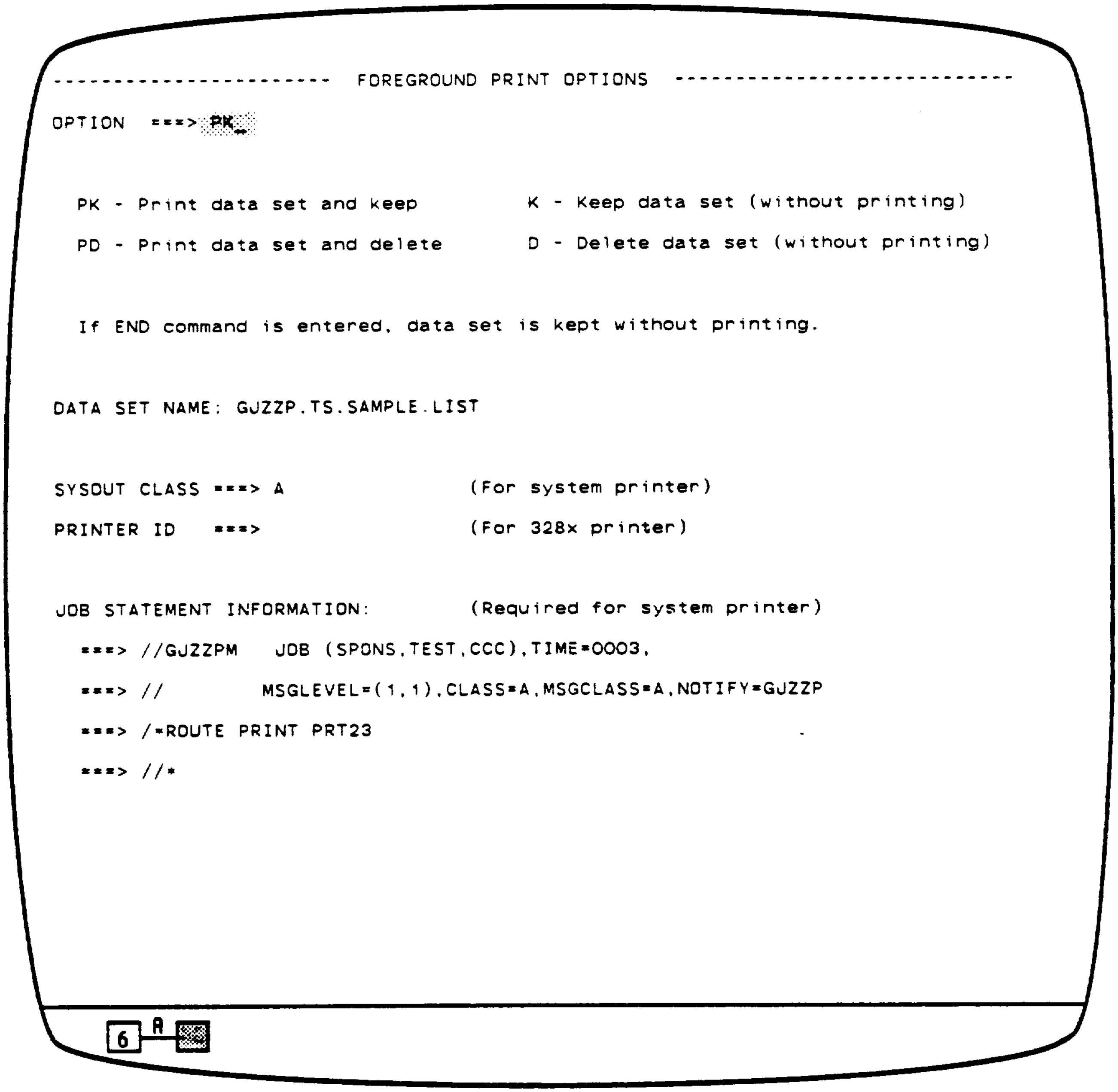
Scenario Step 4: Compiling program and subroutines



```
---------------------- FOREGROUND PRINT OPTIONS --------------------------
OPTION  ===> _

  PK - Print data set and keep          K - Keep data set (without printing)
  PD - Print data set and delete        D - Delete data set (without printing)

  If END command is entered, data set is kept without printing.

DATA SET NAME: GJZZP.TS.SAMPLE.LIST

SYSOUT CLASS ===> A                (For system printer)
PRINTER ID   ===>                  (For 328x printer)

JOB STATEMENT INFORMATION:         (Required for system printer)
  ===> //GJZZPM   JOB (SPONS,TEST,CCC),TIME=0003,
  ===> //         MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP
  ===> /*ROUTE PRINT PRT23
  ===> //*
```

. . . and the print option panel is displayed. Note the job control information is the same as that which we entered in option 0.JC. That's why we did that first.

Scenario Step 4: Compiling program and subroutines

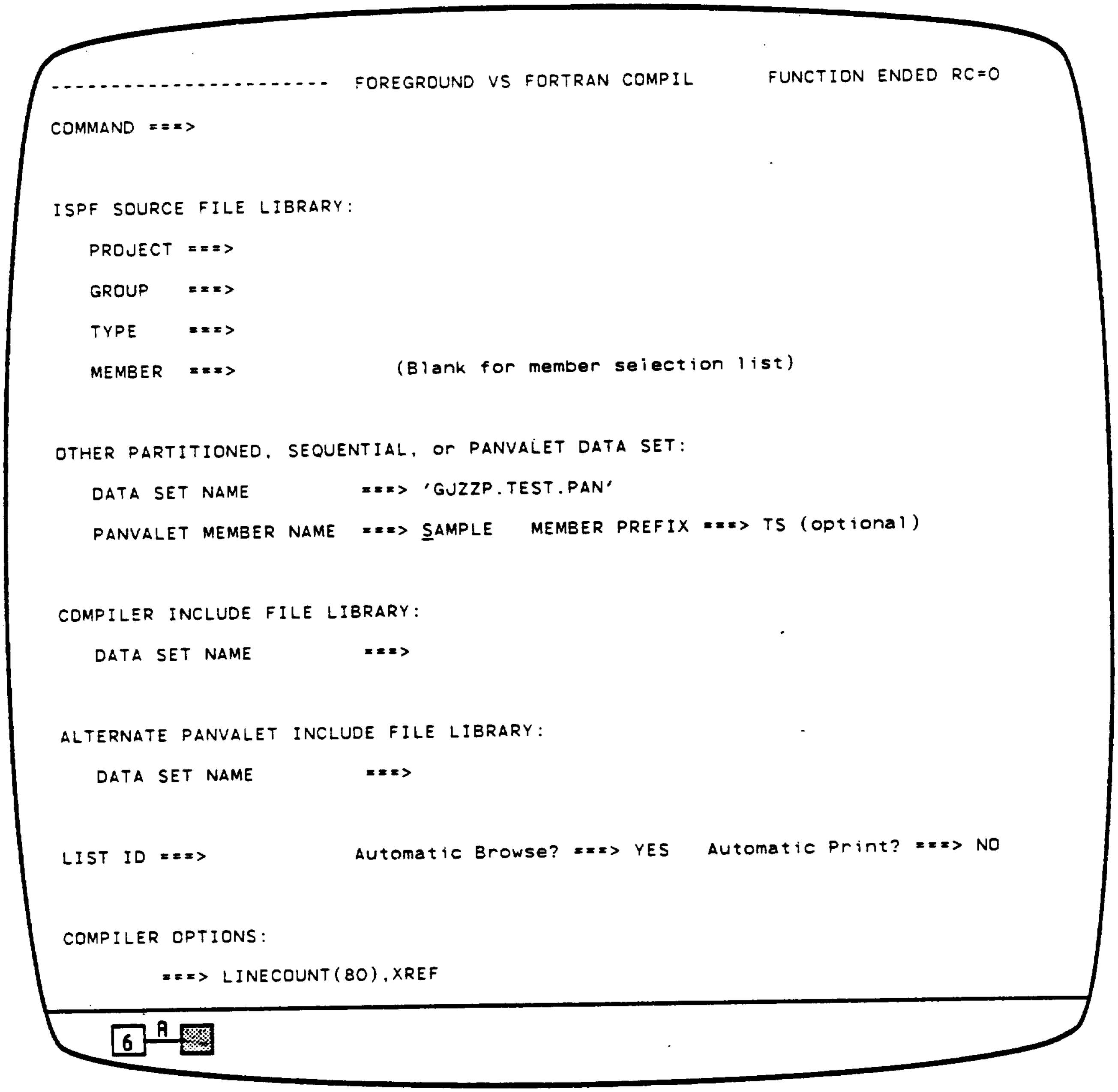


```
---------------------- FOREGROUND PRINT OPTIONS ---------------------------
OPTION  ===> PK


  PK - Print data set and keep          K - Keep data set (without printing)
  PD - Print data set and delete        D - Delete data set (without printing)


  If END command is entered, data set is kept without printing.


DATA SET NAME: GJZZP.TS.SAMPLE.LIST


SYSOUT CLASS ===> A                  (For system printer)
PRINTER ID   ===>                    (For 328x printer)


JOB STATEMENT INFORMATION:           (Required for system printer)
  ===> //GJZZPM   JOB (SPONS,TEST,CCC),TIME=0003,
  ===> //         MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP
  ===> /*ROUTE PRINT PRT23
  ===> //*
```

Type in the option PK to print and keep the listing data set. We want to keep it in order to show you another option later in this scenario.

Press <enter> and . . .

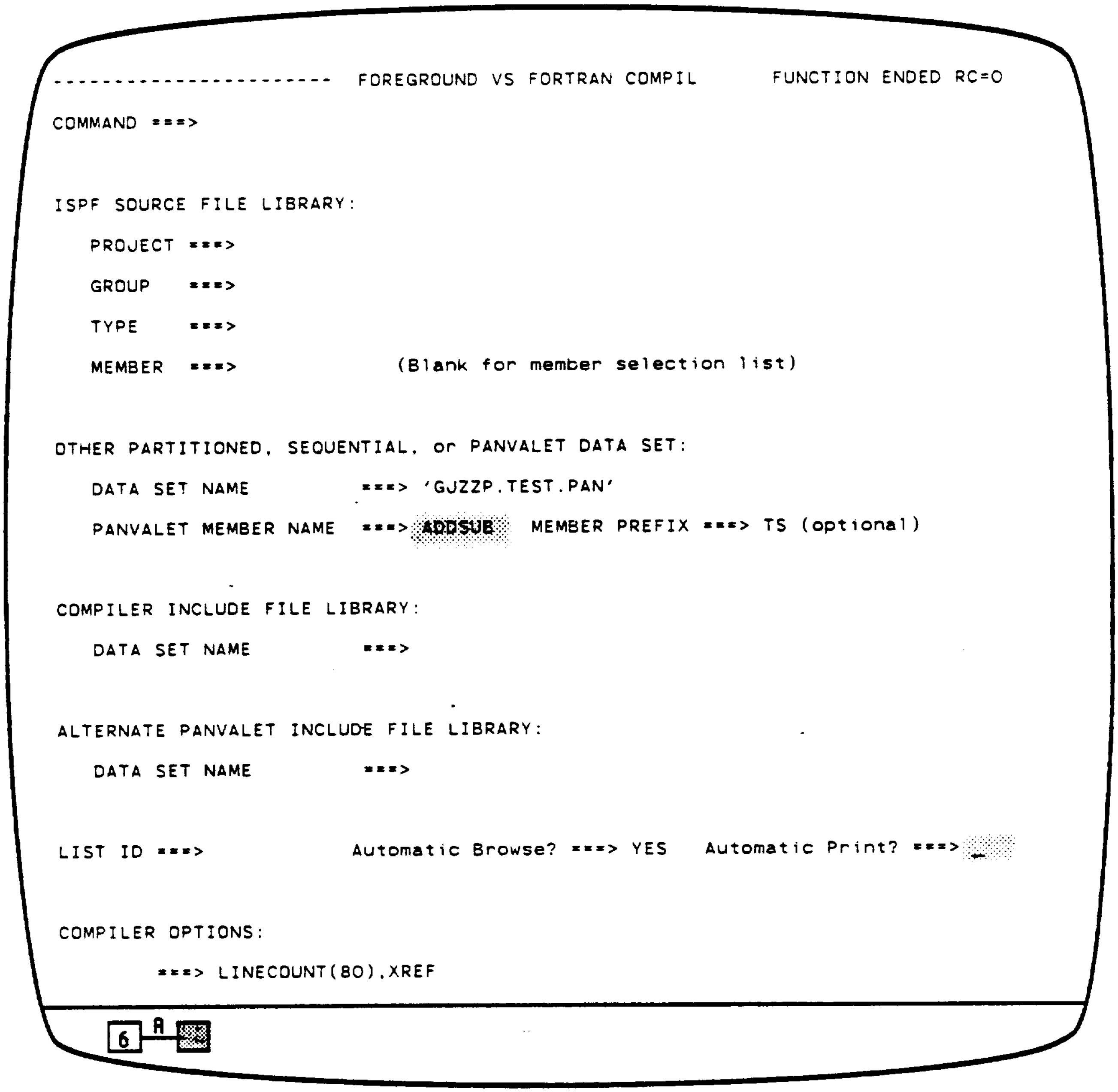
Scenario Step 4: Compiling program and subroutines



```
----------------------- FOREGROUND VS FORTRAN COMPIL      FUNCTION ENDED RC=0
COMMAND ===>

ISPF SOURCE FILE LIBRARY:
   PROJECT ===>
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>             (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
   DATA SET NAME          ===> 'GJZZP.TEST.PAN'
   PANVALET MEMBER NAME   ===> SAMPLE   MEMBER PREFIX ===> TS (optional)

COMPILER INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>

ALTERNATE PANVALET INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>

LIST ID ===>            Automatic Browse? ===> YES   Automatic Print? ===> NO

COMPILER OPTIONS:
        ===> LINECOUNT(80),XREF
```

The VS FORTRAN compile parameter entry panel is redisplayed. Note the short message in the upper right corner of the display. This message indicates the compiler function ended with a return code of zero, meaning no errors were found. The next thing to compile is the ADDSUB subroutine, so we position the cursor at the Panvalet member name field and . . .

Scenario Step 4: Compiling program and subroutines

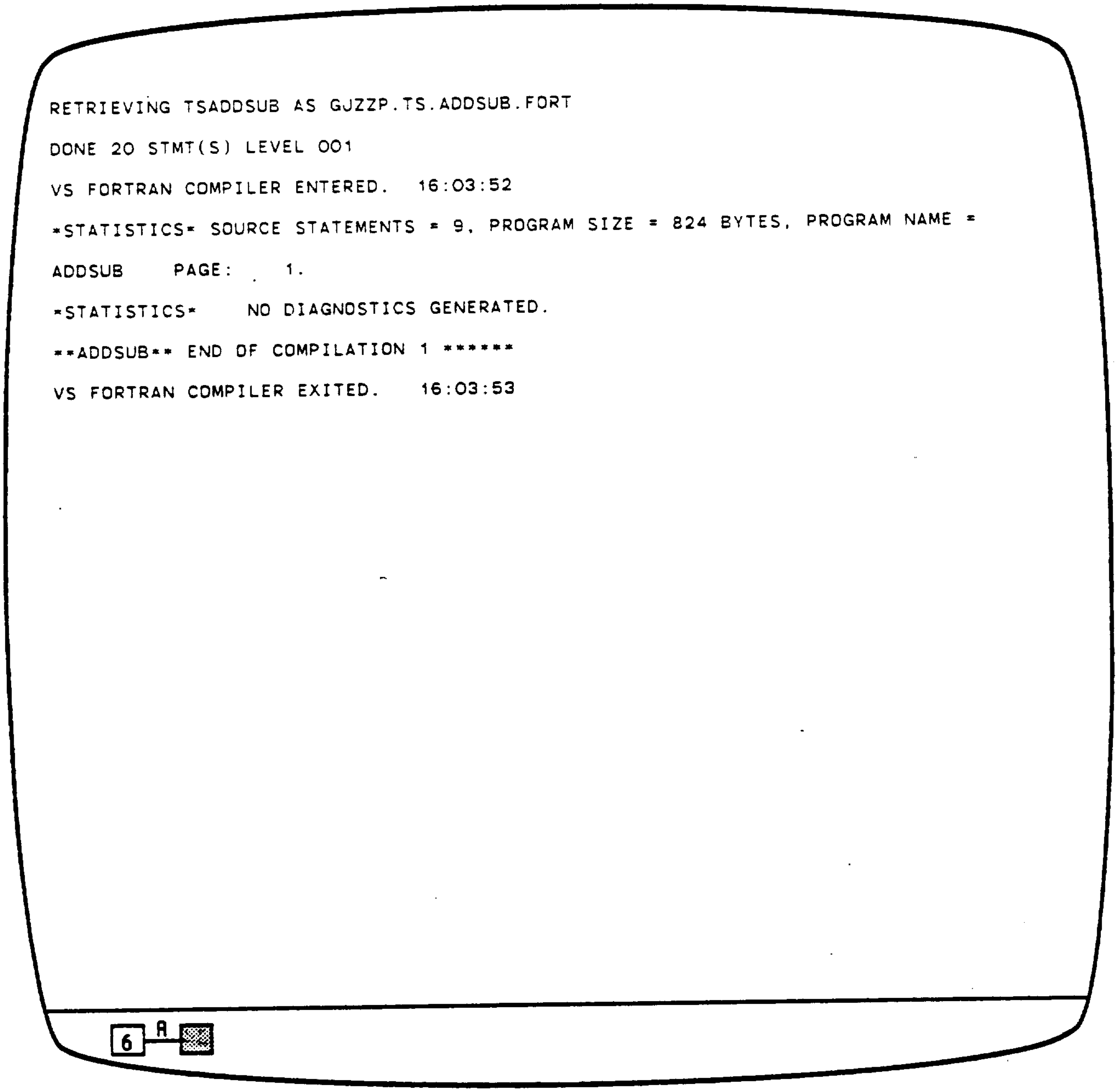


```
----------------------- FOREGROUND VS FORTRAN COMPIL      FUNCTION ENDED RC=0
COMMAND ===>

ISPF SOURCE FILE LIBRARY:
   PROJECT ===>
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>            (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
   DATA SET NAME        ===> 'GJZZP.TEST.PAN'
   PANVALET MEMBER NAME ===> ADDSUB   MEMBER PREFIX ===> TS (optional)

COMPILER INCLUDE FILE LIBRARY:
   DATA SET NAME        ===>

ALTERNATE PANVALET INCLUDE FILE LIBRARY:
   DATA SET NAME        ===>

LIST ID ===>           Automatic Browse? ===> YES   Automatic Print? ===> _

COMPILER OPTIONS:
       ===> LINECOUNT(80),XREF
```

. . . type ADDSUB over top of SAMPLE. We will print the listing file later using another option, so <FT> to the automatic print question and blank out the entry with the <EEOF> key. All blanks are converted to NO by the SDE.

Press <enter> . . .

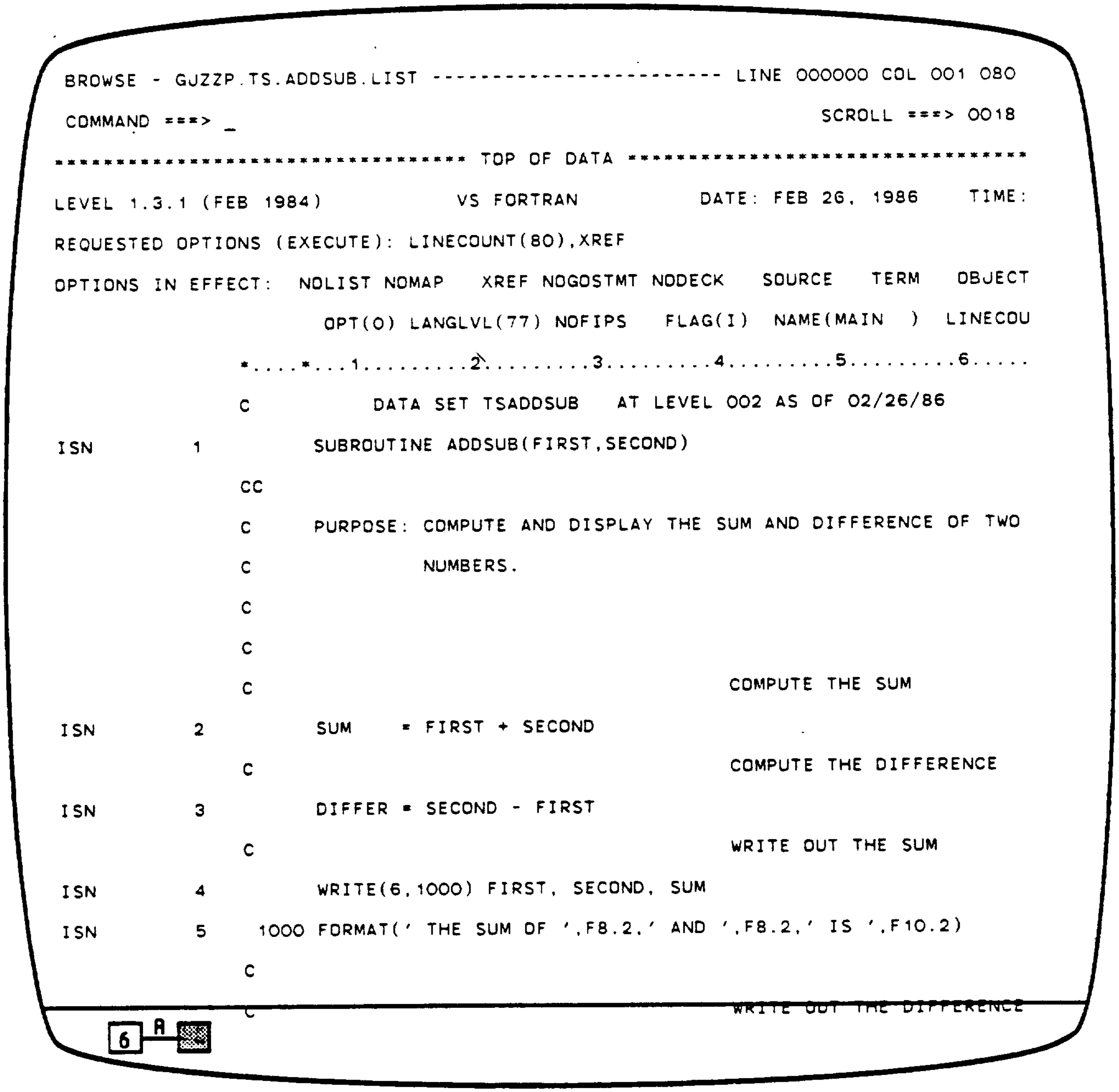
Scenario Step 4: Compiling program and subroutines



```
RETRIEVING TSADDSUB AS GJZZP.TS.ADDSUB.FORT
DONE 20 STMT(S) LEVEL 001
VS FORTRAN COMPILER ENTERED.  16:03:52
*STATISTICS* SOURCE STATEMENTS = 9, PROGRAM SIZE = 824 BYTES, PROGRAM NAME =
ADDSUB    PAGE:   1.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
**ADDSUB** END OF COMPILATION 1 ******
VS FORTRAN COMPILER EXITED.   16:03:53
```

. . . and just like the last compile, the screen will clear, the messages shown above will be displayed, and shortly after the last message . . .

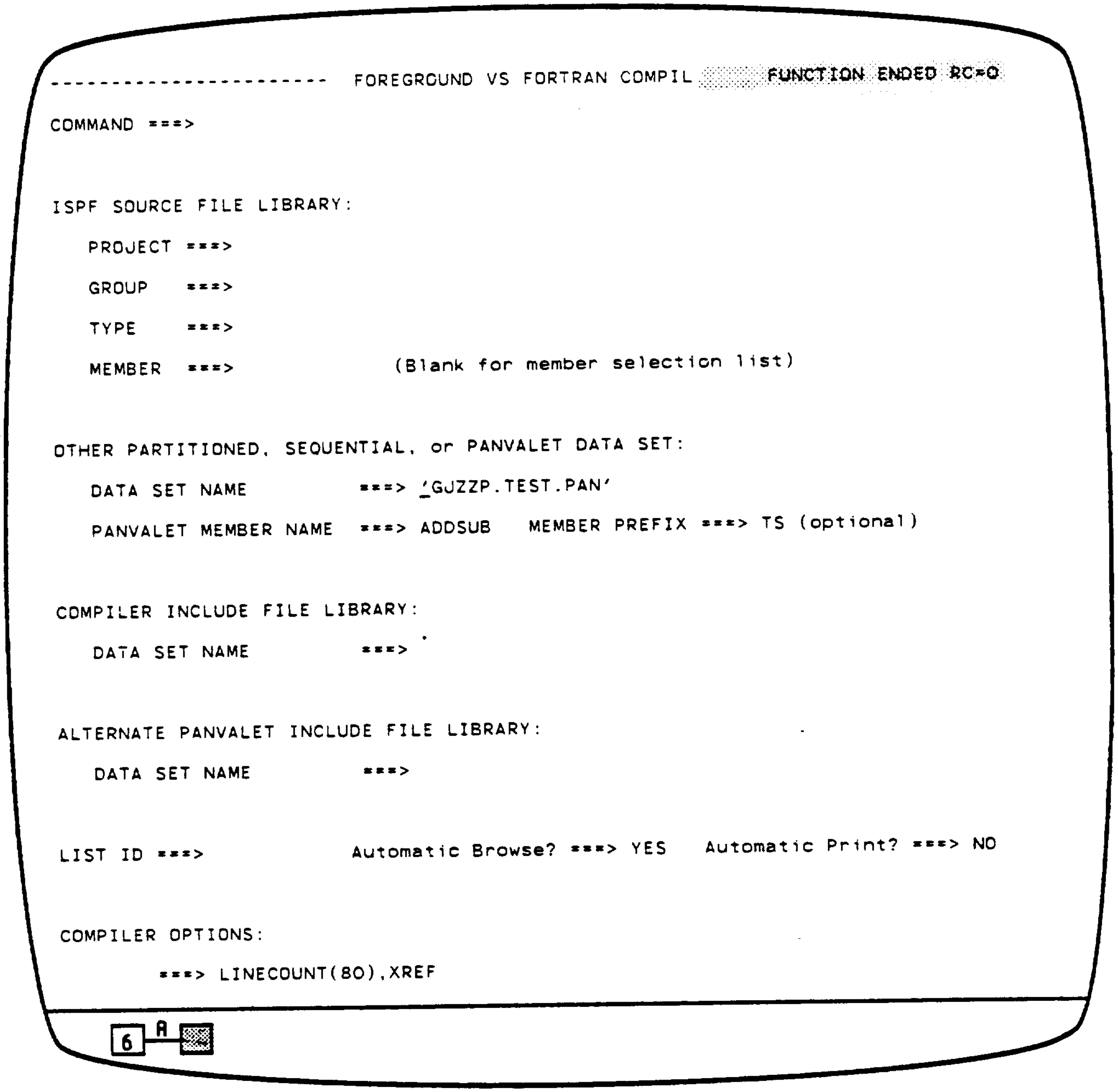
Scenario Step 4: Compiling program and subroutines



```
 BROWSE - GJZZP.TS.ADDSUB.LIST ------------------------ LINE 000000 COL 001 080
 COMMAND ===> _                                                SCROLL ===> 0018
********************************* TOP OF DATA *********************************
LEVEL 1.3.1 (FEB 1984)           VS FORTRAN          DATE: FEB 26, 1986    TIME:
REQUESTED OPTIONS (EXECUTE): LINECOUNT(80),XREF
OPTIONS IN EFFECT:  NOLIST NOMAP   XREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT
                      OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOU
               *....*...1.........2.........3.........4.........5.........6.....
               C          DATA SET TSADDSUB    AT LEVEL 002 AS OF 02/26/86
ISN       1          SUBROUTINE ADDSUB(FIRST,SECOND)
               CC
               C     PURPOSE: COMPUTE AND DISPLAY THE SUM AND DIFFERENCE OF TWO
               C              NUMBERS.
               C
               C
               C                                      COMPUTE THE SUM
ISN       2          SUM    = FIRST + SECOND
               C                                      COMPUTE THE DIFFERENCE
ISN       3          DIFFER = SECOND - FIRST
               C                                      WRITE OUT THE SUM
ISN       4          WRITE(6,1000) FIRST, SECOND, SUM
ISN       5     1000 FORMAT(' THE SUM OF ',F8.2,' AND ',F8.2,' IS ',F10.2)
               C
               C                                      WRITE OUT THE DIFFERENCE
```

. . . we will automatically browse this listing file. Press <end>, and . . .

Scenario Step 4: Compiling program and subroutines

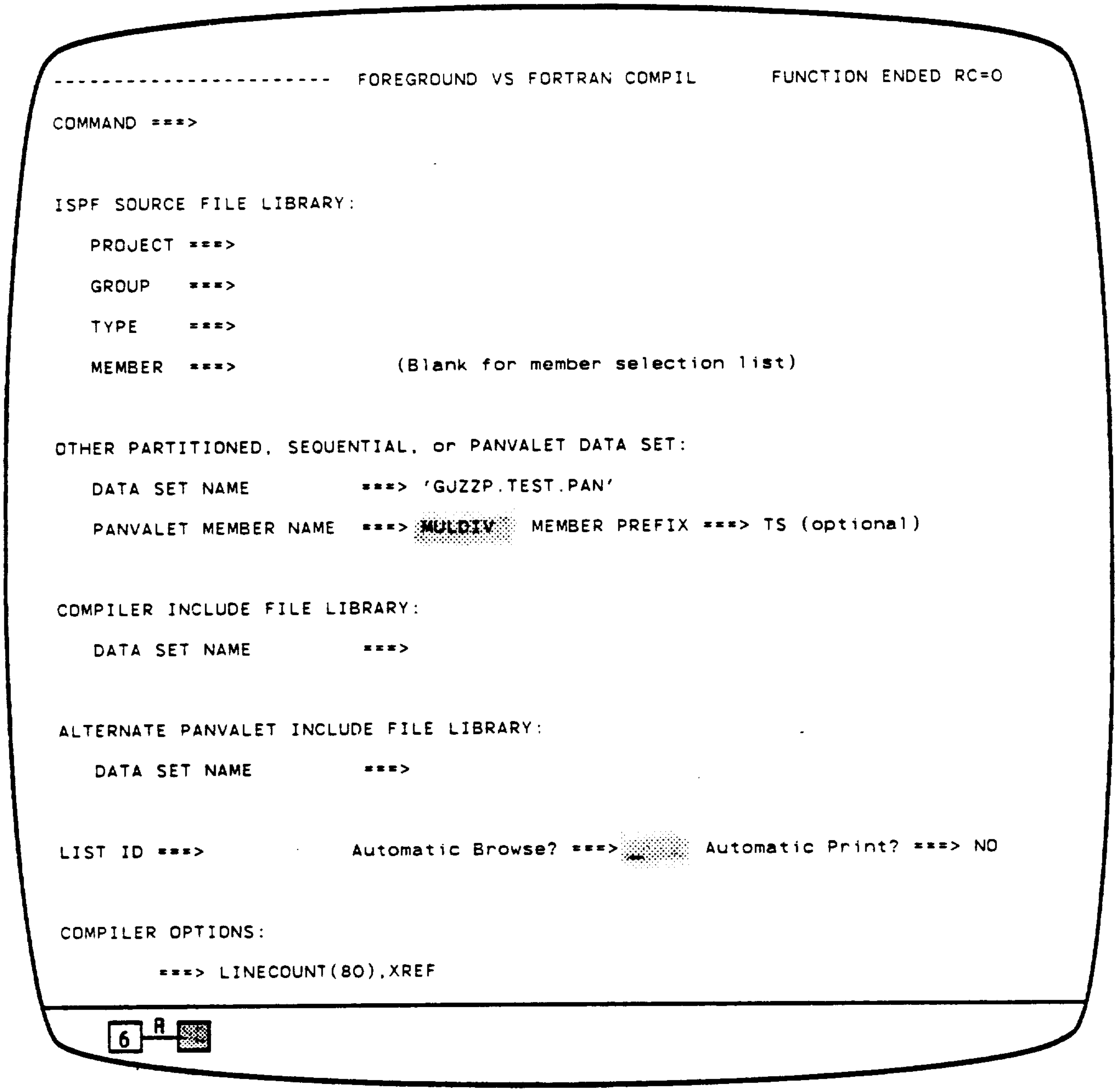


```
----------------------- FOREGROUND VS FORTRAN COMPIL    FUNCTION ENDED RC=0
COMMAND ===>


ISPF SOURCE FILE LIBRARY:
   PROJECT ===>
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>              (Blank for member selection list)


OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
   DATA SET NAME          ===> 'GJZZP.TEST.PAN'
   PANVALET MEMBER NAME   ===> ADDSUB   MEMBER PREFIX ===> TS (optional)


COMPILER INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>


ALTERNATE PANVALET INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>


LIST ID ===>             Automatic Browse? ===> YES   Automatic Print? ===> NO


COMPILER OPTIONS:
        ===> LINECOUNT(80),XREF
```

. . . and the VS FORTRAN compiler parameter entry panel is again displayed.

Note the short message indicates no errors were found (return code = 0). Object module 'GJZZP.TS.ADDSUB.OBJ' (or TS.ADDSUB.OBJ) has been created.

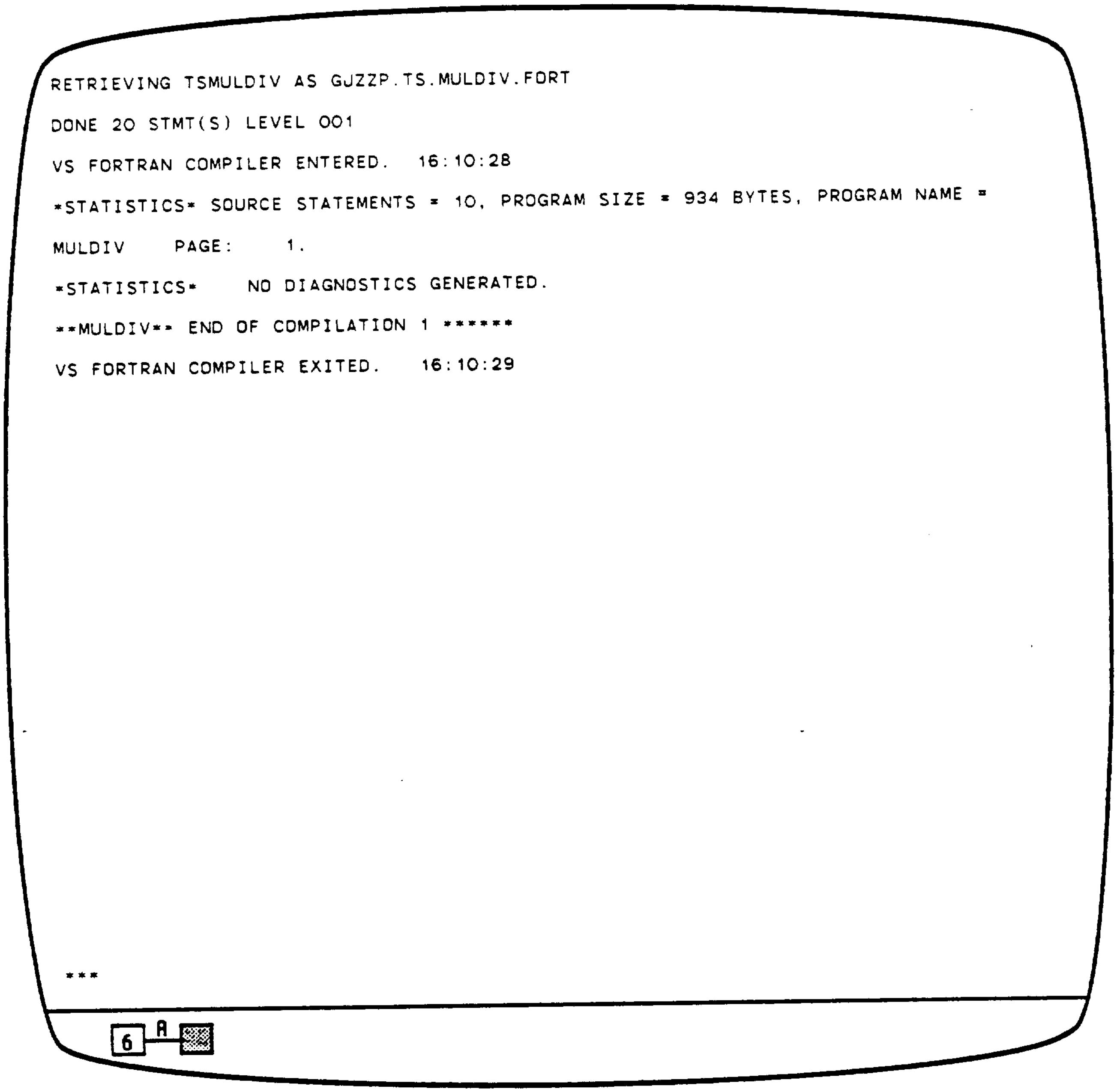
Scenario Step 4: Compiling program and subroutines



```
----------------------  FOREGROUND VS FORTRAN COMPIL      FUNCTION ENDED RC=0
COMMAND ===>


ISPF SOURCE FILE LIBRARY:
   PROJECT ===>
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>             (Blank for member selection list)


OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
   DATA SET NAME          ===> 'GJZZP.TEST.PAN'
   PANVALET MEMBER NAME   ===> MULDIV  MEMBER PREFIX ===> TS (optional)


COMPILER INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>


ALTERNATE PANVALET INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>


LIST ID ===>          Automatic Browse? ===> _   Automatic Print? ===> NO


COMPILER OPTIONS:
        ===> LINECOUNT(80),XREF
```

Now it's time to create the last object module, TS.MULDIV.OBJ. Type in MULDIV as the member name, <FT> to the automatic browse question and blank it out; then press <enter>.

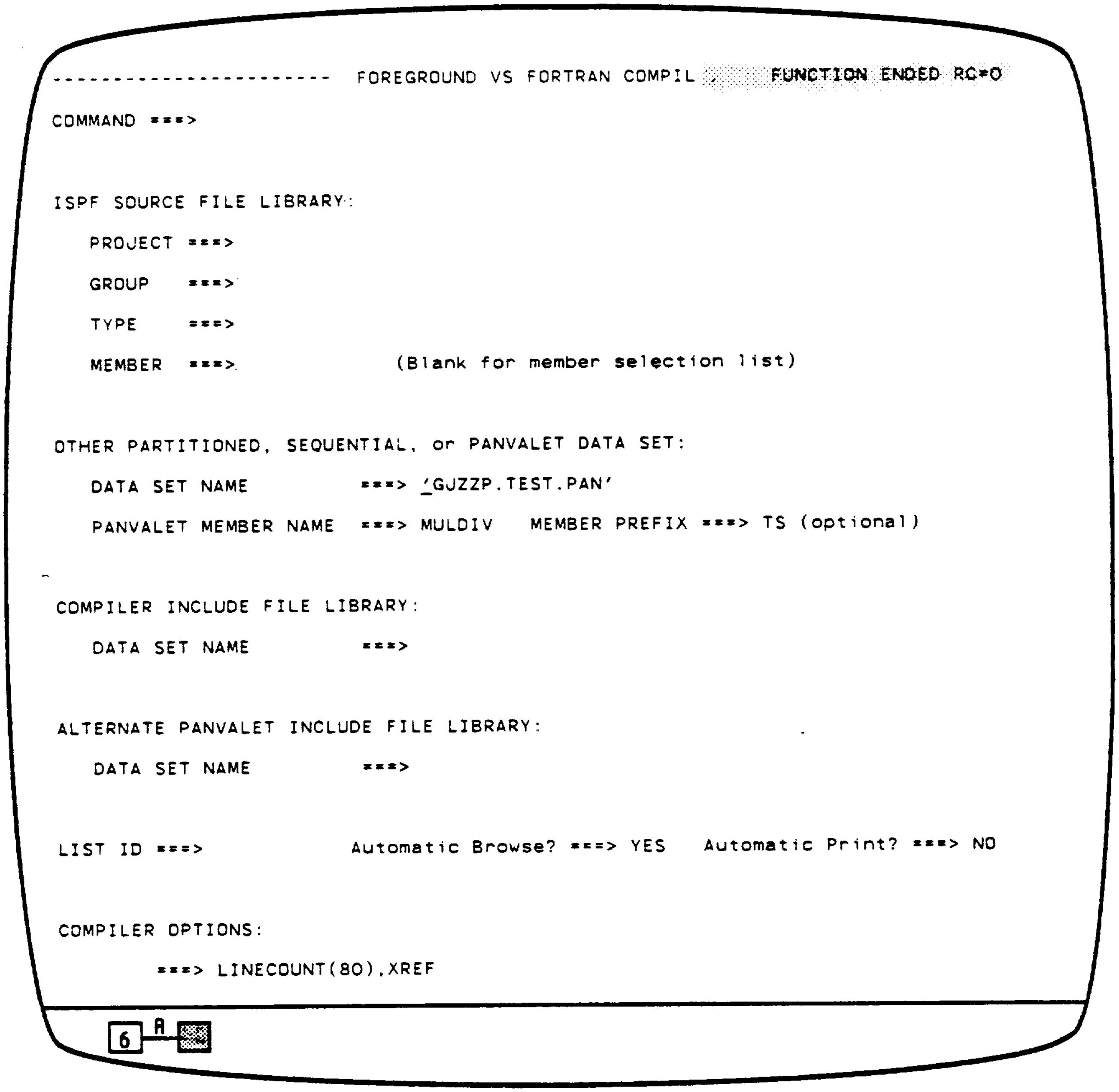
Scenario Step 4: Compiling program and subroutines



```
RETRIEVING TSMULDIV AS GJZZP.TS.MULDIV.FORT
DONE 20 STMT(S) LEVEL 001
VS FORTRAN COMPILER ENTERED.  16:10:28
*STATISTICS* SOURCE STATEMENTS = 10, PROGRAM SIZE = 934 BYTES, PROGRAM NAME =
MULDIV    PAGE:     1.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
**MULDIV** END OF COMPILATION 1 ******
VS FORTRAN COMPILER EXITED.   16:10:29







 ***
```

The messages for the compile will be displayed, but this time three asterisks will appear at the bottom of the screen. Three asterisks are displayed until we acknowledge having seen the results. Press <enter> to acknowledge, and . . .

Scenario Step 4: Compiling program and subroutines

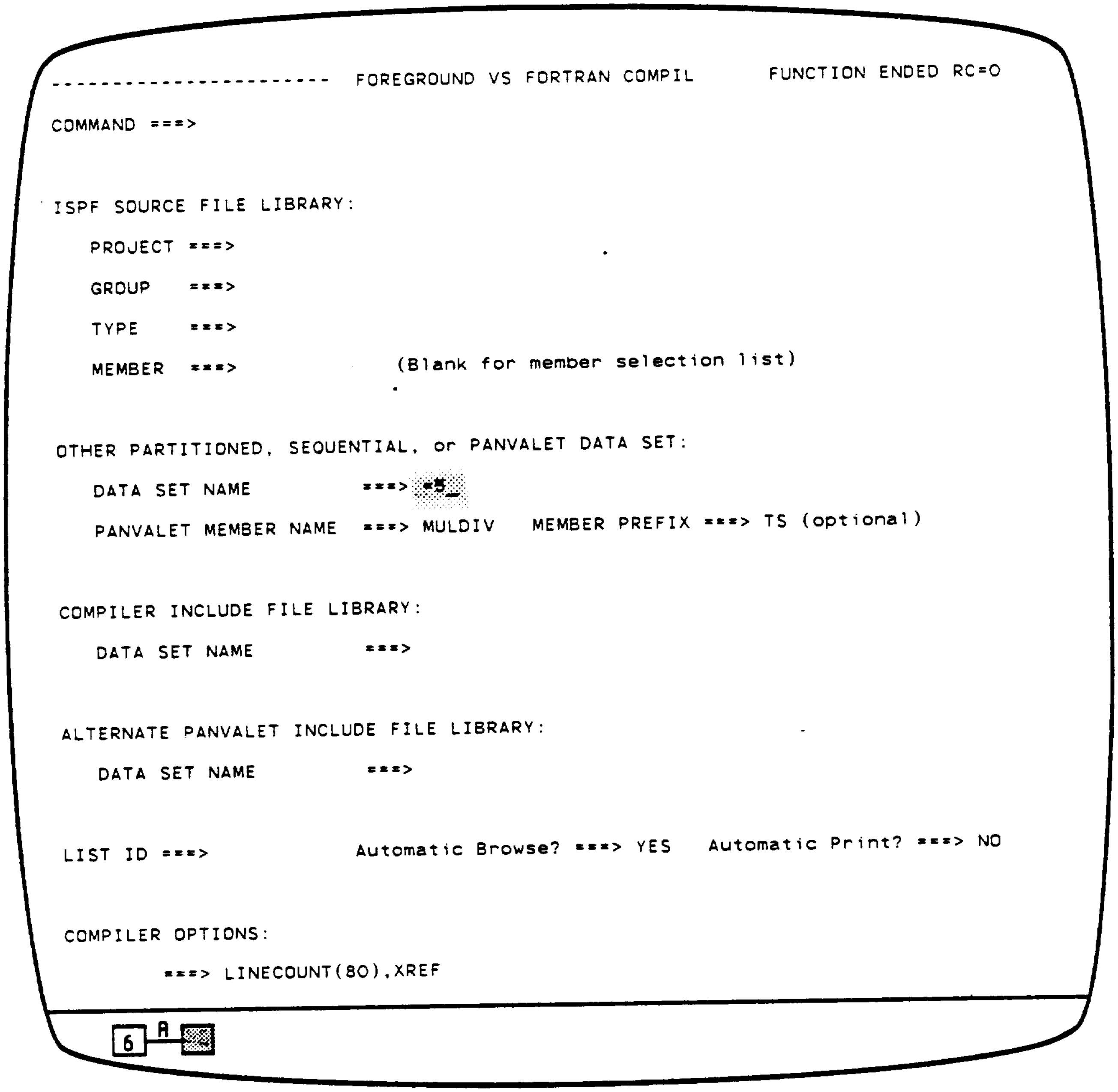


```
----------------------  FOREGROUND VS FORTRAN COMPIL   FUNCTION ENDED RC=0
COMMAND ===>


ISPF SOURCE FILE LIBRARY:

   PROJECT ===>

   GROUP   ===>

   TYPE    ===>

   MEMBER  ===>            (Blank for member selection list)


OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:

   DATA SET NAME          ===> 'GJZZP.TEST.PAN'

   PANVALET MEMBER NAME   ===> MULDIV   MEMBER PREFIX ===> TS (optional)


COMPILER INCLUDE FILE LIBRARY:

   DATA SET NAME          ===>


ALTERNATE PANVALET INCLUDE FILE LIBRARY:

   DATA SET NAME          ===>


LIST ID ===>            Automatic Browse? ===> YES   Automatic Print? ===> NO


COMPILER OPTIONS:

        ===> LINECOUNT(80),XREF
```

. . . and the VS FORTRAN compile parameter entry screen is redisplayed. Note by the short message that the function once again ended with a return code of 0. Boy are we good (actually this scenario was rigged).

This concludes step 4 of our scenario. Now it's time to learn a handy trick.

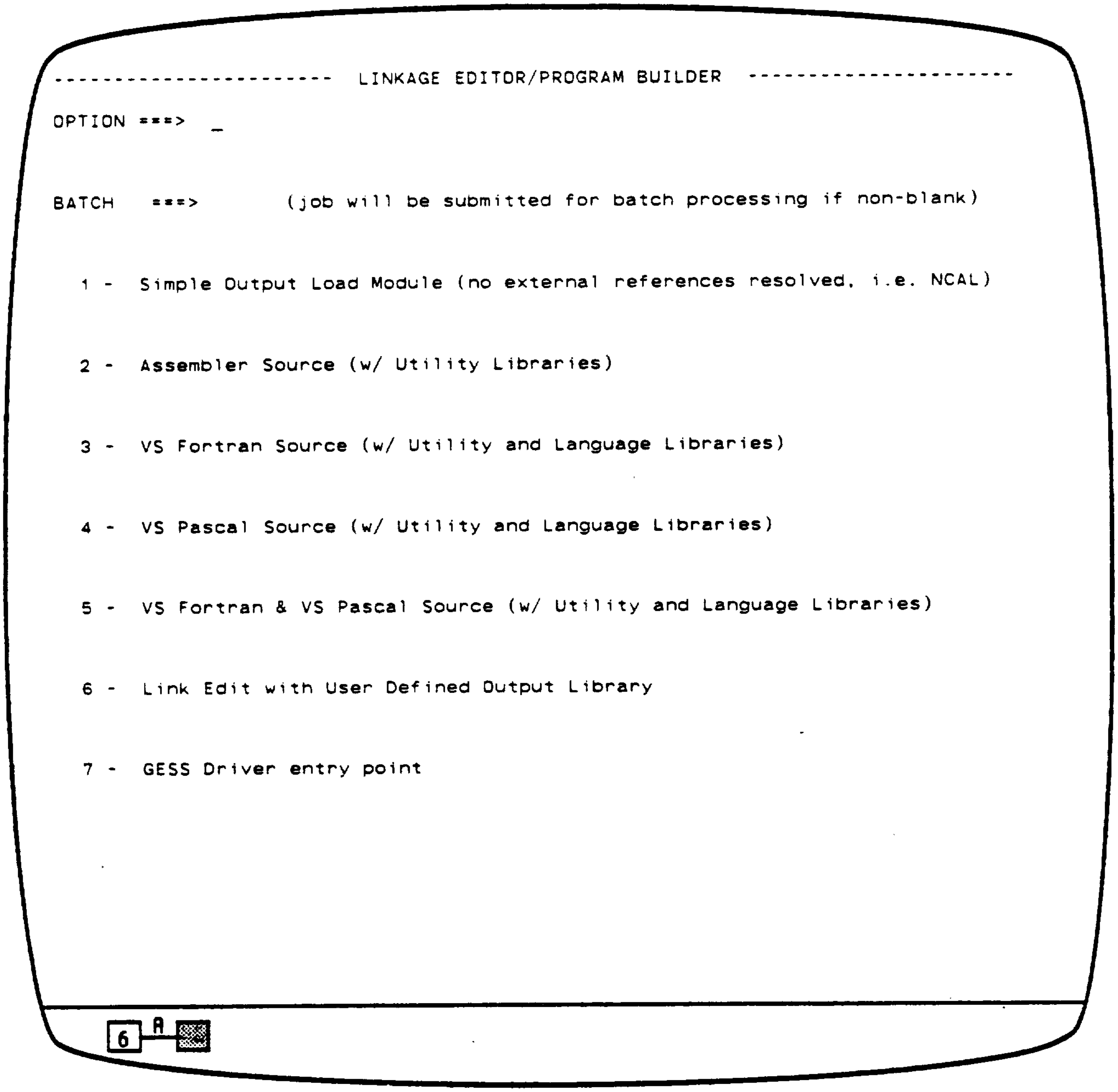
Scenario Step 5: Linking program



```
----------------------  FOREGROUND VS FORTRAN COMPIL      FUNCTION ENDED RC=0
COMMAND ===>

ISPF SOURCE FILE LIBRARY:
   PROJECT ===>
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>               (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
   DATA SET NAME          ===> =5_
   PANVALET MEMBER NAME   ===> MULDIV   MEMBER PREFIX ===> TS (optional)

COMPILER INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>

ALTERNATE PANVALET INCLUDE FILE LIBRARY:
   DATA SET NAME          ===>

LIST ID ===>             Automatic Browse? ===> YES   Automatic Print? ===> NO

COMPILER OPTIONS:
        ===> LINECOUNT(80),XREF
```

What we're going to do is jump directly to the link edit function, without pressing <end> to back up to the primary option panel and descending down the path to the link edit function. To do this, position the cursor in any input field and type in =5 (this means "select option 5 from the primary option panel"). Press <enter> . . .

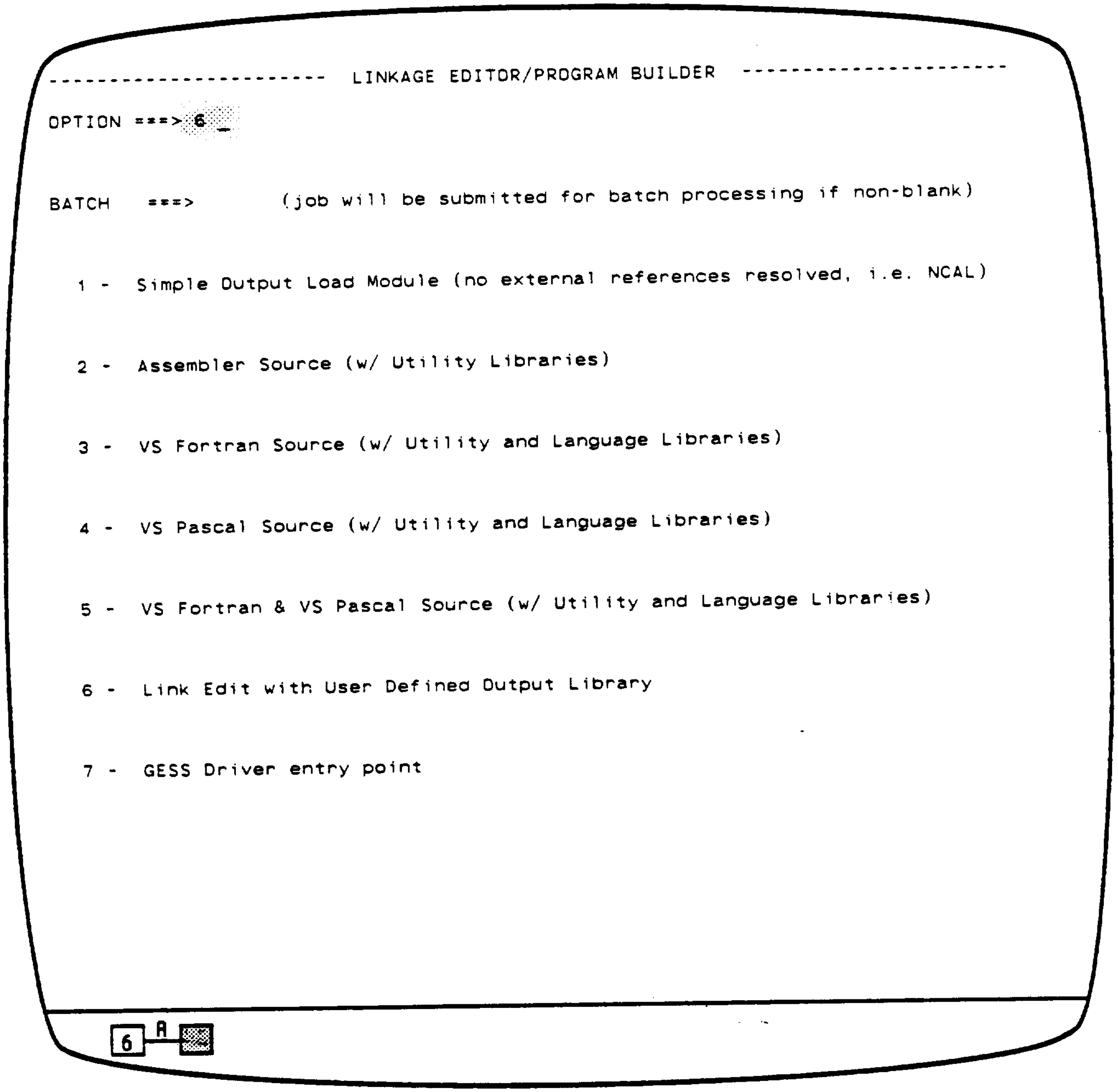
Scenario Step 5: Linking program



```
----------------------  LINKAGE EDITOR/PROGRAM BUILDER  ---------------------
OPTION ===> _

BATCH   ===>       (job will be submitted for batch processing if non-blank)

  1 -  Simple Output Load Module (no external references resolved, i.e. NCAL)

  2 -  Assembler Source (w/ Utility Libraries)

  3 -  VS Fortran Source (w/ Utility and Language Libraries)

  4 -  VS Pascal Source (w/ Utility and Language Libraries)

  5 -  VS Fortran & VS Pascal Source (w/ Utility and Language Libraries)

  6 -  Link Edit with User Defined Output Library

  7 -  GESS Driver entry point
```

. . . and we scoot to the linkage editor selection panel. This is option 5 from the primary option panel. We want to build a program using a temporary library that we will define, so . . .

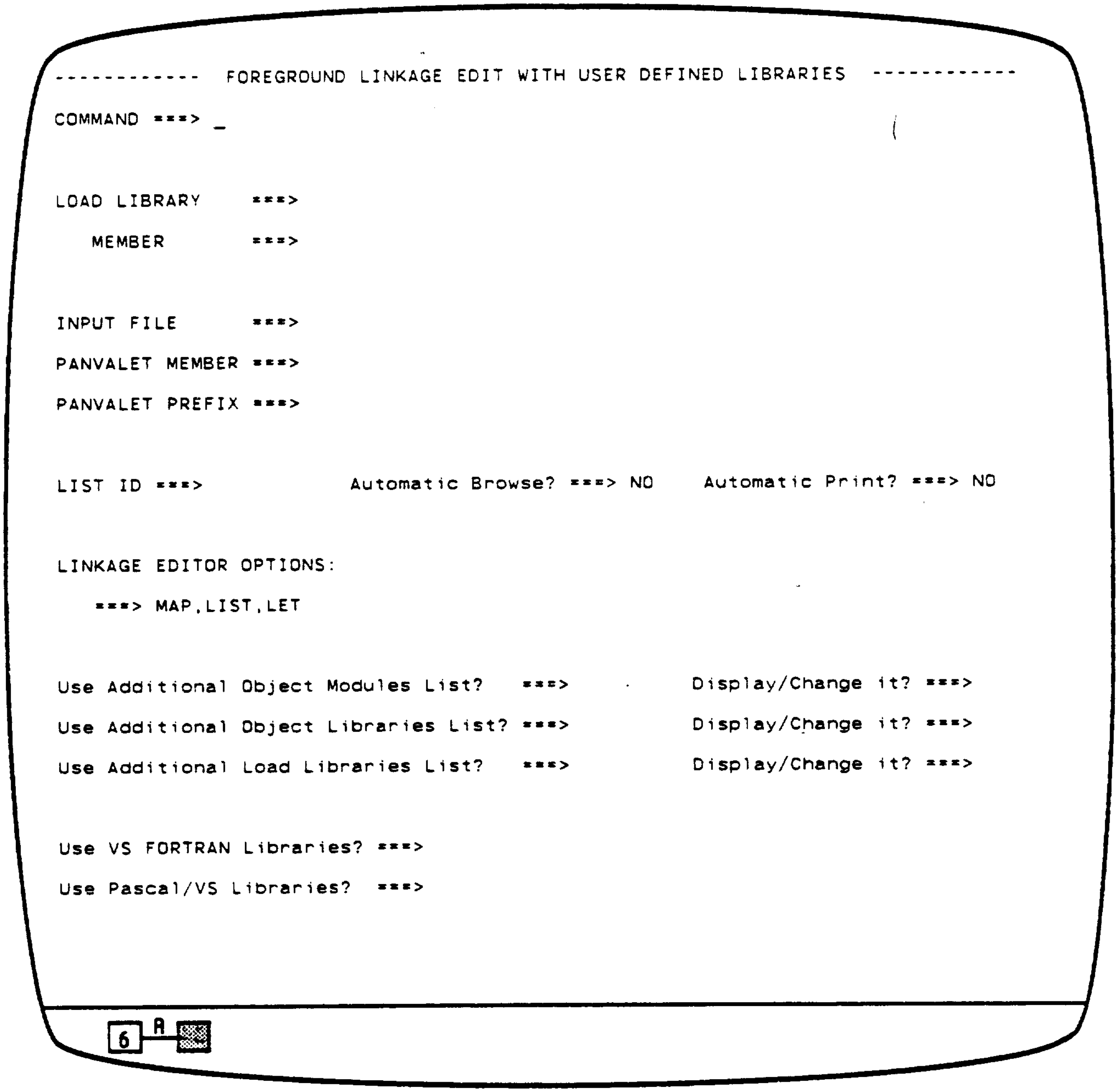
Scenario Step 5: Linking program



```
----------------------- LINKAGE EDITOR/PROGRAM BUILDER ---------------------
OPTION ===> 6 _

BATCH   ===>       (job will be submitted for batch processing if non-blank)

  1 -  Simple Output Load Module (no external references resolved, i.e. NCAL)

  2 -  Assembler Source (w/ Utility Libraries)

  3 -  VS Fortran Source (w/ Utility and Language Libraries)

  4 -  VS Pascal Source (w/ Utility and Language Libraries)

  5 -  VS Fortran & VS Pascal Source (w/ Utility and Language Libraries)

  6 -  Link Edit with User Defined Output Library

  7 -  GESS Driver entry point
```

. . . select option 6 and press <enter>.

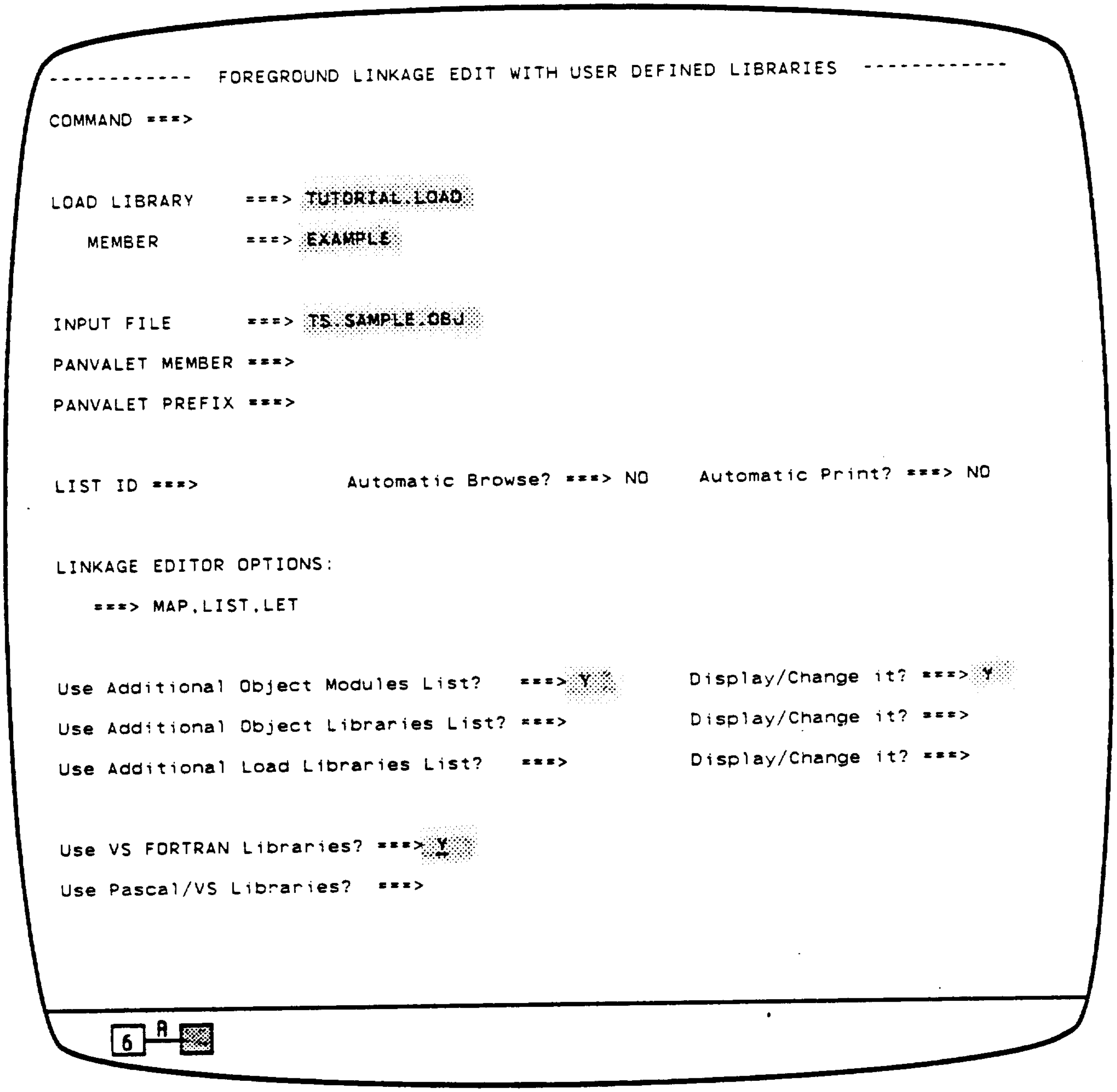
Scenario Step 5: Linking program



```
------------  FOREGROUND LINKAGE EDIT WITH USER DEFINED LIBRARIES  ------------
COMMAND ===> _


LOAD LIBRARY     ===>
   MEMBER        ===>


INPUT FILE       ===>
PANVALET MEMBER ===>
PANVALET PREFIX ===>


LIST ID ===>             Automatic Browse? ===> NO    Automatic Print? ===> NO


LINKAGE EDITOR OPTIONS:
   ===> MAP,LIST,LET


Use Additional Object Modules List?   ===>          Display/Change it? ===>
Use Additional Object Libraries List? ===>          Display/Change it? ===>
Use Additional Load Libraries List?   ===>          Display/Change it? ===>


Use VS FORTRAN Libraries? ===>
Use Pascal/VS Libraries?  ===>
```

The user-defined libraries linkage edit parameter entry panel is displayed. Note the automatic browse and automatic print answers are the same as we saw in the VS FORTRAN compile panel. The answers to these two questions will always be retained between functions and between SDE sessions.

Scenario Step 5: Linking program

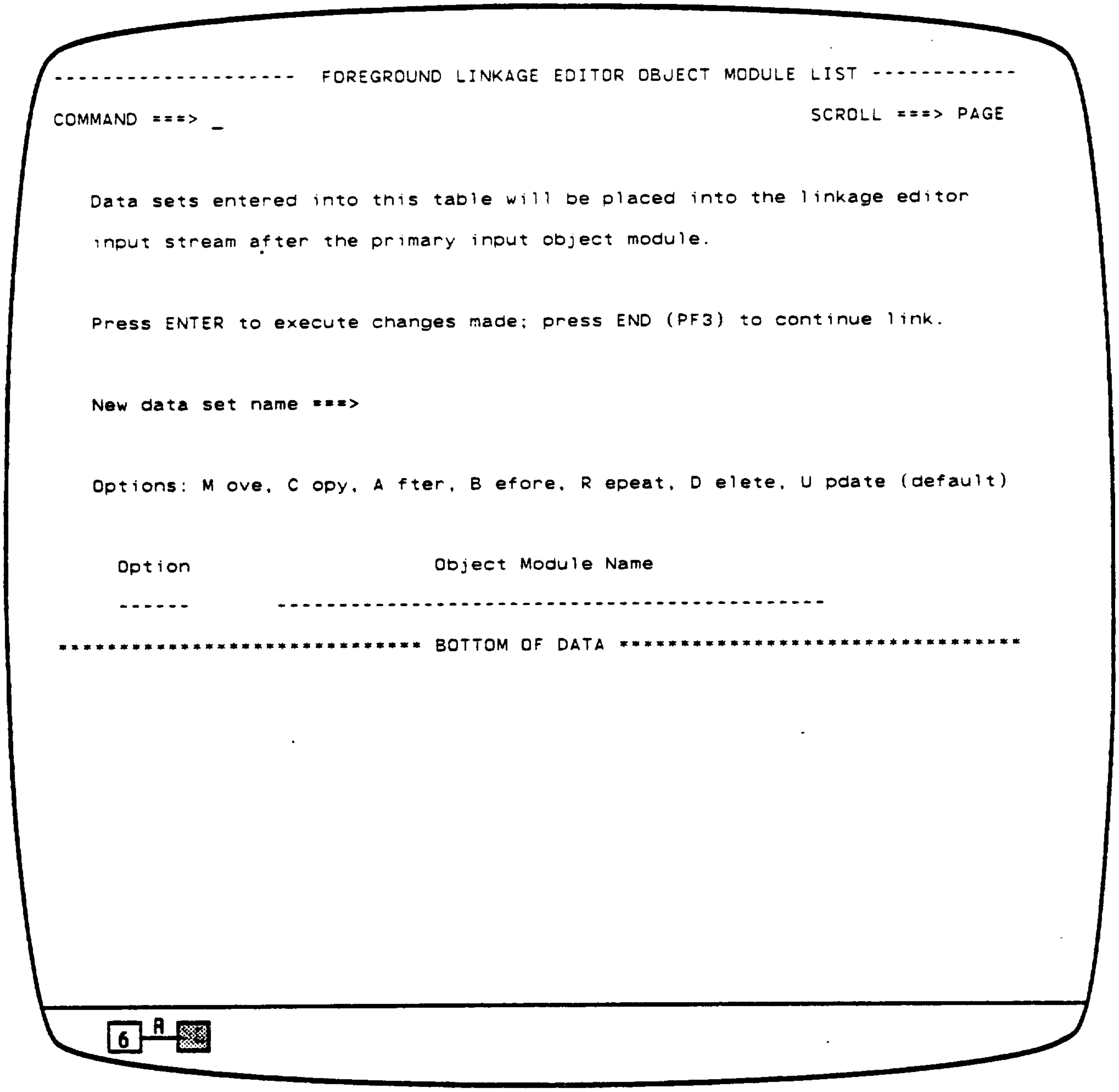


```
------------   FOREGROUND LINKAGE EDIT WITH USER DEFINED LIBRARIES   ------------
COMMAND ===>


LOAD LIBRARY     ===> TUTORIAL.LOAD
   MEMBER        ===> EXAMPLE


INPUT FILE       ===> TS.SAMPLE.OBJ
PANVALET MEMBER ===>
PANVALET PREFIX ===>


LIST ID ===>             Automatic Browse? ===> NO    Automatic Print? ===> NO


LINKAGE EDITOR OPTIONS:
   ===> MAP,LIST,LET


Use Additional Object Modules List?   ===> Y      Display/Change it? ===> Y
Use Additional Object Libraries List? ===>        Display/Change it? ===>
Use Additional Load Libraries List?   ===>        Display/Change it? ===>


Use VS FORTRAN Libraries? ===> Y
Use Pascal/VS Libraries?  ===>
```

<FT> to each position indicated by a shaded area and type in the corresponding parameter. The program will be called EXAMPLE and will be saved in a library called GJZZP.TUTORIAL.LOAD. The list ID for link-edits defaults to <userid>.<member name>.LINKLIST, GJZZP.EXAMPLE.LINKLIST in this case.

Press <enter> when finished. The screen will clear, and soon . . .

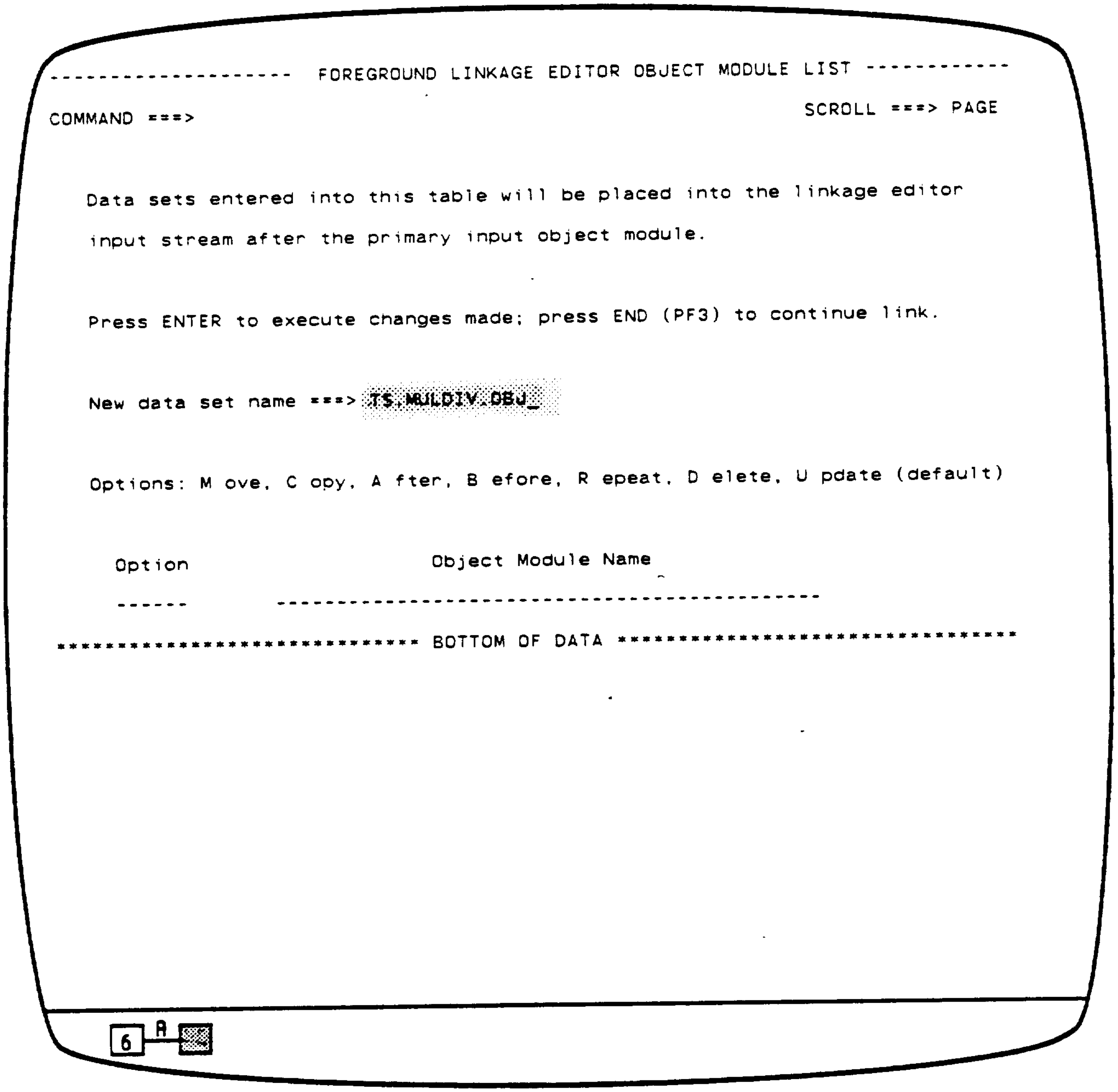
Scenario Step 5: Linking program



```
-------------------- FOREGROUND LINKAGE EDITOR OBJECT MODULE LIST ------------
COMMAND ===> _                                                 SCROLL ===> PAGE

   Data sets entered into this table will be placed into the linkage editor
   input stream after the primary input object module.

   Press ENTER to execute changes made; press END (PF3) to continue link.

   New data set name ===>

   Options: M ove, C opy, A fter, B efore, R epeat, D elete, U pdate (default)

     Option                    Object Module Name
     ------      ---------------------------------------------
****************************** BOTTOM OF DATA ********************************
```

. . . this panel appears. This panel is displayed in response to the Y in the "Display/Change it" question about the Additional Object Modules List. We want to change the list (because there is nothing in it yet).

Scenario Step 5: Linking program

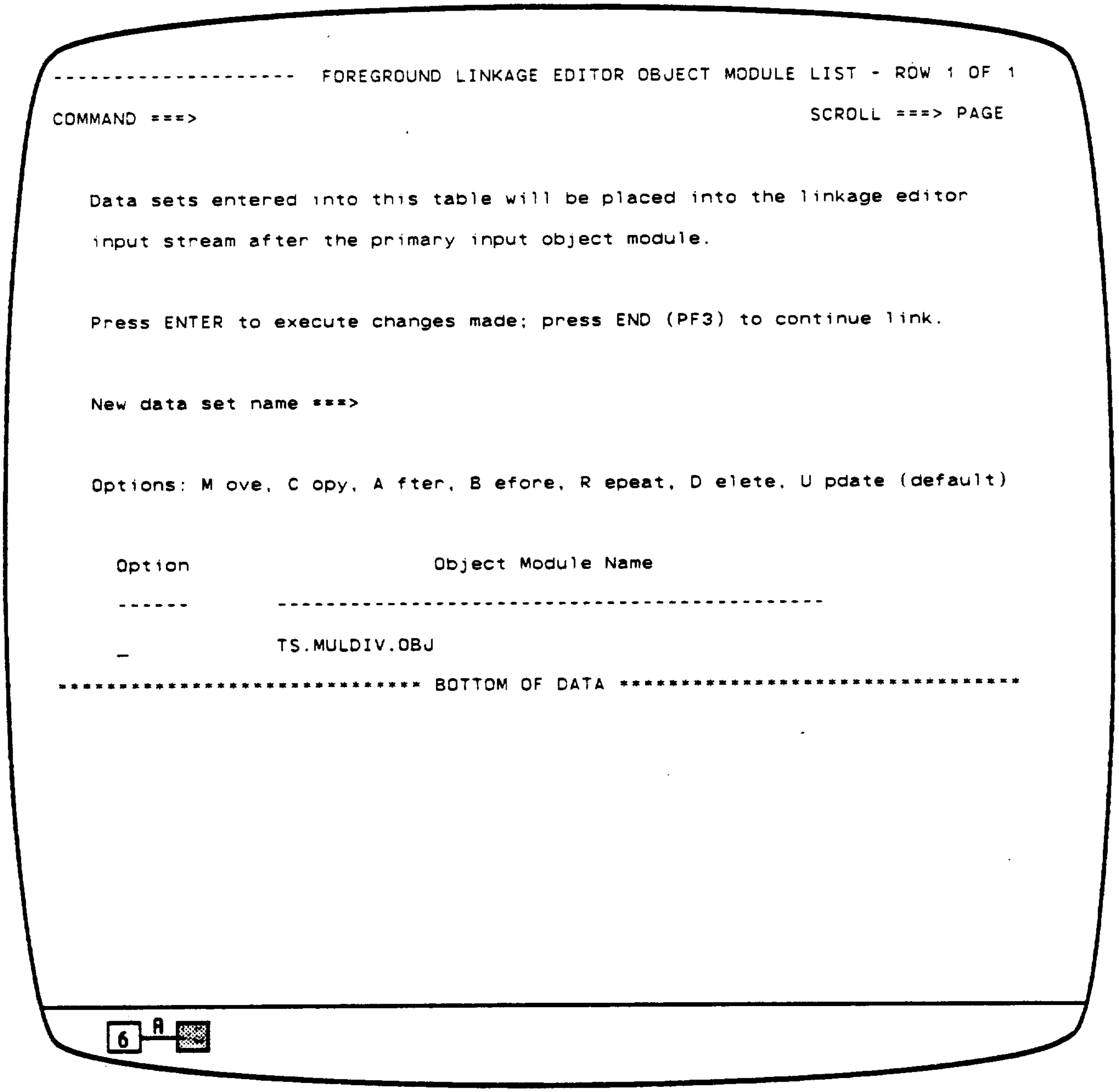


```
-------------------- FOREGROUND LINKAGE EDITOR OBJECT MODULE LIST ------------
COMMAND ===>                                                     SCROLL ===> PAGE

   Data sets entered into this table will be placed into the linkage editor
   input stream after the primary input object module.

   Press ENTER to execute changes made; press END (PF3) to continue link.

   New data set name ===> TS.MULDIV.OBJ_

   Options: M ove, C opy, A fter, B efore, R epeat, D elete, U pdate (default)

     Option                  Object Module Name
     ------      ---------------------------------------------
****************************** BOTTOM OF DATA ********************************
```

<FT> to position the cursor in the NEW DATA SET NAME field of the panel. Type in the name of one of the subroutines and press <enter>.

C-2

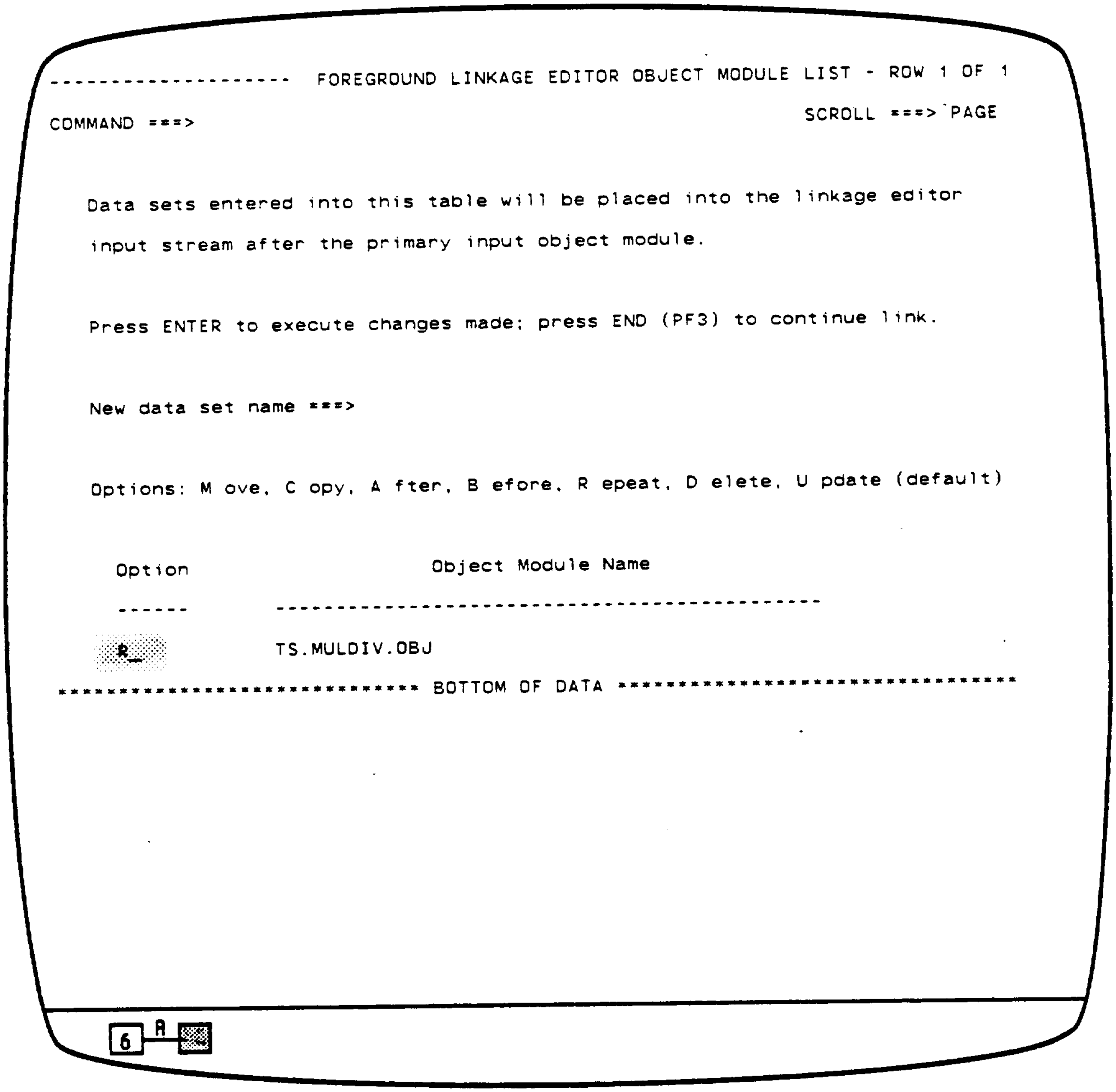
Scenario Step 5: Linking program



```
-------------------- FOREGROUND LINKAGE EDITOR OBJECT MODULE LIST - ROW 1 OF 1
COMMAND ===>                                                  SCROLL ===> PAGE

   Data sets entered into this table will be placed into the linkage editor
   input stream after the primary input object module.

   Press ENTER to execute changes made; press END (PF3) to continue link.

   New data set name ===>

   Options: M ove, C opy, A fter, B efore, R epeat, D elete, U pdate (default)

     Option                    Object Module Name
     ------       --------------------------------------------
     _            TS.MULDIV.OBJ
****************************** BOTTOM OF DATA *********************************
```

The name will be converted to all caps and will be entered into the list. The cursor will be positioned in the option field for that member name.

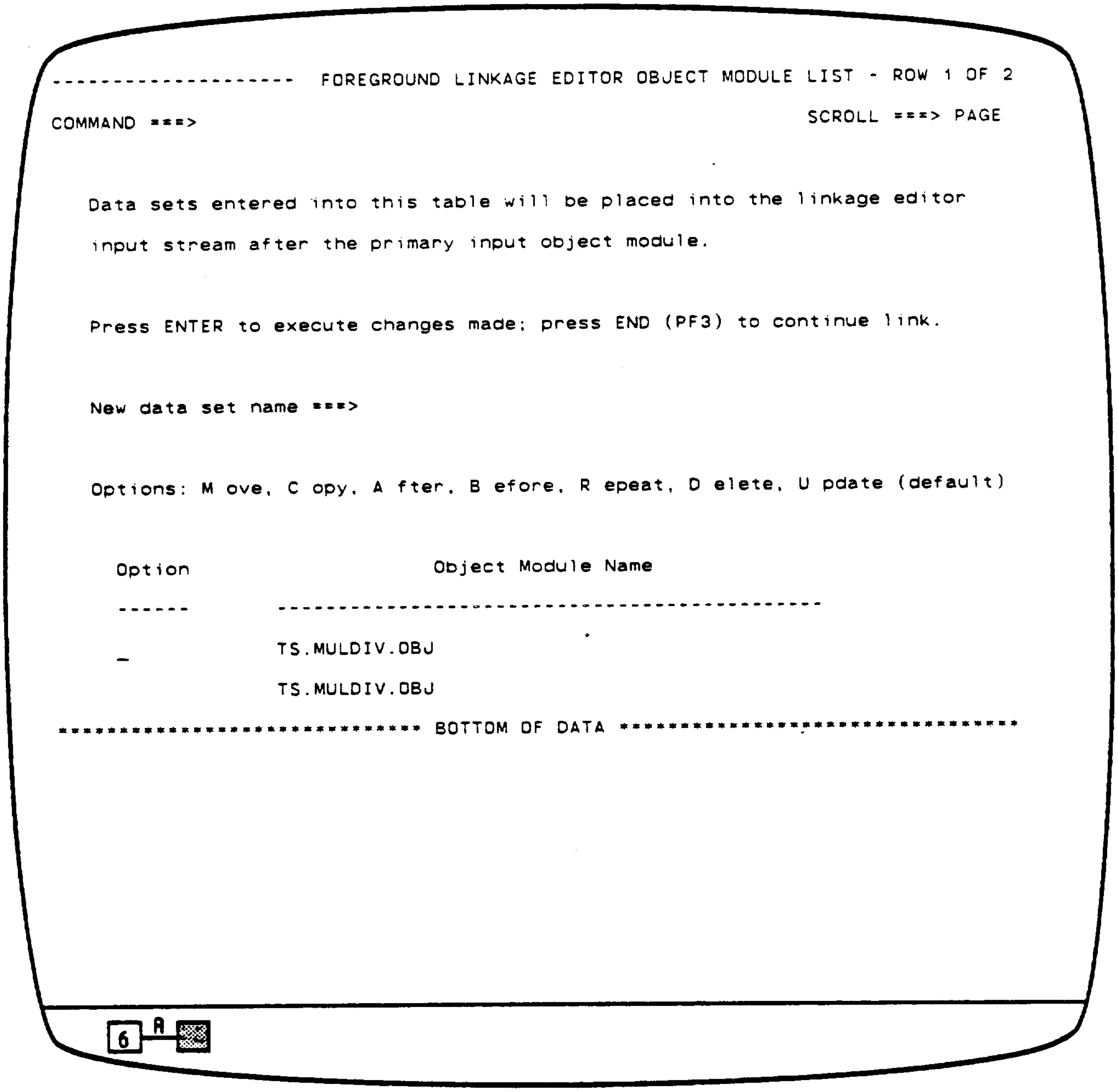
Scenario Step 5: Linking program



```
-------------------- FOREGROUND LINKAGE EDITOR OBJECT MODULE LIST - ROW 1 OF 1
COMMAND ===>                                                  SCROLL ===> PAGE

   Data sets entered into this table will be placed into the linkage editor
   input stream after the primary input object module.

   Press ENTER to execute changes made; press END (PF3) to continue link.

   New data set name ===>

   Options: M ove, C opy, A fter, B efore, R epeat, D elete, U pdate (default)

     Option                   Object Module Name
     ------        -------------------------------------------
     R_            TS.MULDIV.OBJ
***************************** BOTTOM OF DATA ********************************
```

Type an R (Repeat) as the option for this list name, press <enter>, and . . .

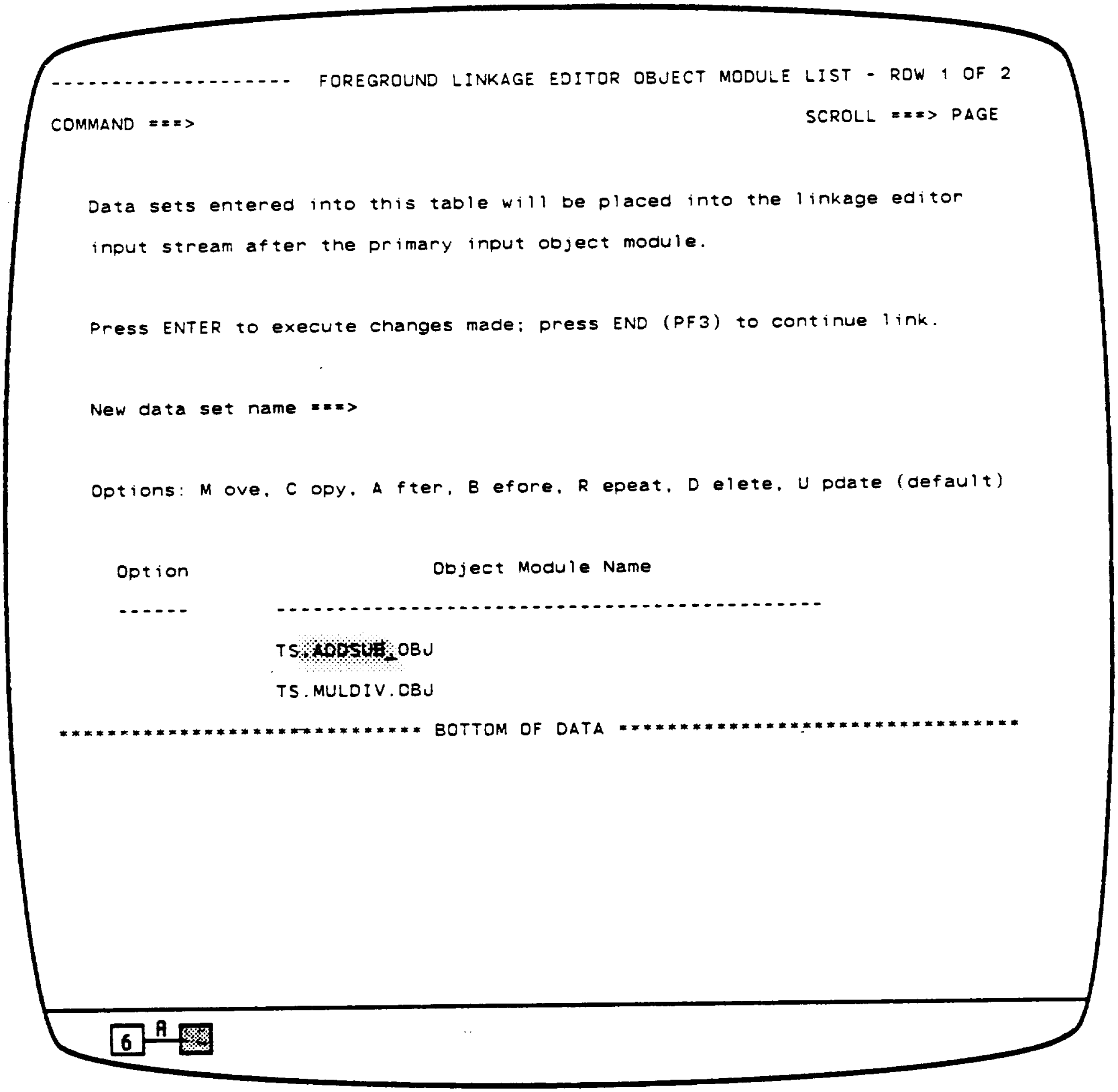
Scenario Step 5: Linking program



```
-------------------- FOREGROUND LINKAGE EDITOR OBJECT MODULE LIST - ROW 1 OF 2
COMMAND ===>                                                  SCROLL ===> PAGE

   Data sets entered into this table will be placed into the linkage editor
   input stream after the primary input object module.

   Press ENTER to execute changes made; press END (PF3) to continue link.

   New data set name ===>

   Options: M ove, C opy, A fter, B efore, R epeat, D elete, U pdate (default)

     Option                     Object Module Name
     ------       ---------------------------------------------
                  TS.MULDIV.OBJ
     _
                  TS.MULDIV.OBJ
****************************** BOTTOM OF DATA ********************************
```

. . . the name will be repeated. Now move the cursor . . .

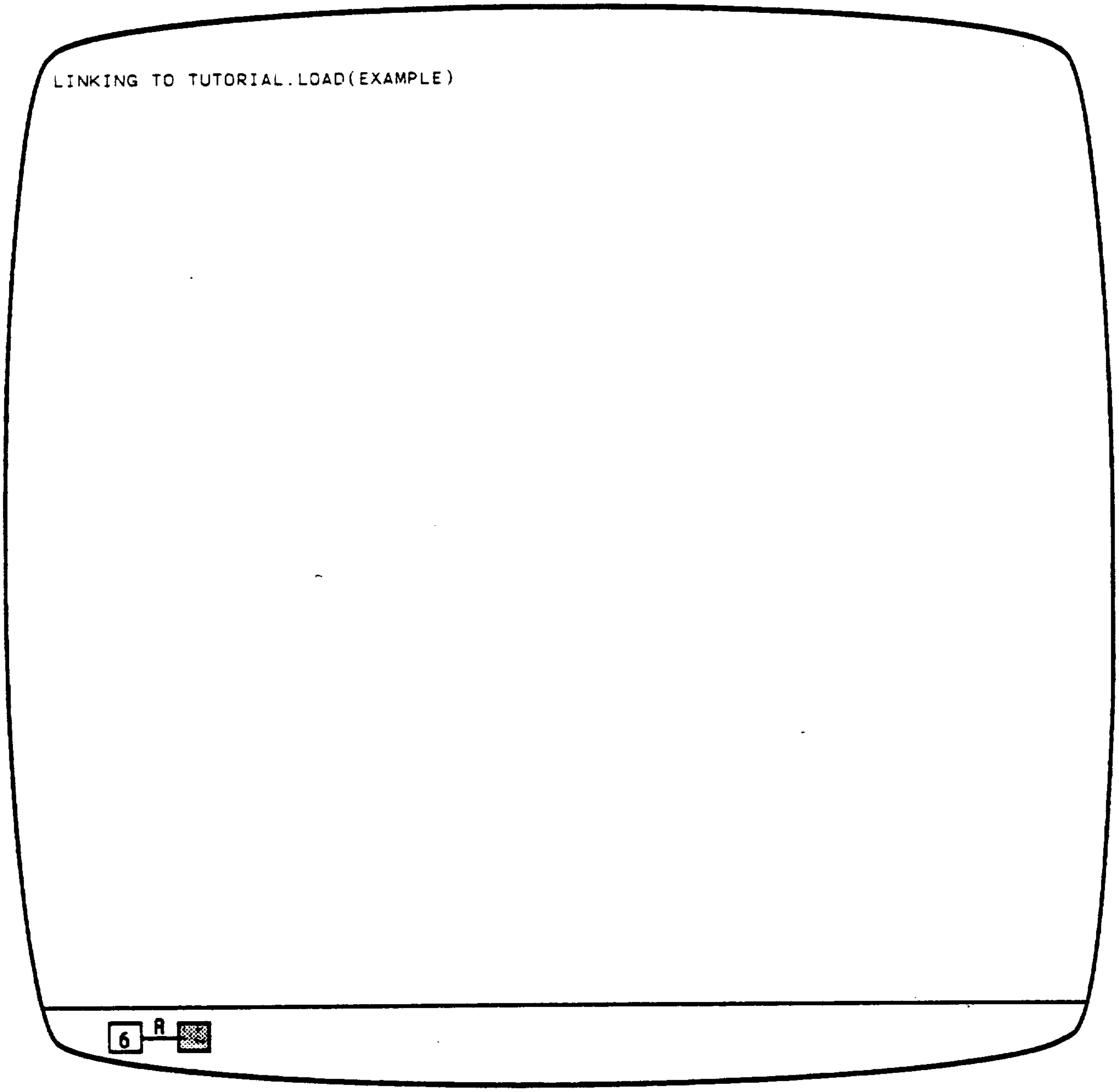
Scenario Step 5: Linking program



```
--------------------  FOREGROUND LINKAGE EDITOR OBJECT MODULE LIST - ROW 1 OF 2
COMMAND ===>                                                     SCROLL ===> PAGE

   Data sets entered into this table will be placed into the linkage editor
   input stream after the primary input object module.

   Press ENTER to execute changes made; press END (PF3) to continue link.

   New data set name ===>

   Options: M ove, C opy, A fter, B efore, R epeat, D elete, U pdate (default)

     Option                   Object Module Name
     ------        ---------------------------------------------
                   TS.ADDSUB.OBJ
                   TS.MULDIV.OBJ
****************************** BOTTOM OF DATA *********************************
```

. . . and type ADDSUB over MULDIV. Press <enter> to update the list, then press <end> to continue the link.

Scenario Step 5: Linking program

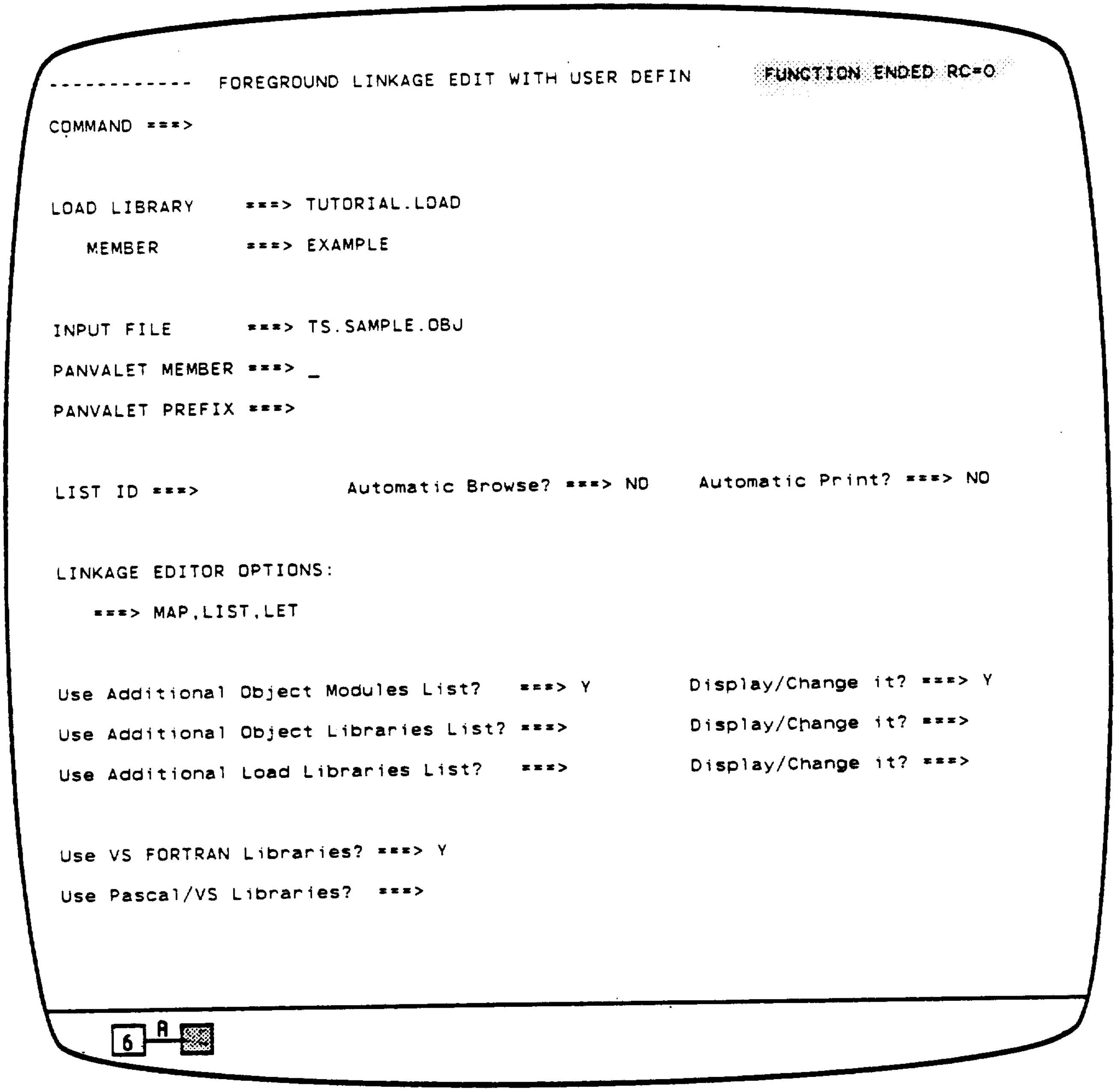


```
LINKING TO TUTORIAL.LOAD(EXAMPLE)
```

The screen will clear and soon this message will be displayed.

In a short time . . .

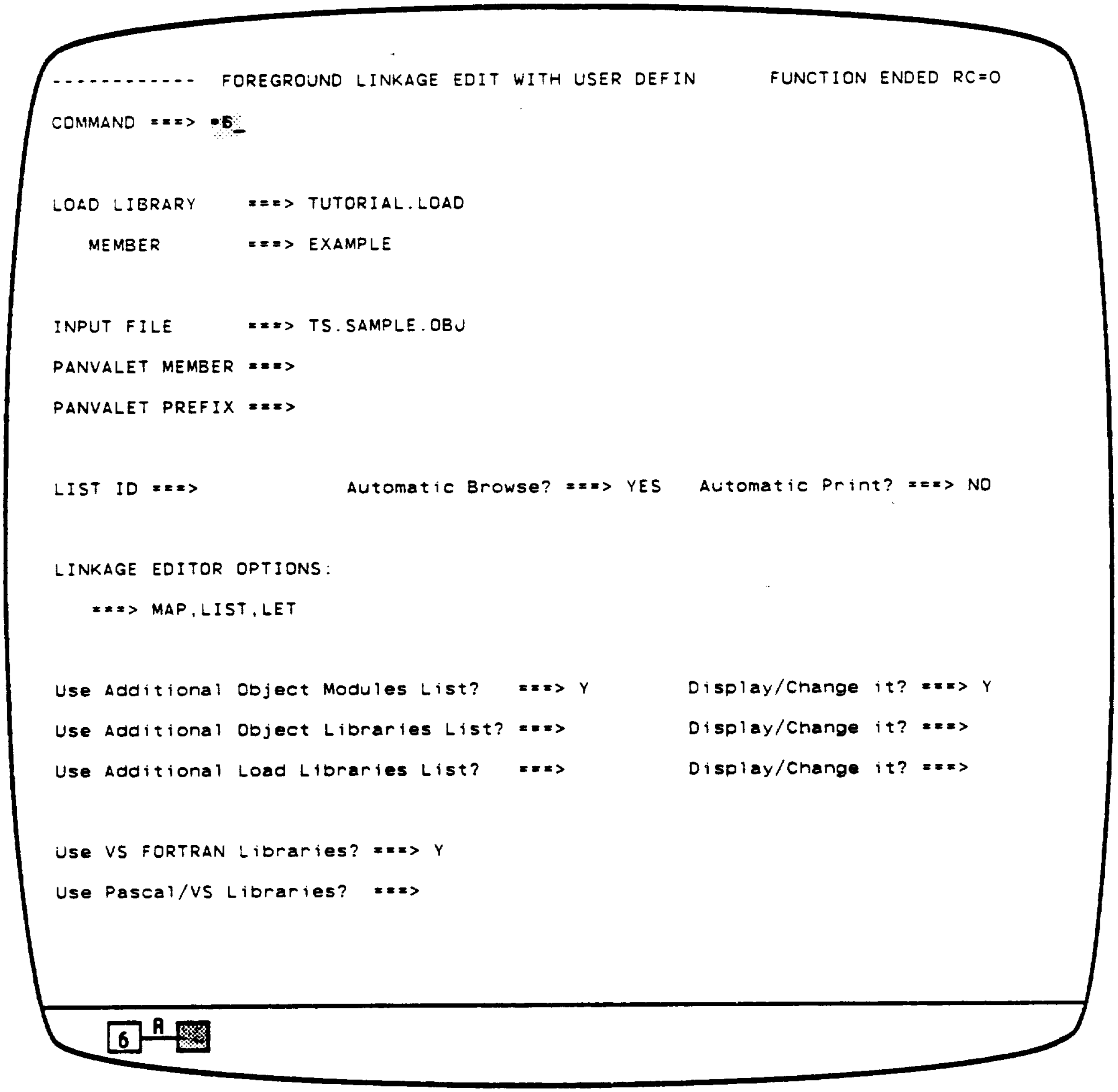
Scenario Step 5: Linking program



```
------------  FOREGROUND LINKAGE EDIT WITH USER DEFIN       FUNCTION ENDED RC=0
COMMAND ===>


LOAD LIBRARY     ===> TUTORIAL.LOAD
   MEMBER        ===> EXAMPLE


INPUT FILE       ===> TS.SAMPLE.OBJ
PANVALET MEMBER ===> _
PANVALET PREFIX ===>


LIST ID ===>             Automatic Browse? ===> NO    Automatic Print? ===> NO


LINKAGE EDITOR OPTIONS:
   ===> MAP,LIST,LET


Use Additional Object Modules List?   ===> Y       Display/Change it? ===> Y
Use Additional Object Libraries List? ===>         Display/Change it? ===>
Use Additional Load Libraries List?   ===>         Display/Change it? ===>


Use VS FORTRAN Libraries? ===> Y
Use Pascal/VS Libraries?  ===>
```

. . . the link edit parameter entry panel will be redisplayed. This function should end with a return code of 0, too. We're batting 1000.

Scenario Step 5: Linking program

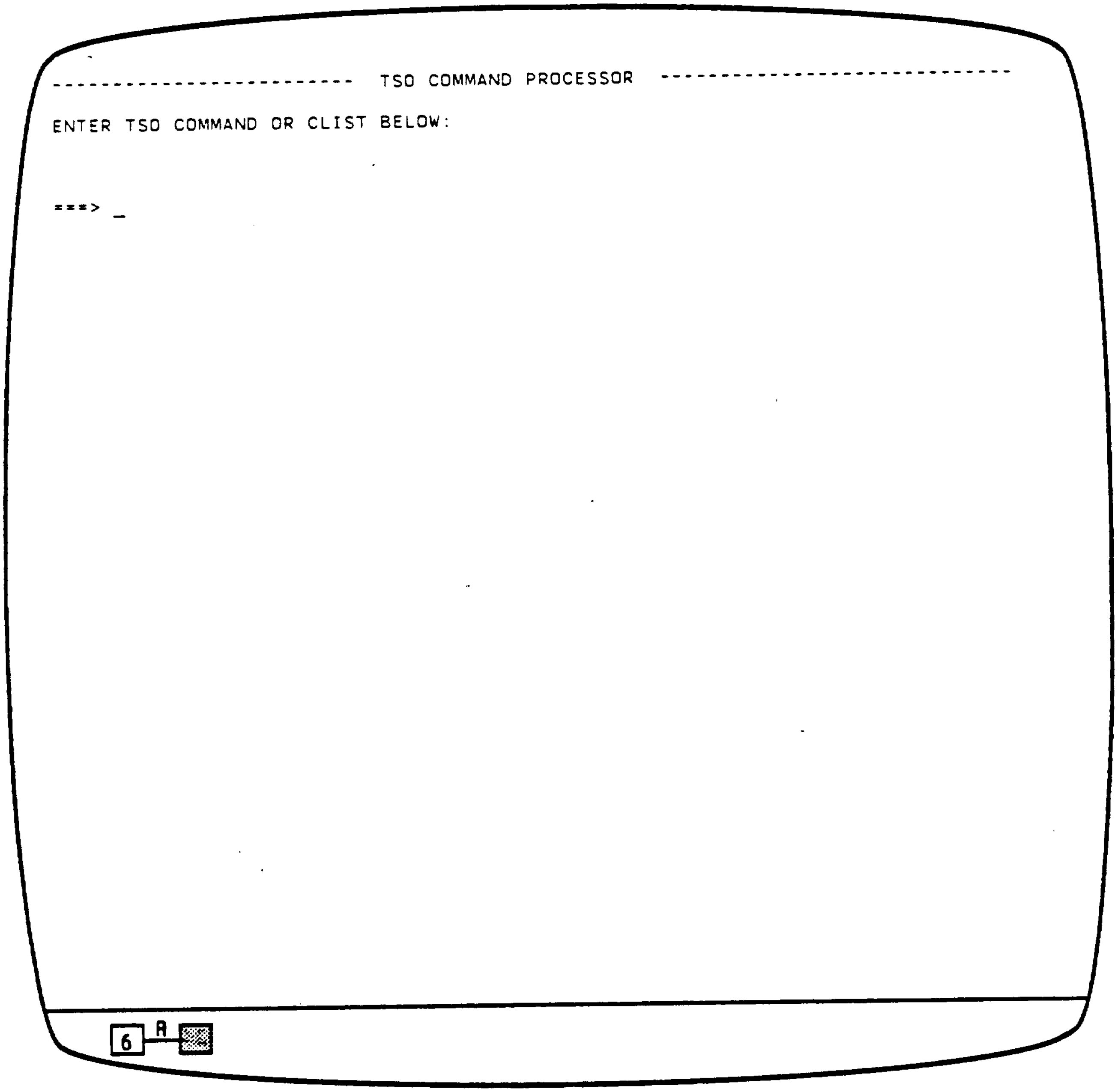


```
------------  FOREGROUND LINKAGE EDIT WITH USER DEFIN      FUNCTION ENDED RC=0
COMMAND ===> =6_


LOAD LIBRARY    ===> TUTORIAL.LOAD

   MEMBER       ===> EXAMPLE


INPUT FILE      ===> TS.SAMPLE.OBJ

PANVALET MEMBER ===>

PANVALET PREFIX ===>


LIST ID ===>            Automatic Browse? ===> YES   Automatic Print? ===> NO


LINKAGE EDITOR OPTIONS:

   ===> MAP,LIST,LET


Use Additional Object Modules List?   ===> Y        Display/Change it? ===> Y

Use Additional Object Libraries List? ===>          Display/Change it? ===>

Use Additional Load Libraries List?   ===>          Display/Change it? ===>


Use VS FORTRAN Libraries? ===> Y

Use Pascal/VS Libraries?  ===>
```

This completes step 5 of the scenario. Now let's jump to the TSO function to run our program.

Use the <home> key, to put the cursor in the command field. Type in =6 and press <enter>.

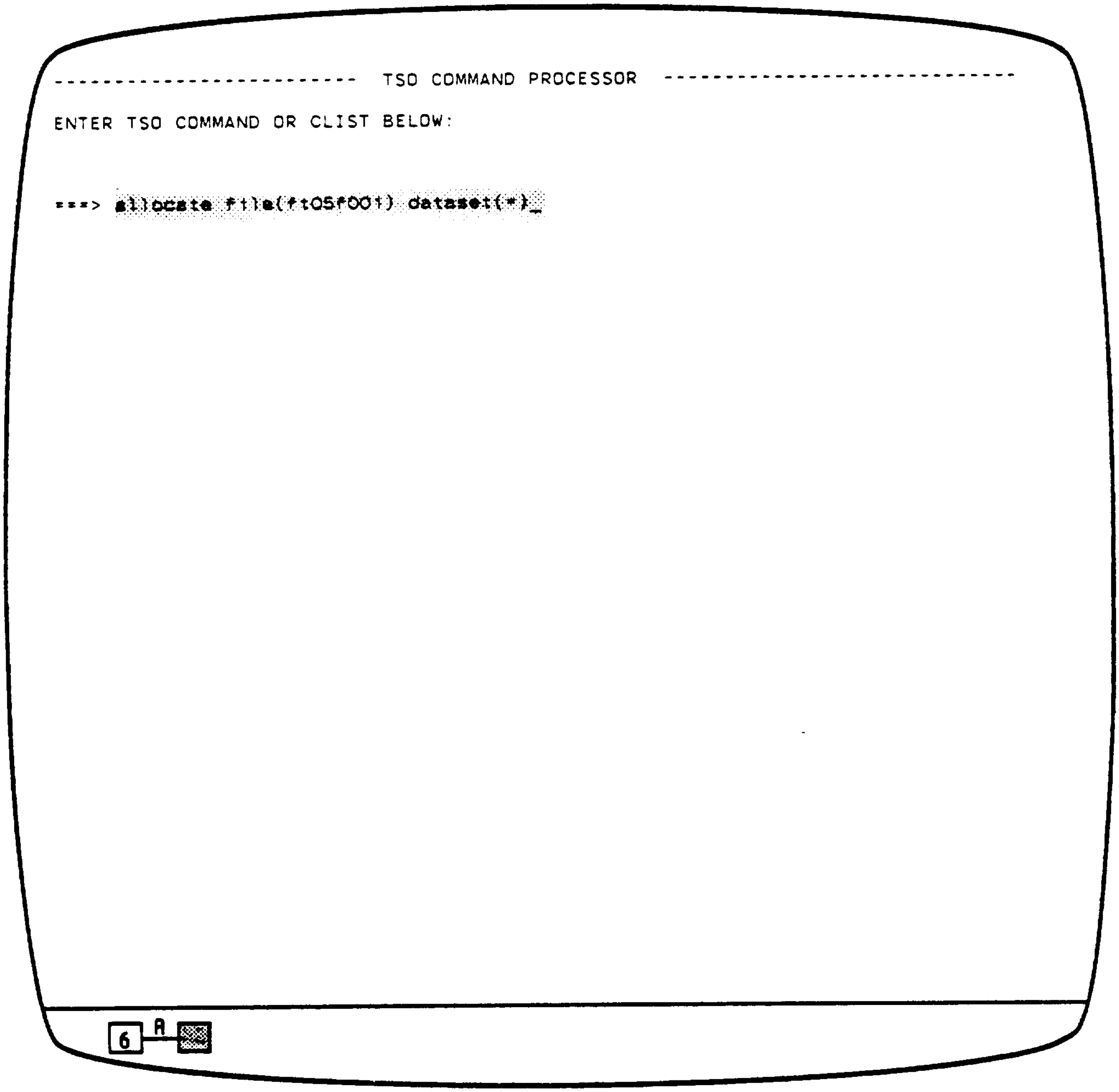
Scenario Step 6: Running program



```
------------------------- TSO COMMAND PROCESSOR -----------------------------
ENTER TSO COMMAND OR CLIST BELOW:


===> _
```

Here we are in the TSO command processing function of SDE. We need to allocate logical unit 5 for input, logical unit 6 for output, and then fire up the program that we have just entered, compiled, and linked.

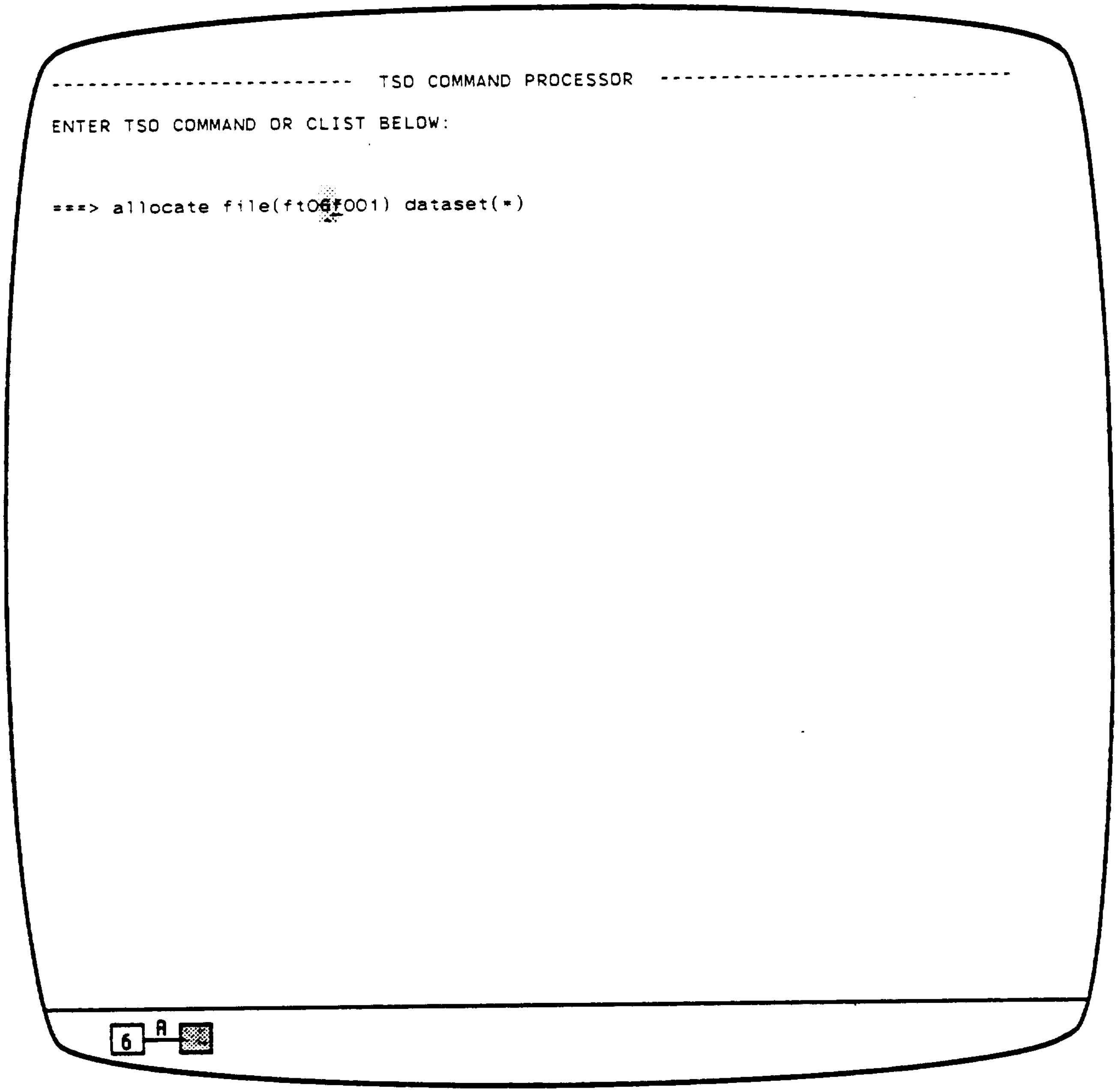
Scenario Step 6: Running program



```
------------------------ TSO COMMAND PROCESSOR ----------------------------
ENTER TSO COMMAND OR CLIST BELOW:


===> allocate file(ft05f001) dataset(*)_
```

To allocate unit 5 to the keyboard for input, we type in this command and press <enter>.

Scenario Step 6: Running program

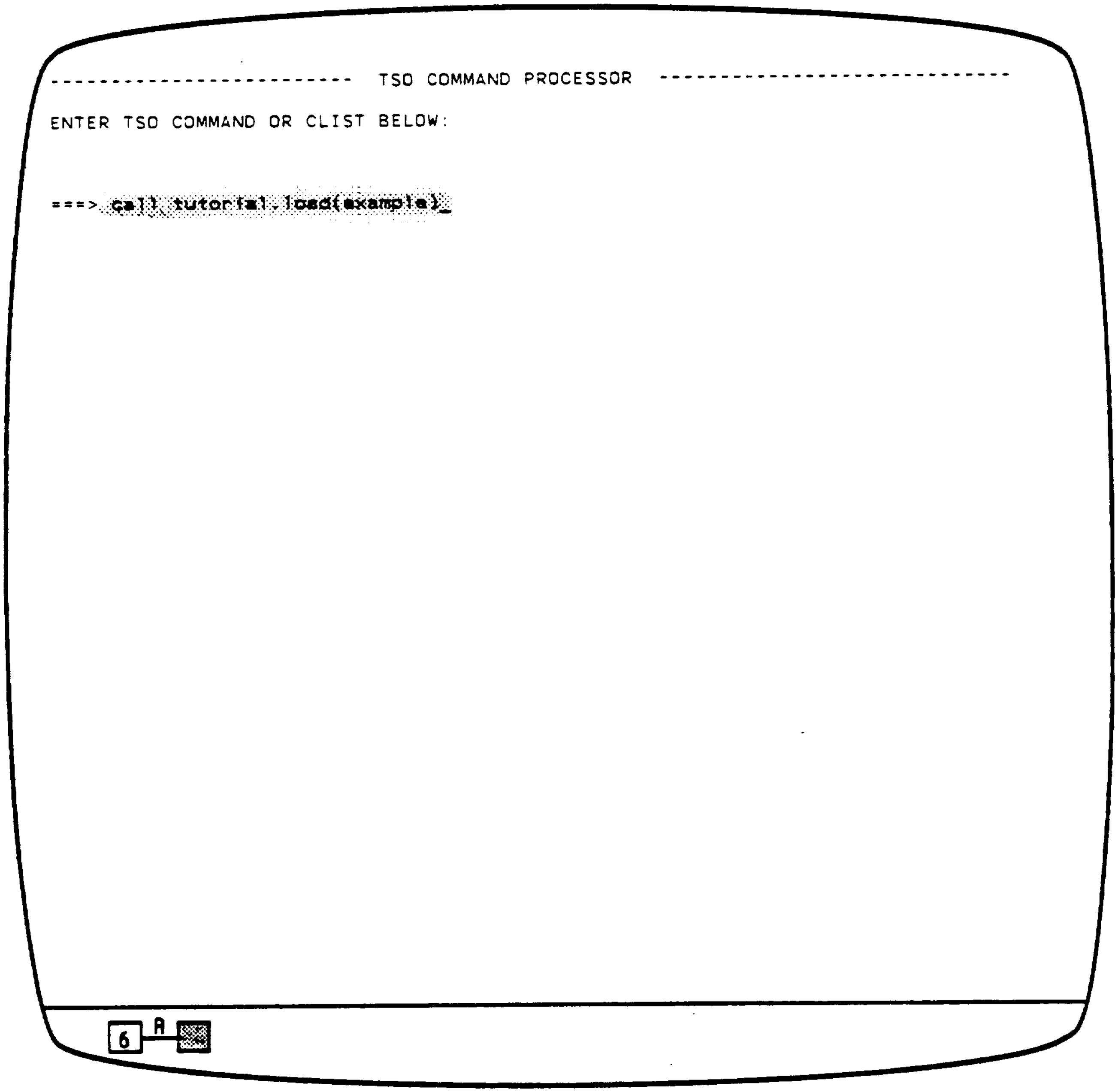


```
------------------------ TSO COMMAND PROCESSOR -----------------------------
ENTER TSO COMMAND OR CLIST BELOW:

===> allocate file(ft06f001) dataset(*)
```

To allocate unit 6 to the display tube for output, move the cursor; change the 5 to a 6; and press <enter>.

FT05F001 is the IBM default for logical unit 5, FT06f001 is the IBM default for logical unit 6.

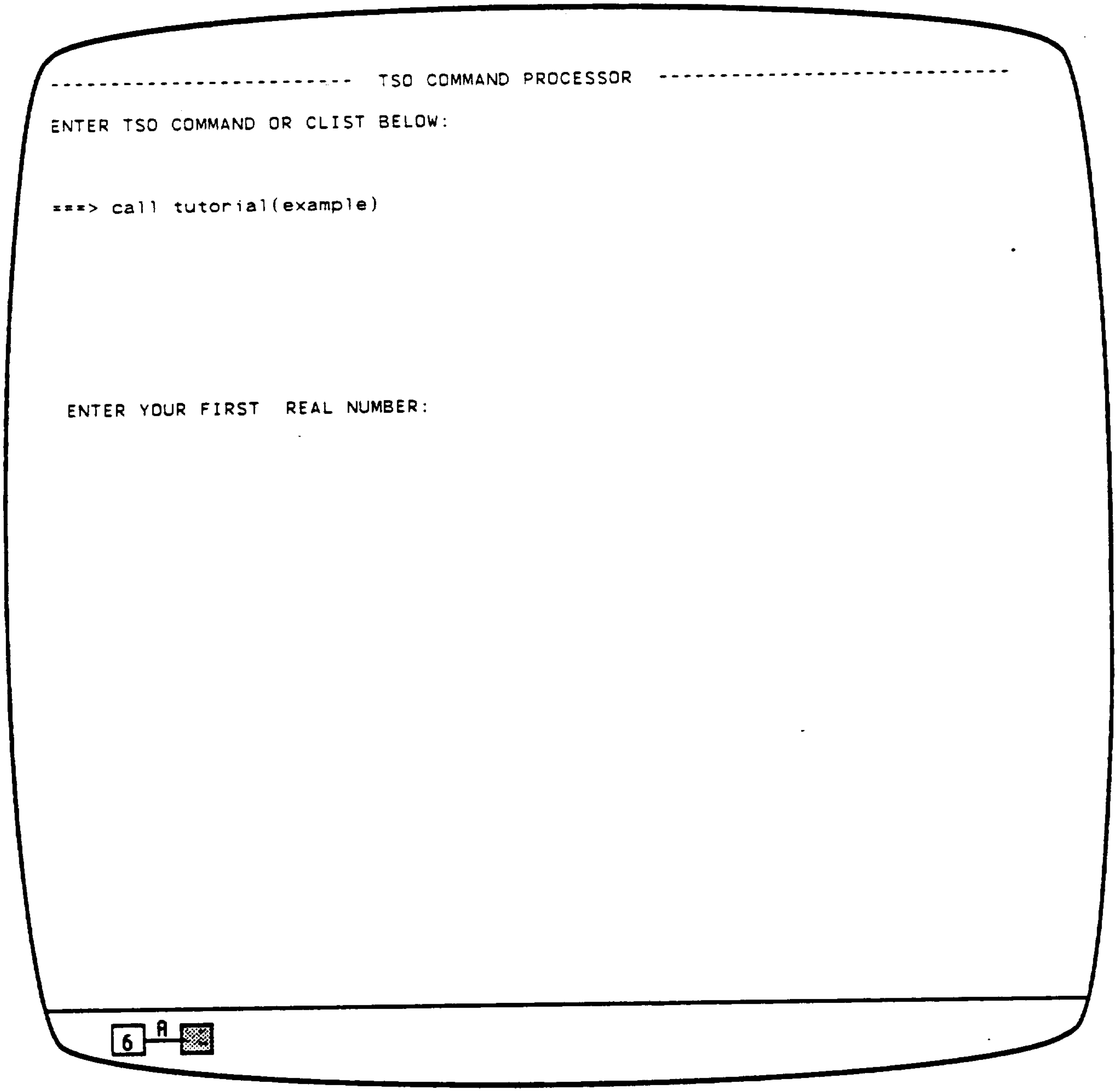
Scenario Step 6: Running program



```
------------------------ TSO COMMAND PROCESSOR ----------------------------
ENTER TSO COMMAND OR CLIST BELOW:


===> call tutorial.load(example)_
```

To execute our program in the foreground, we type in a call to the library and member, and press <enter>.

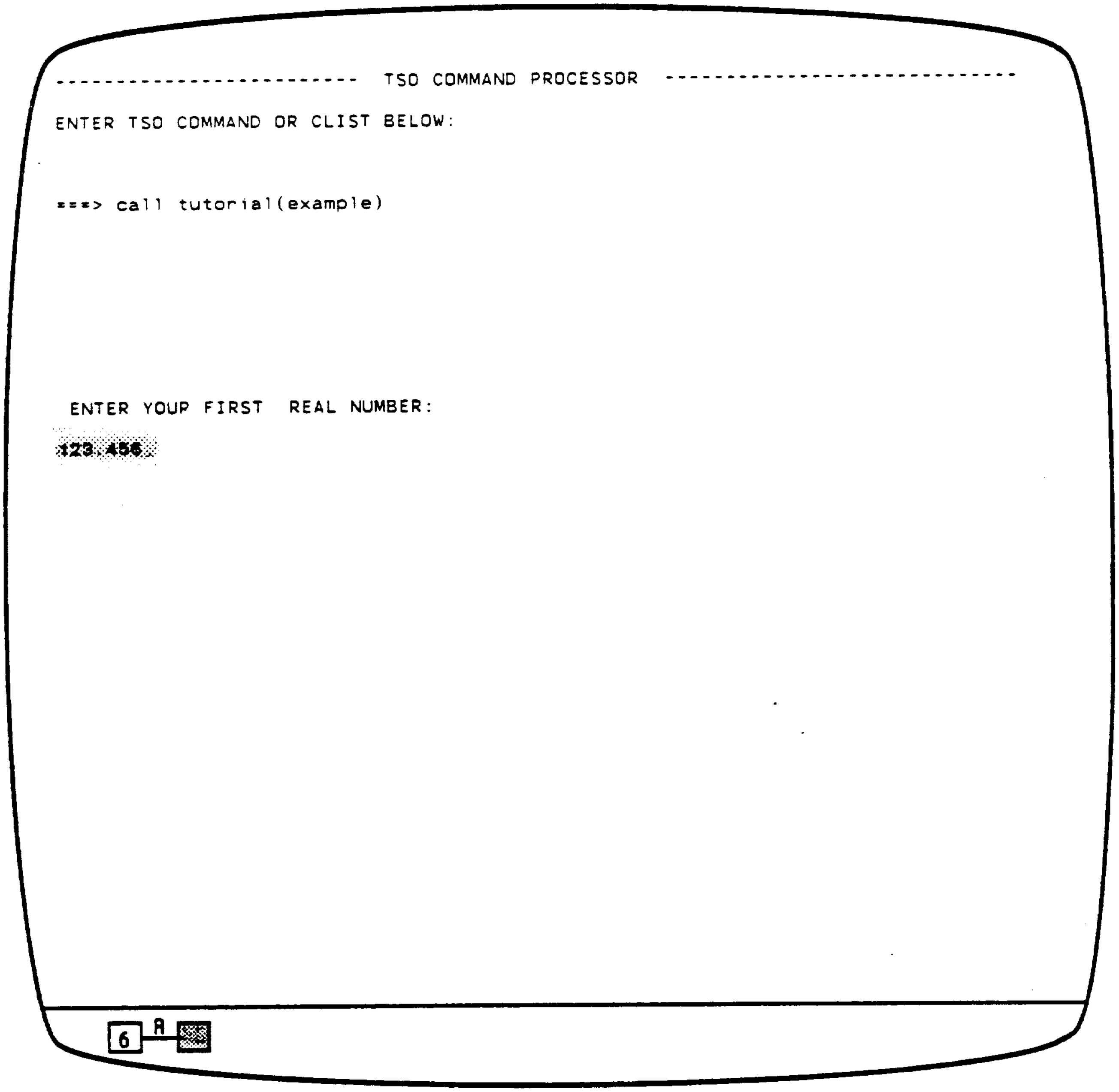
Scenario Step 6: Running program



```
------------------------- TSO COMMAND PROCESSOR -----------------------------
ENTER TSO COMMAND OR CLIST BELOW:


===> call tutorial(example)






 ENTER YOUR FIRST  REAL NUMBER:
```

Our program will prompt us for a number . . .

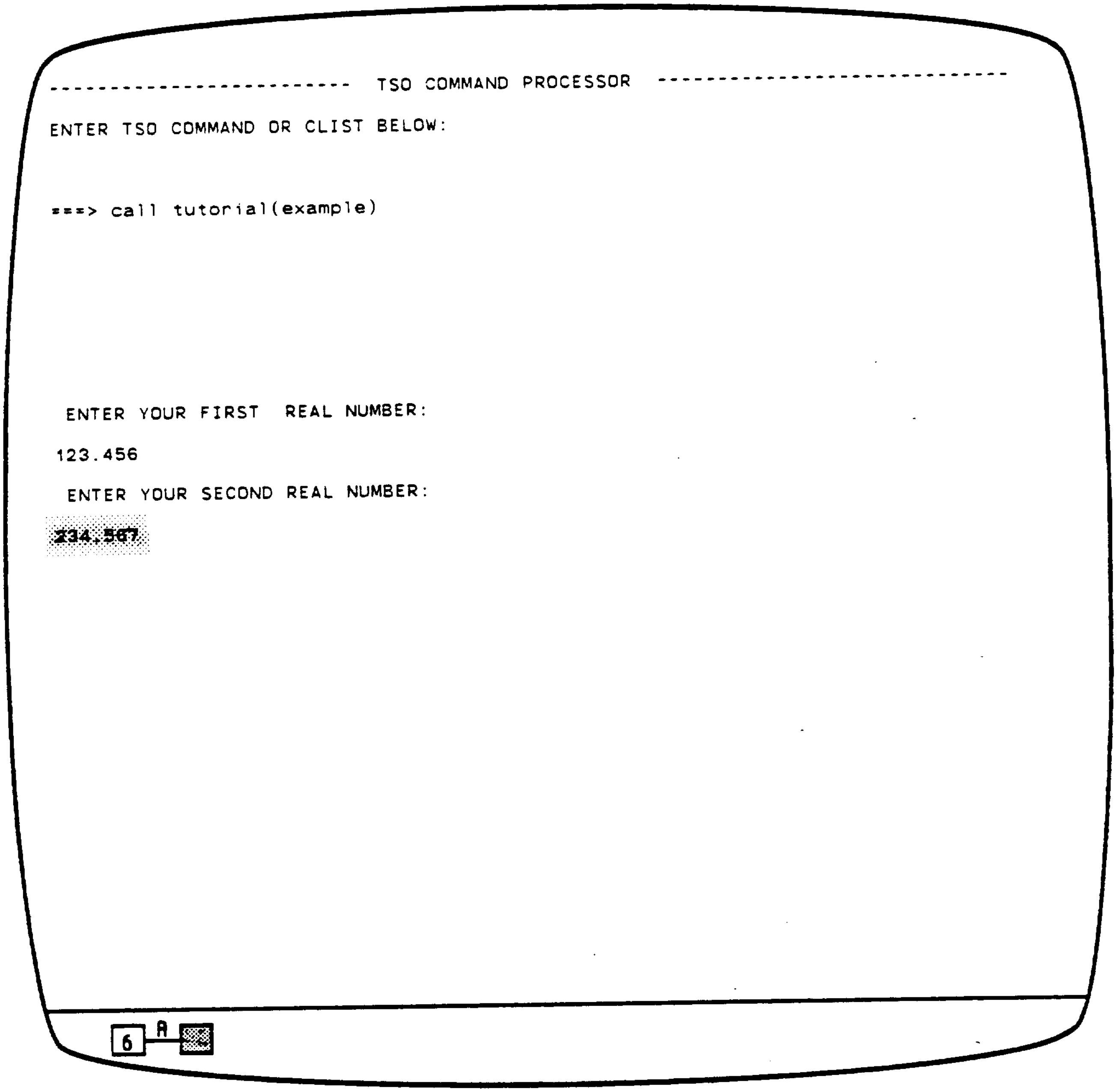
Scenario Step 6: Running program



```
------------------------ TSO COMMAND PROCESSOR ----------------------------
ENTER TSO COMMAND OR CLIST BELOW:


===> call tutorial(example)






 ENTER YOUP FIRST  REAL NUMBER:
123.456
```

We type it in. Then our program will prompt us for another number . . .

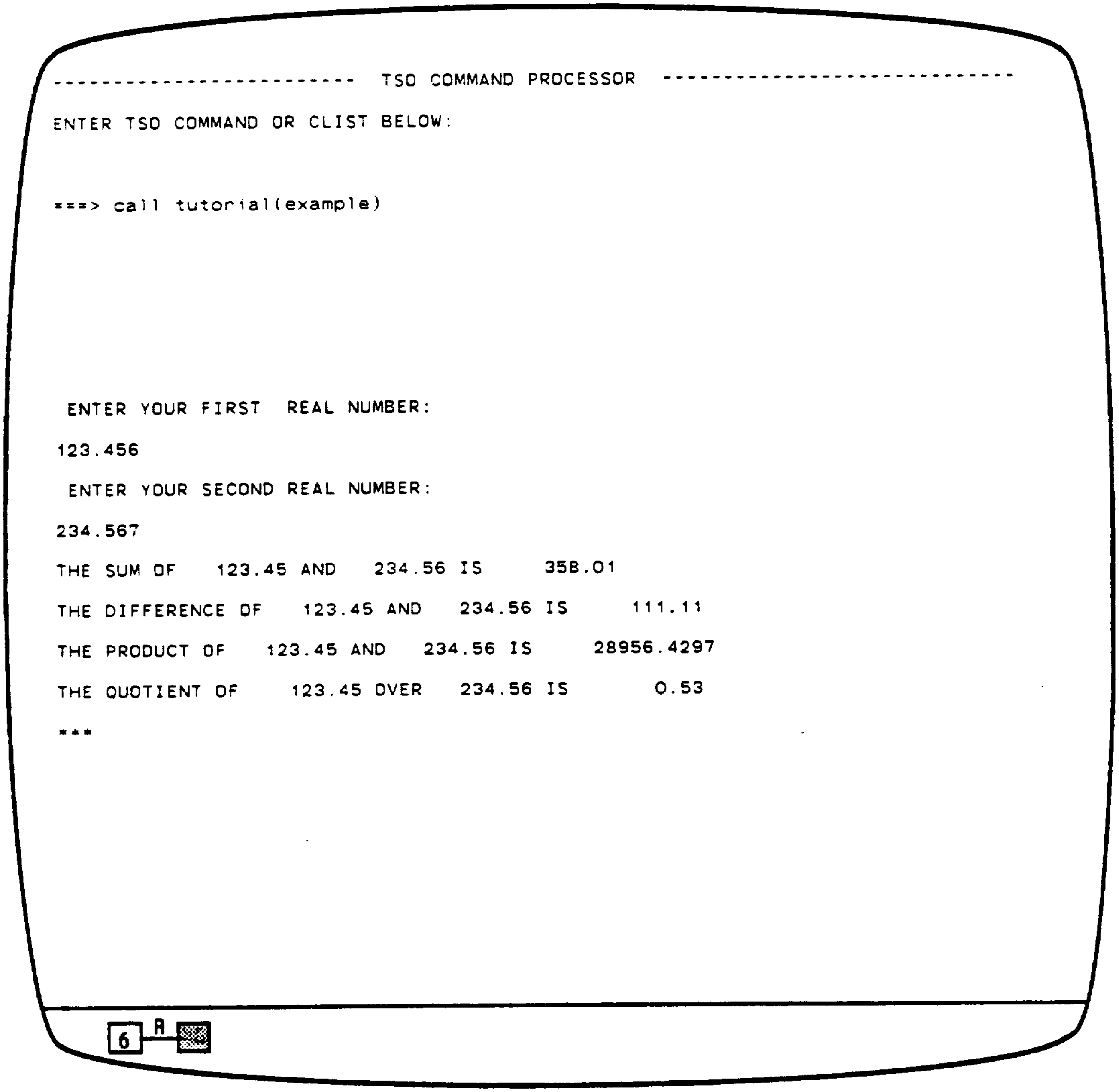
Scenario Step 6: Running program



```
------------------------- TSO COMMAND PROCESSOR -----------------------------
ENTER TSO COMMAND OR CLIST BELOW:


===> call tutorial(example)






 ENTER YOUR FIRST  REAL NUMBER:
123.456
 ENTER YOUR SECOND REAL NUMBER:
234.567
```

Type it in too. After we press <enter> this time . . .

Scenario Step 6: Running program



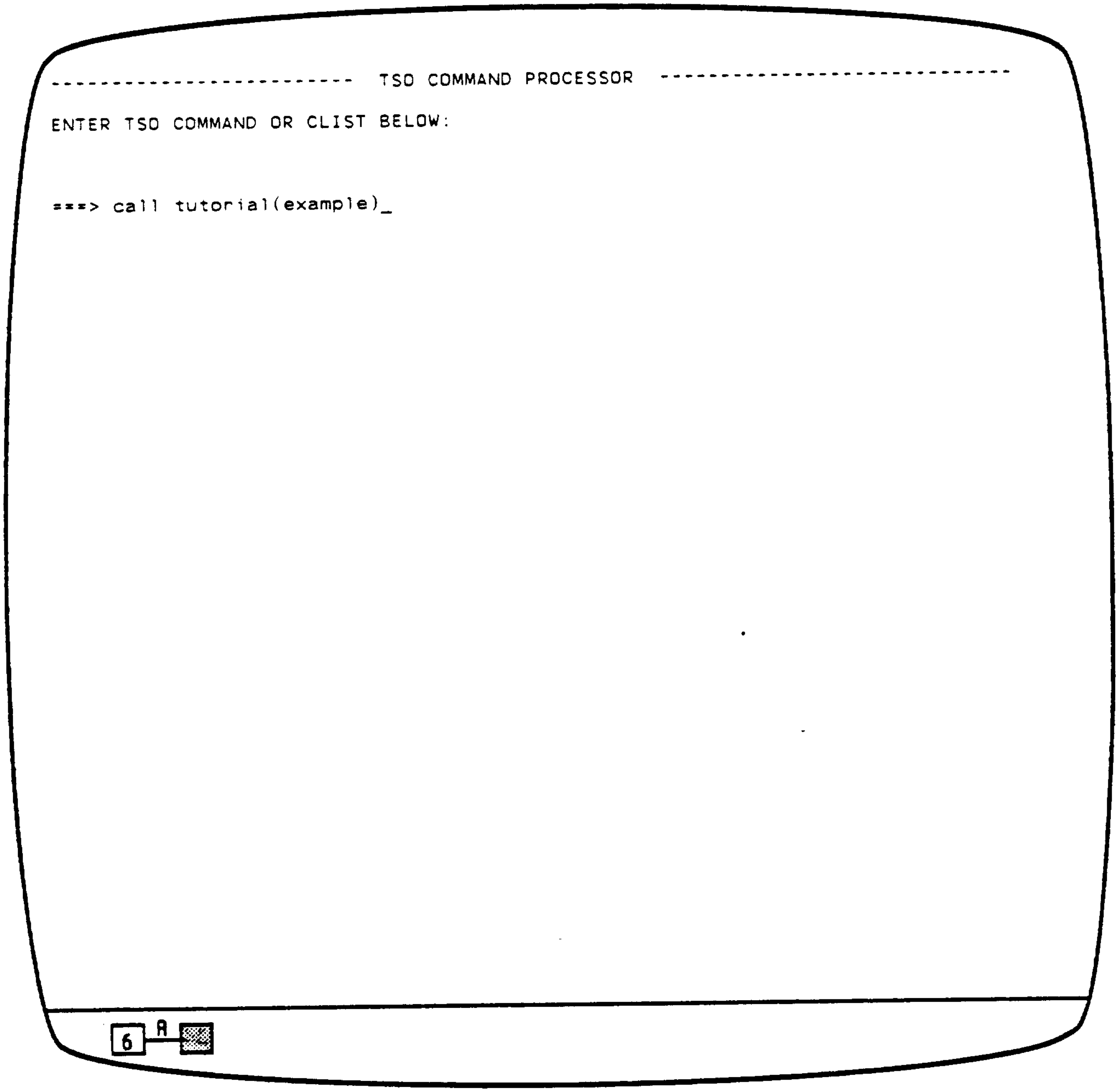
```
------------------------- TSO COMMAND PROCESSOR ----------------------------
ENTER TSO COMMAND OR CLIST BELOW:


===> call tutorial(example)






 ENTER YOUR FIRST  REAL NUMBER:
123.456
 ENTER YOUR SECOND REAL NUMBER:
234.567
THE SUM OF    123.45 AND    234.56 IS      358.01
THE DIFFERENCE OF    123.45 AND    234.56 IS      111.11
THE PRODUCT OF    123.45 AND    234.56 IS      28956.4297
THE QUOTIENT OF     123.45 OVER    234.56 IS        0.53
***
```

. . . Our program will display the results.

Three asterisks are displayed until we acknowledge having seen the results.

Press <enter> to acknowledge, and . . .

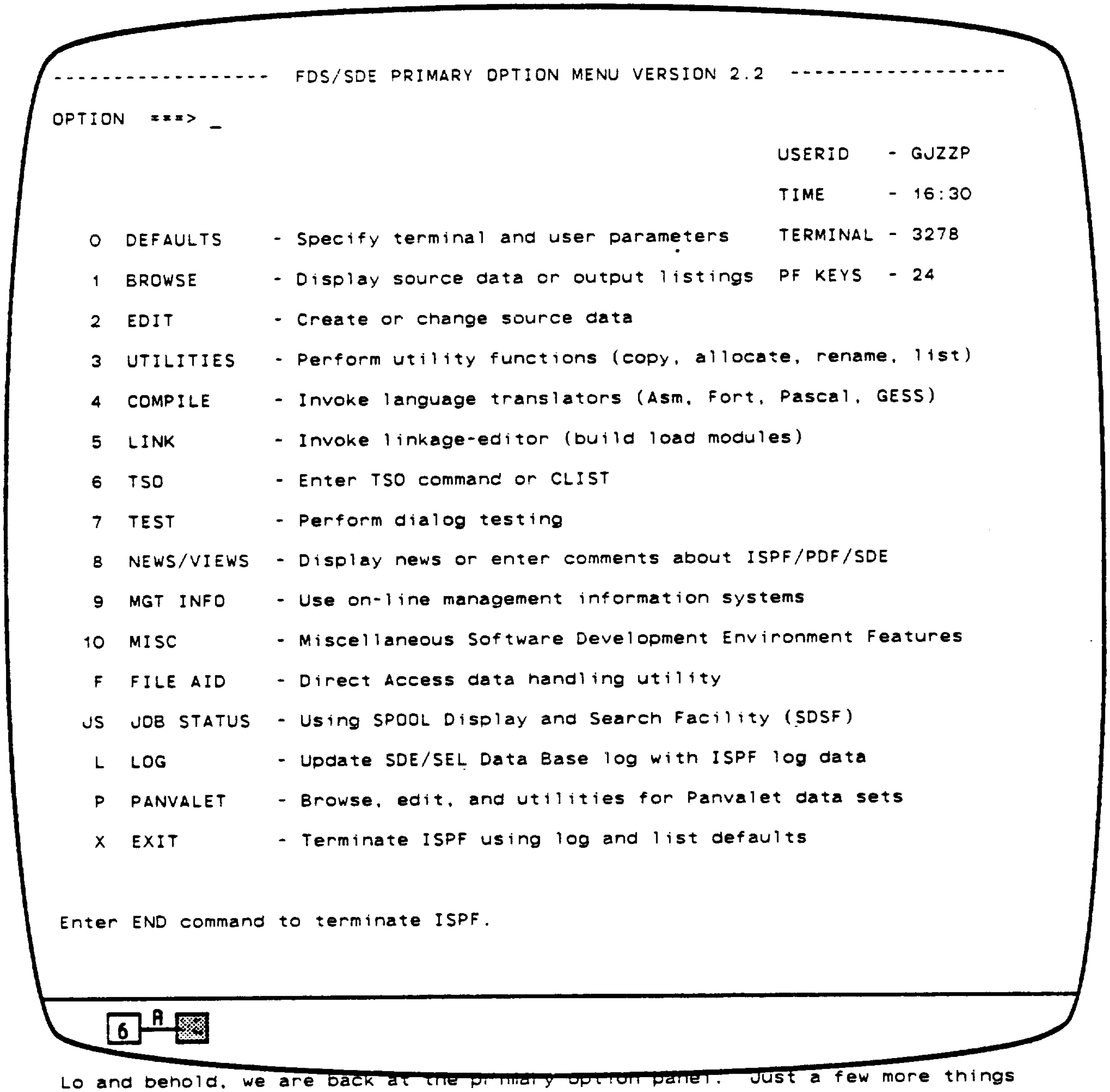
Scenario Step 6: Running program



```
------------------------ TSO COMMAND PROCESSOR -----------------------------
ENTER TSO COMMAND OR CLIST BELOW:


===> call tutorial(example)_
```

the TSO command panel is redisplayed. Press <end>, and . . .

Scenario Step 7: Printing listing files

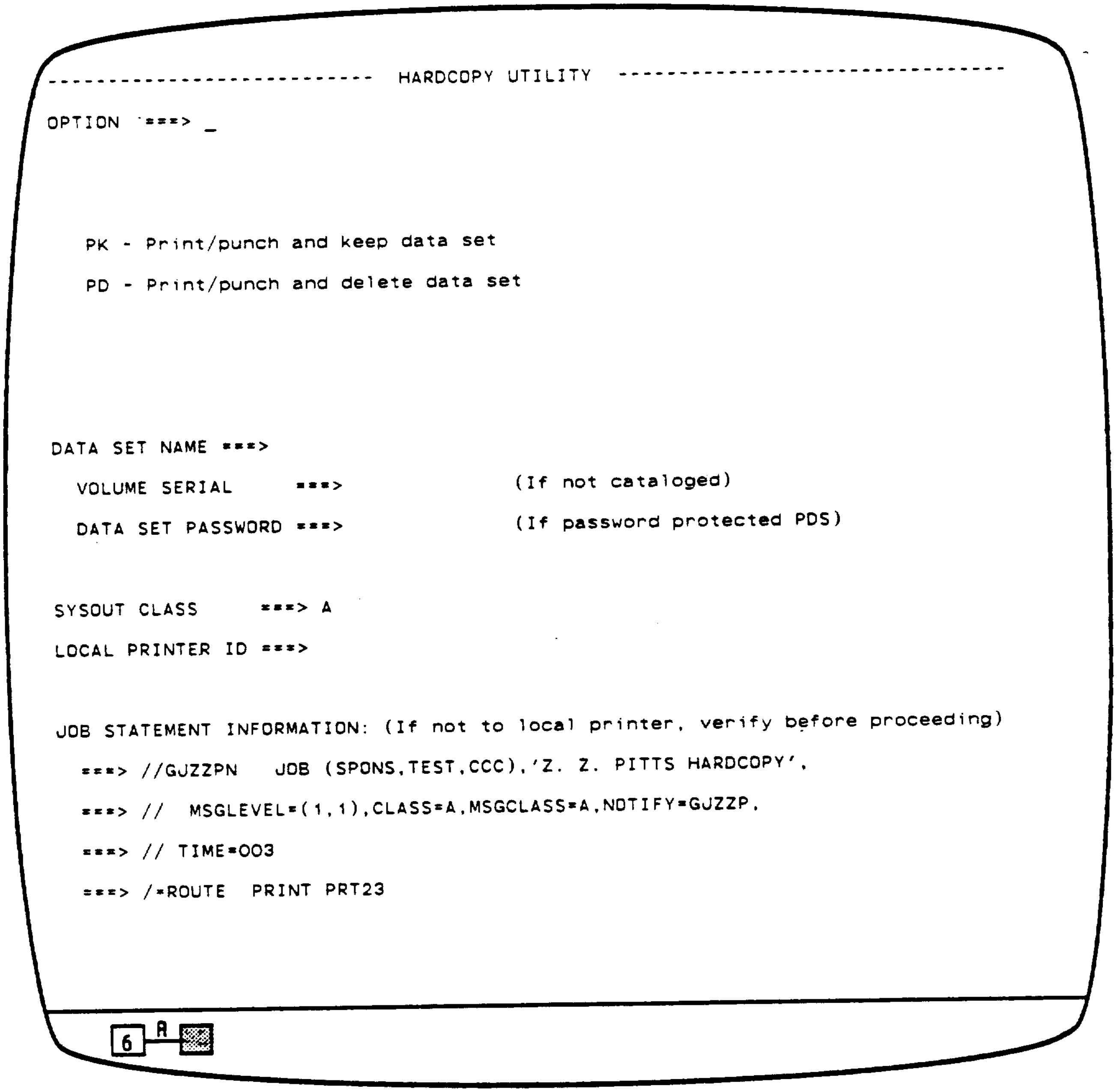


```
------------------  FDS/SDE PRIMARY OPTION MENU VERSION 2.2  -----------------
OPTION  ===> _
                                                               USERID   - GJZZP
                                                               TIME     - 16:30
   0  DEFAULTS     - Specify terminal and user parameters      TERMINAL - 3278
   1  BROWSE       - Display source data or output listings    PF KEYS  - 24
   2  EDIT         - Create or change source data
   3  UTILITIES    - Perform utility functions (copy, allocate, rename, list)
   4  COMPILE      - Invoke language translators (Asm, Fort, Pascal, GESS)
   5  LINK         - Invoke linkage-editor (build load modules)
   6  TSO          - Enter TSO command or CLIST
   7  TEST         - Perform dialog testing
   8  NEWS/VIEWS   - Display news or enter comments about ISPF/PDF/SDE
   9  MGT INFO     - Use on-line management information systems
  10  MISC         - Miscellaneous Software Development Environment Features
   F  FILE AID     - Direct Access data handling utility
  JS  JOB STATUS   - Using SPOOL Display and Search Facility (SDSF)
   L  LOG          - Update SDE/SEL Data Base log with ISPF log data
   P  PANVALET     - Browse, edit, and utilities for Panvalet data sets
   X  EXIT         - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

Lo and behold, we are back at the primary option panel. Just a few more things to do and we can quit.

As you can see, Utility functions are option 3. Take it on faith that listing utilities are utility option 4; and that hardcopy listings are produced as listing option 2. Type =3.4.2 in the option input area. This will jump option 3 (utilities), suboption 4 (listings), sub-suboption 2 (hardcopy). Press <enter>, and . . .

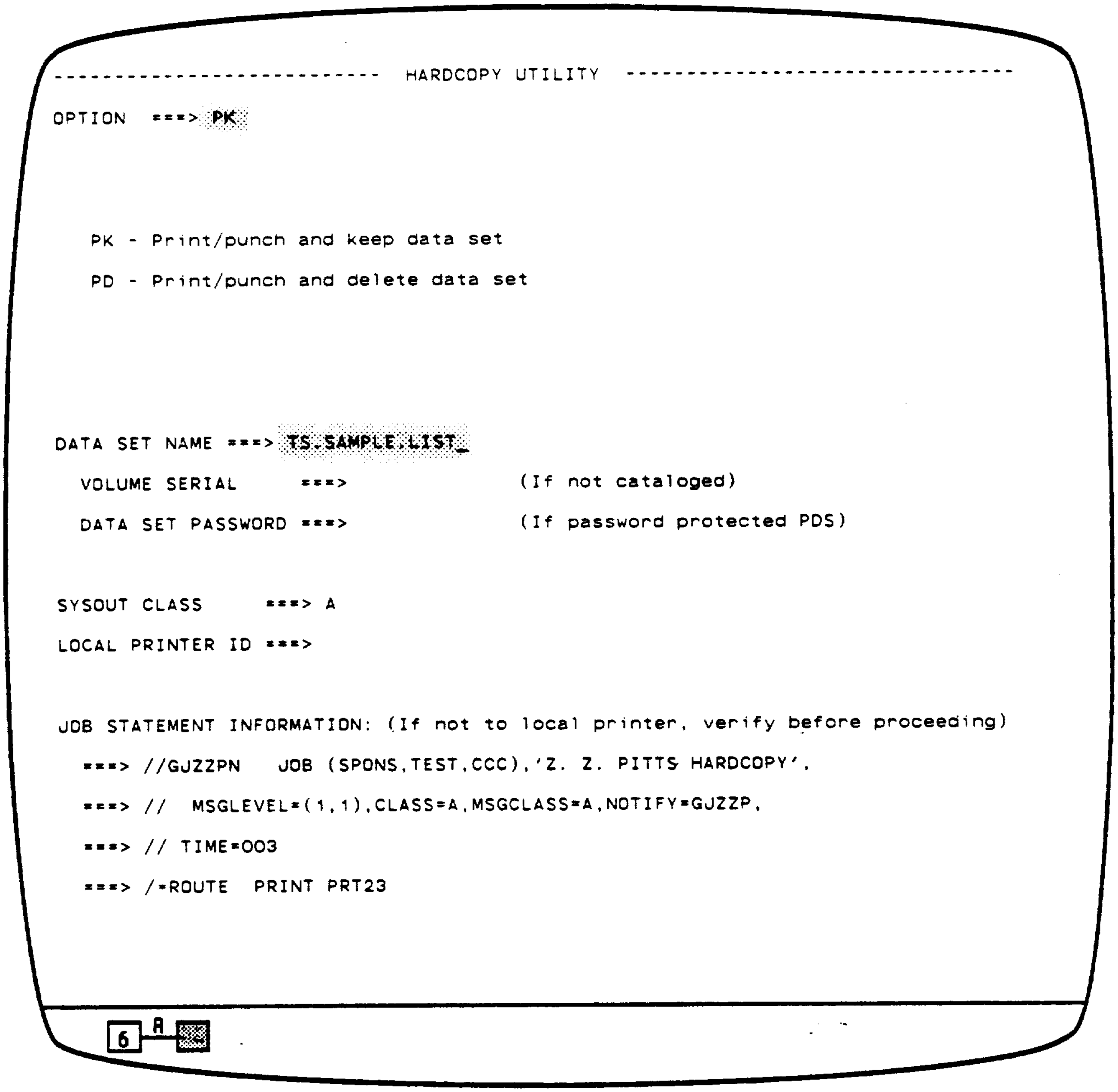
Scenario Step 7: Printing listing files



```
-------------------------- HARDCOPY UTILITY -------------------------------
OPTION  ===> _



   PK - Print/punch and keep data set

   PD - Print/punch and delete data set





DATA SET NAME ===>
  VOLUME SERIAL     ===>                (If not cataloged)
  DATA SET PASSWORD ===>                (If password protected PDS)


SYSOUT CLASS     ===> A
LOCAL PRINTER ID ===>


JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)
  ===> //GJZZPN   JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY',
  ===> //   MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP,
  ===> // TIME=003
  ===> /*ROUTE  PRINT PRT23
```

. . . we get to the hardcopy utility parameter entry panel.

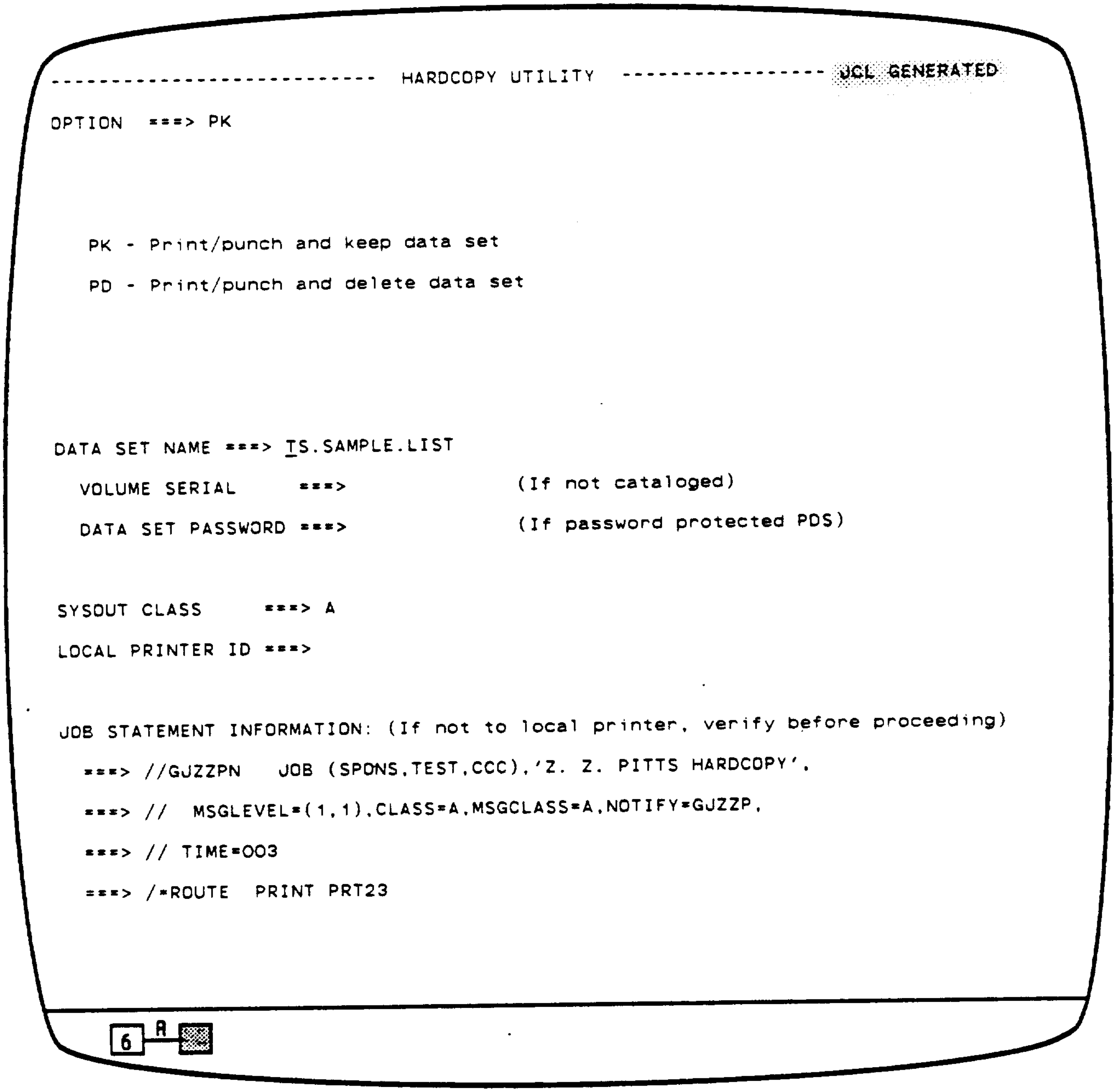
Scenario Step 7: Printing listing files



```
--------------------------- HARDCOPY UTILITY --------------------------------
OPTION  ===> PK



   PK - Print/punch and keep data set

   PD - Print/punch and delete data set




DATA SET NAME ===> TS.SAMPLE.LIST_
  VOLUME SERIAL      ===>              (If not cataloged)
  DATA SET PASSWORD ===>              (If password protected PDS)


SYSOUT CLASS      ===> A
LOCAL PRINTER ID ===>


JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)
  ===> //GJZZPN   JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY',
  ===> //  MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP,
  ===> // TIME=003
  ===> /*ROUTE  PRINT PRT23
```

Type in PK for a selection option, then <NL> to the data set name parameter entry field. TS.SAMPLE.LIST is the name of the compile output file. Type it in and press <enter>.

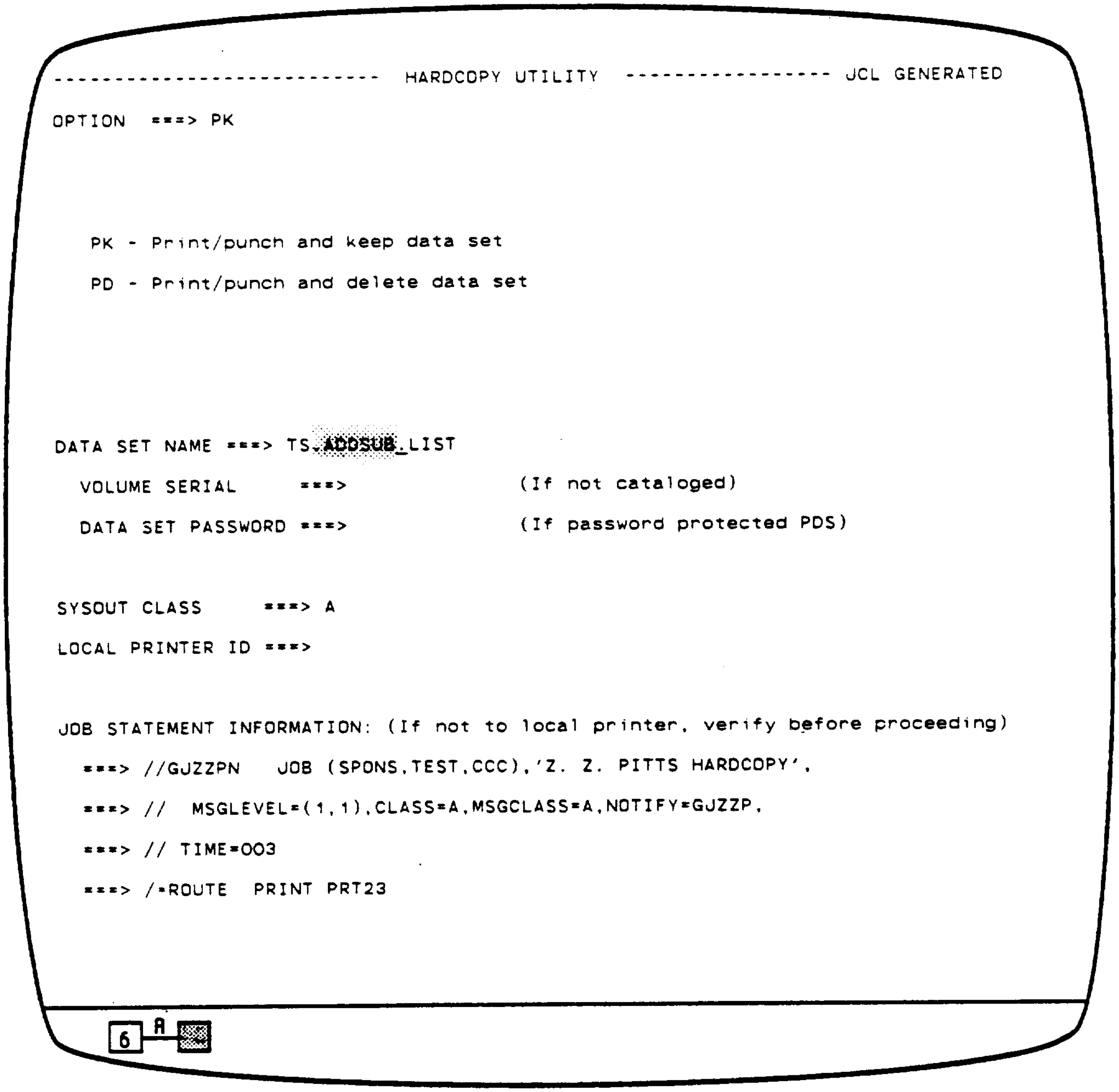
Scenario Step 7: Printing listing files



```
-------------------------- HARDCOPY UTILITY ----------------- JCL GENERATED
OPTION  ===> PK



   PK - Print/punch and keep data set

   PD - Print/punch and delete data set




DATA SET NAME ===> TS.SAMPLE.LIST
  VOLUME SERIAL     ===>              (If not cataloged)
  DATA SET PASSWORD ===>              (If password protected PDS)


SYSOUT CLASS      ===> A
LOCAL PRINTER ID ===>


JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)
  ===> //GJZZPN   JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY',
  ===> //  MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP,
  ===> // TIME=003
  ===> /*ROUTE  PRINT PRT23
```

The short message, JCL GENERATED, is displayed, indicating that a job control language file has been created and instructions that will print the data set have been put into it. The cursor will move to the beginning of the data set name parameter entry field.

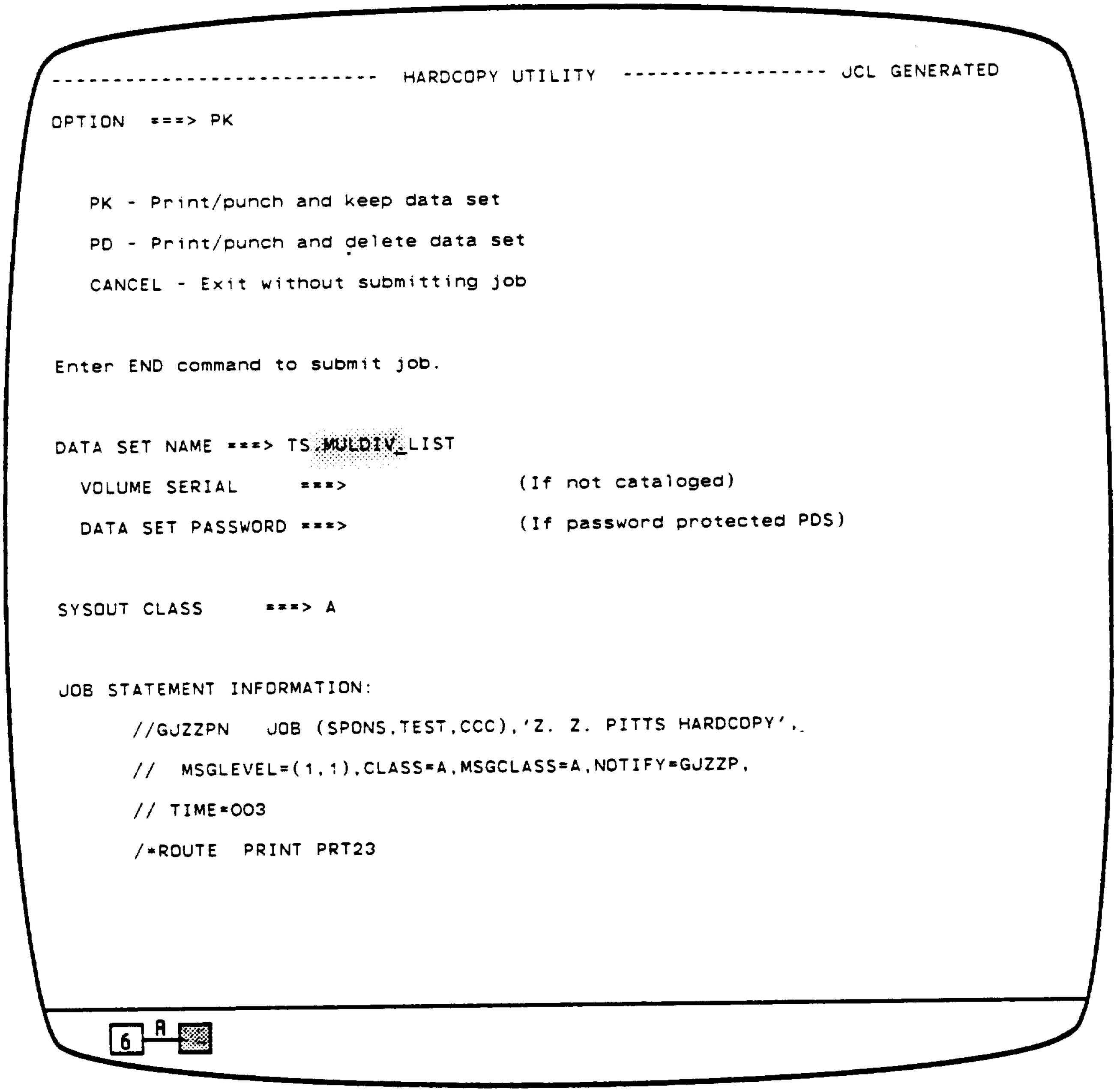
Scenario Step 7: Printing listing files



```
---------------------------  HARDCOPY UTILITY  -----------------  JCL GENERATED
OPTION  ===> PK



   PK - Print/punch and keep data set

   PD - Print/punch and delete data set





DATA SET NAME ===> TS.ADDSUB.LIST
  VOLUME SERIAL      ===>                (If not cataloged)
  DATA SET PASSWORD ===>                 (If password protected PDS)


SYSOUT CLASS     ===> A
LOCAL PRINTER ID ===>


JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)
  ===> //GJZZPN   JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY',
  ===> //  MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP,
  ===> // TIME=003
  ===> /*ROUTE  PRINT PRT23
```

Type in the name of the ADDSUB subroutine and press <enter> . . .

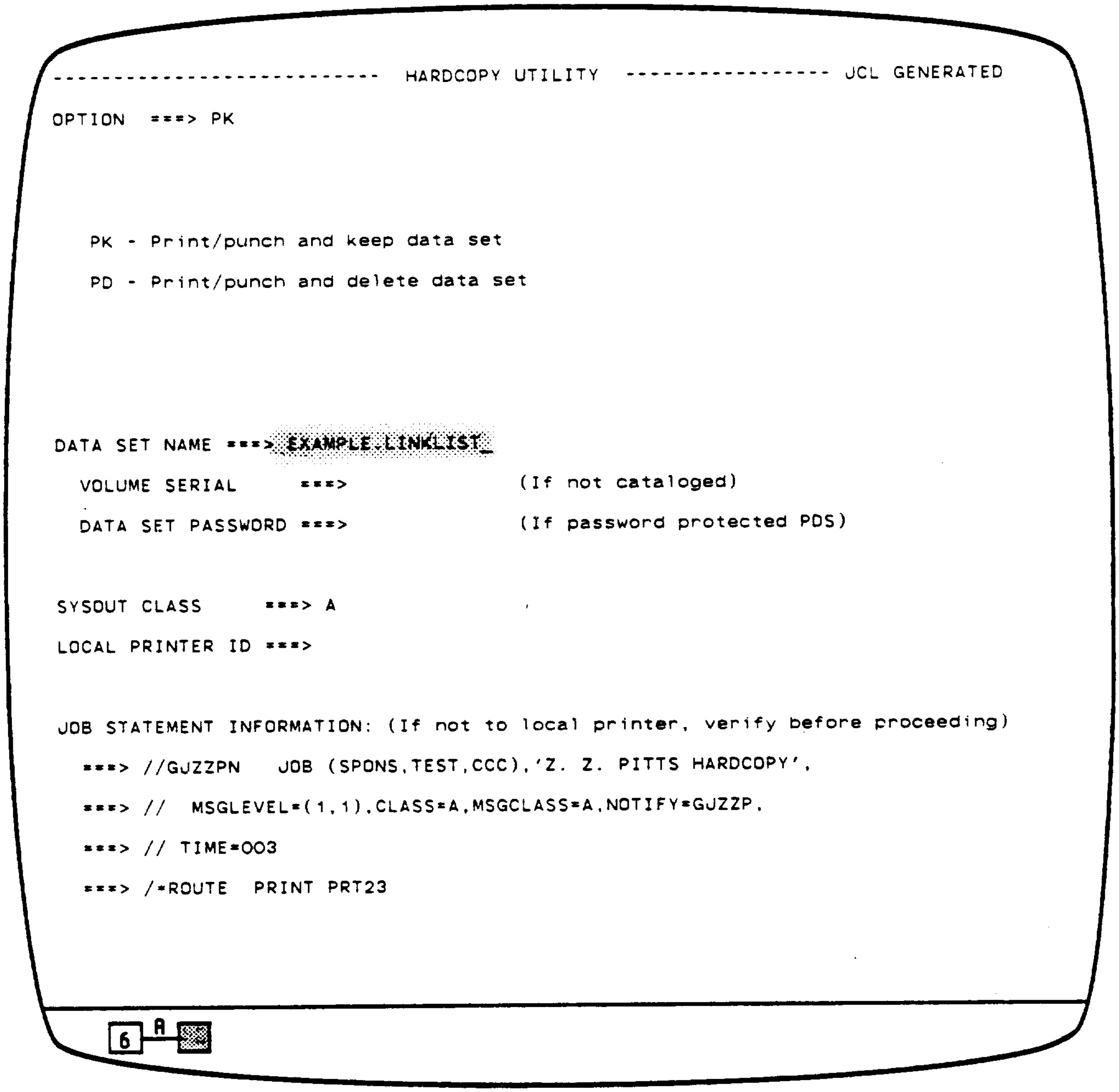
Scenario Step 7: Printing listing files



```
--------------------------- HARDCOPY UTILITY ----------------- JCL GENERATED
OPTION  ===> PK


   PK - Print/punch and keep data set

   PD - Print/punch and delete data set

   CANCEL - Exit without submitting job


Enter END command to submit job.


DATA SET NAME ===> TS.MULDIV.LIST
  VOLUME SERIAL     ===>                (If not cataloged)
  DATA SET PASSWORD ===>                (If password protected PDS)


SYSOUT CLASS     ===> A


JOB STATEMENT INFORMATION:
      //GJZZPN   JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY',
      //  MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP,
      // TIME=003
      /*ROUTE  PRINT PRT23
```

. . . and the JCL GENERATED message remains. The JCL to print the ADDSUB listing file has been added to the file created for SAMPLE. Move the cursor and type MULDIV over top of ADDSUB; then press <enter> . . .

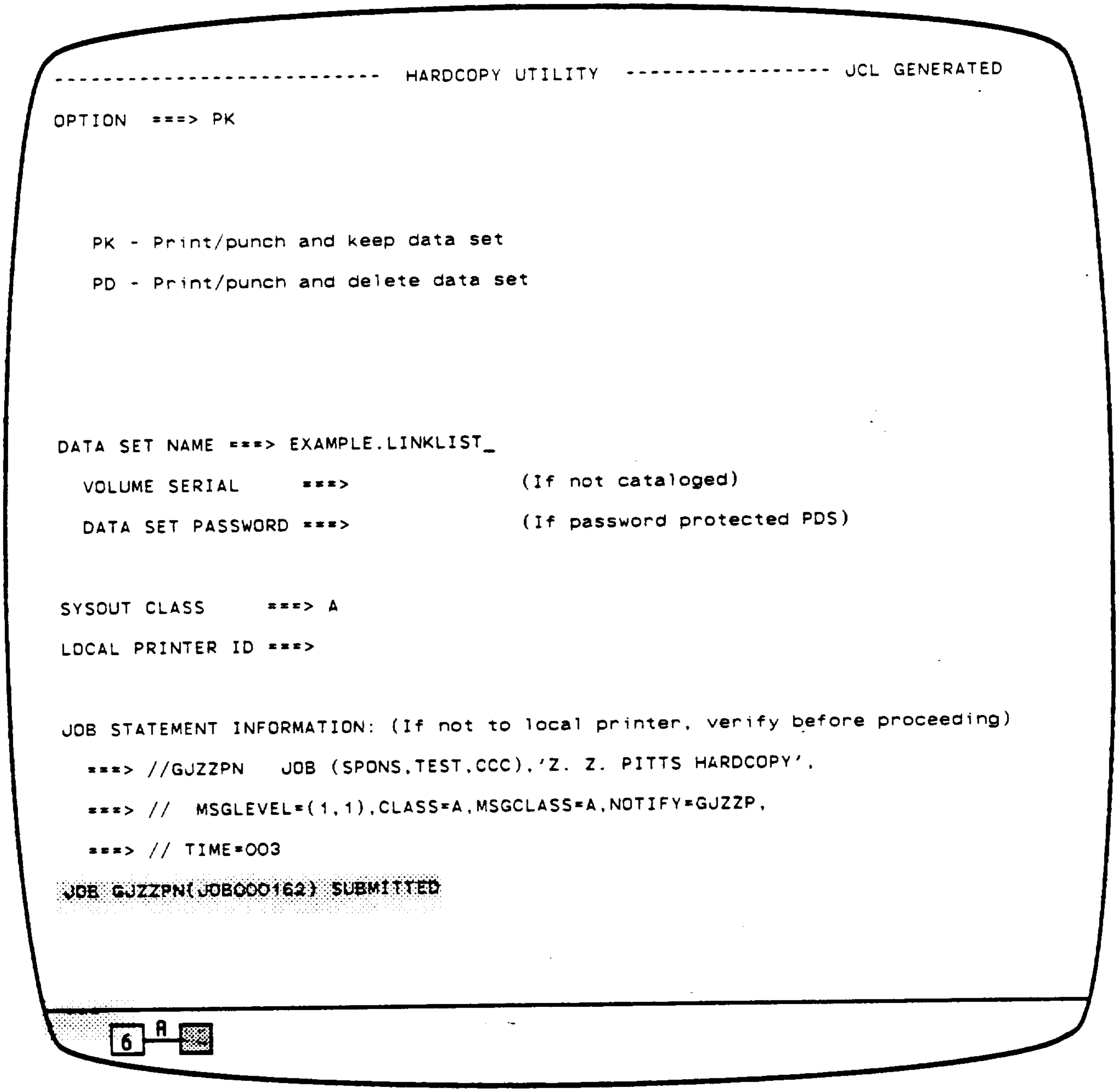
Scenario Step 7: Printing listing files



```
----------------------------  HARDCOPY UTILITY  ----------------- JCL GENERATED
OPTION  ===> PK



   PK - Print/punch and keep data set

   PD - Print/punch and delete data set




DATA SET NAME ===> EXAMPLE.LINKLIST_
  VOLUME SERIAL      ===>               (If not cataloged)
  DATA SET PASSWORD ===>                (If password protected PDS)

SYSOUT CLASS      ===> A
LOCAL PRINTER ID ===>

JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)
  ===> //GJZZPN   JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY',
  ===> //  MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP,
  ===> // TIME=003
  ===> /*ROUTE  PRINT PRT23
```

. . . and instructions to print the MULDIV listing file are added to the JCL file. Now move the cursor and add the linkage editor listing file to the list of files to be printed. Press <enter> and the JCL GENERATED message remains. Press <end> and . . .

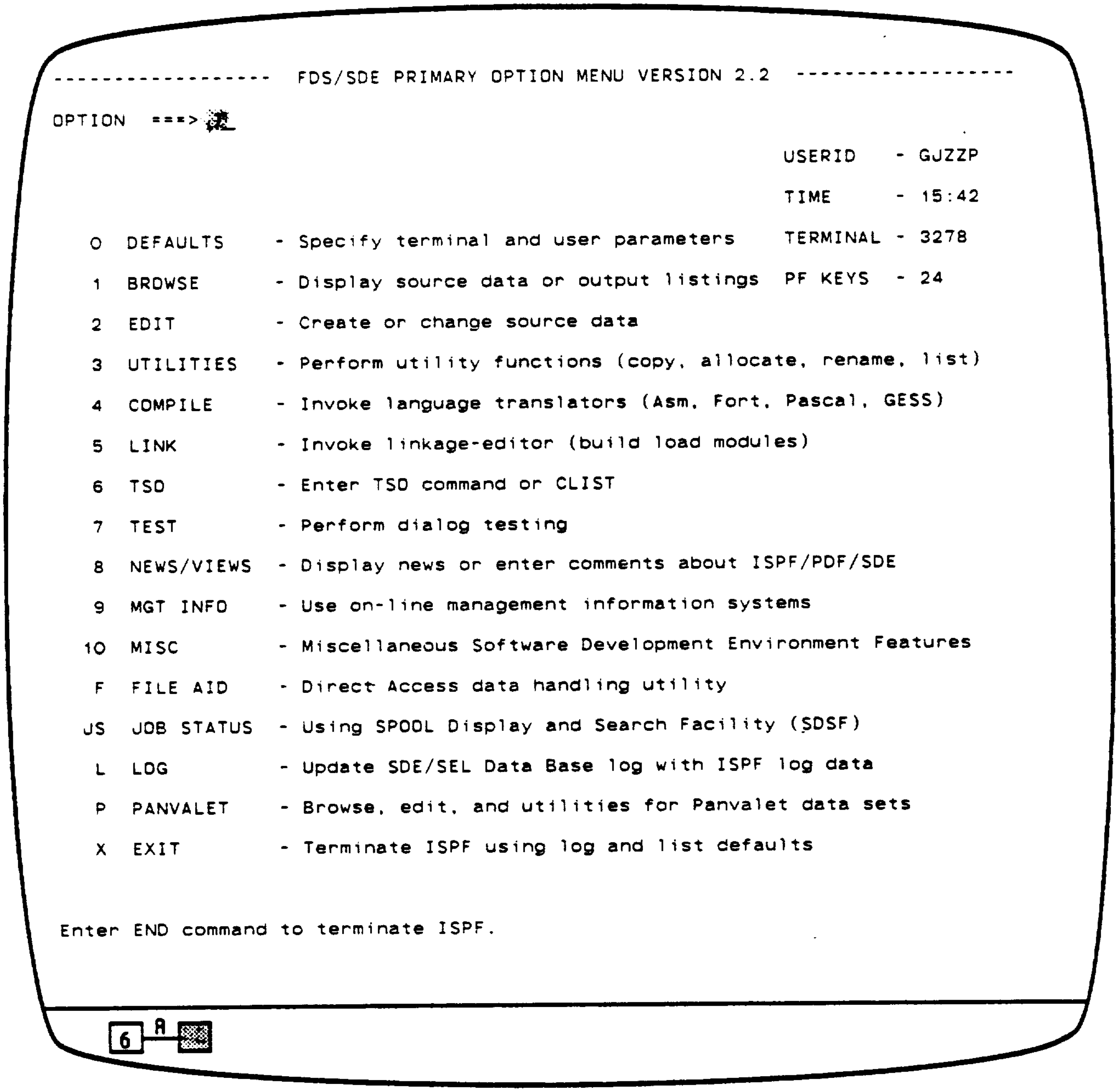
Scenario Step 7: Printing listing files



```
--------------------------- HARDCOPY UTILITY ----------------- JCL GENERATED
OPTION  ===> PK



   PK - Print/punch and keep data set

   PD - Print/punch and delete data set




DATA SET NAME ===> EXAMPLE.LINKLIST_
  VOLUME SERIAL     ===>               (If not cataloged)
  DATA SET PASSWORD ===>               (If password protected PDS)

SYSOUT CLASS      ===> A
LOCAL PRINTER ID ===>

JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)
  ===> //GJZZPN   JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY',
  ===> //  MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP,
  ===> // TIME=003
JOB GJZZPN(JOB00162) SUBMITTED
```

. . . and a job submission message is displayed. Three asterisks remain at the bottom of the screen until we press <enter> to acknowledge. At that time . . .

Scenario Step 8: Updating SDE log and exiting the SDE

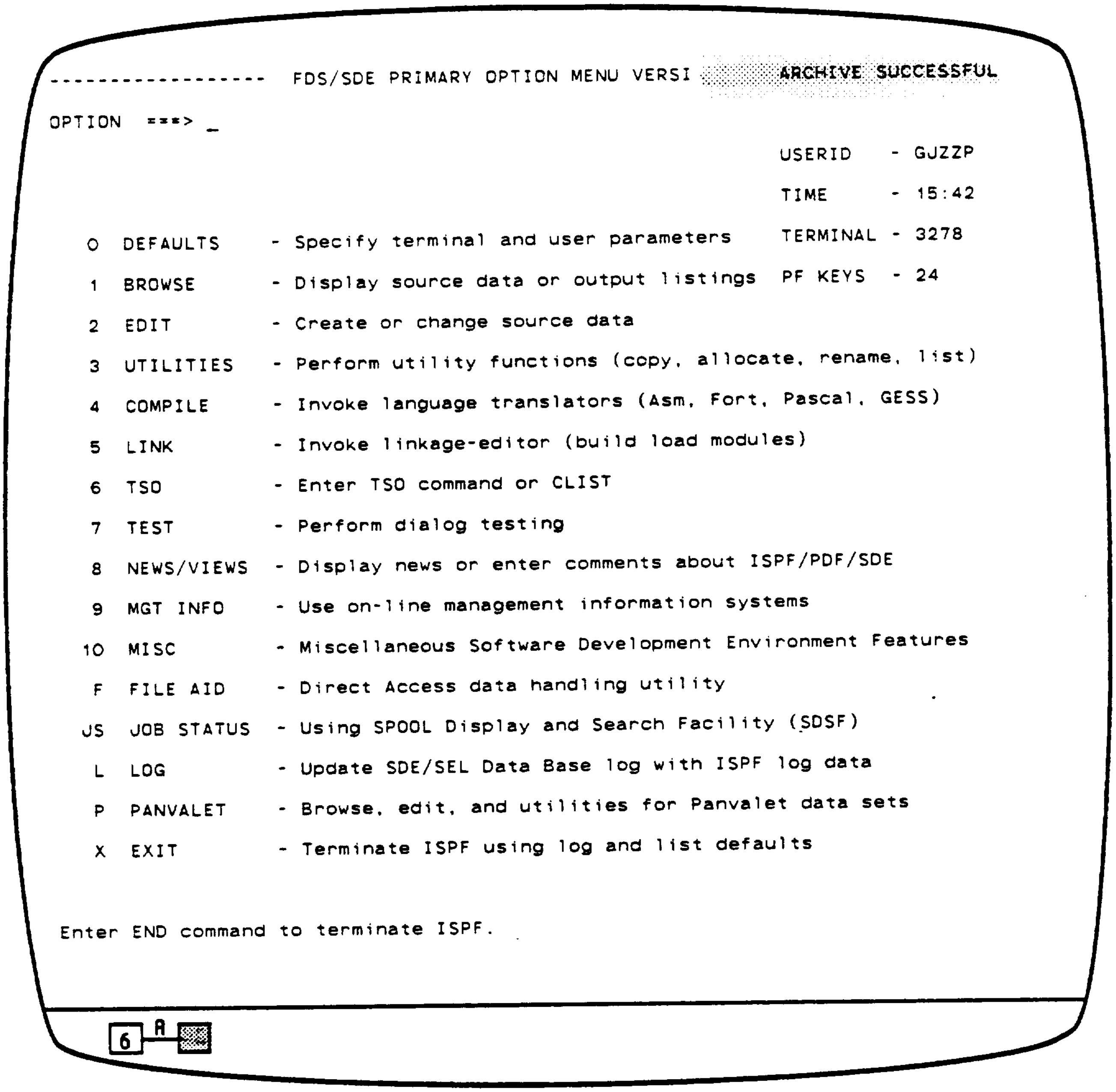


```
-----------------  FDS/SDE PRIMARY OPTION MENU VERSION 2.2  -----------------
OPTION  ===>
                                                             USERID   - GJZZP
                                                             TIME     - 15:42
   0  DEFAULTS     - Specify terminal and user parameters    TERMINAL - 3278
   1  BROWSE       - Display source data or output listings  PF KEYS  - 24
   2  EDIT         - Create or change source data
   3  UTILITIES    - Perform utility functions (copy, allocate, rename, list)
   4  COMPILE      - Invoke language translators (Asm, Fort, Pascal, GESS)
   5  LINK         - Invoke linkage-editor (build load modules)
   6  TSO          - Enter TSO command or CLIST
   7  TEST         - Perform dialog testing
   8  NEWS/VIEWS   - Display news or enter comments about ISPF/PDF/SDE
   9  MGT INFO     - Use on-line management information systems
  10  MISC         - Miscellaneous Software Development Environment Features
   F  FILE AID     - Direct Access data handling utility
  JS  JOB STATUS   - Using SPOOL Display and Search Facility (SDSF)
   L  LOG          - Update SDE/SEL Data Base log with ISPF log data
   P  PANVALET     - Browse, edit, and utilities for Panvalet data sets
   X  EXIT         - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

. . . we are back at the primary option panel.

Type L to select the log update function. This copies our ISPF session log to an archive file so the SDE development team can see which options get the most use and which ones get the most errors. They will provide more help panels and/or redesign functions that generate a lot of error messages.

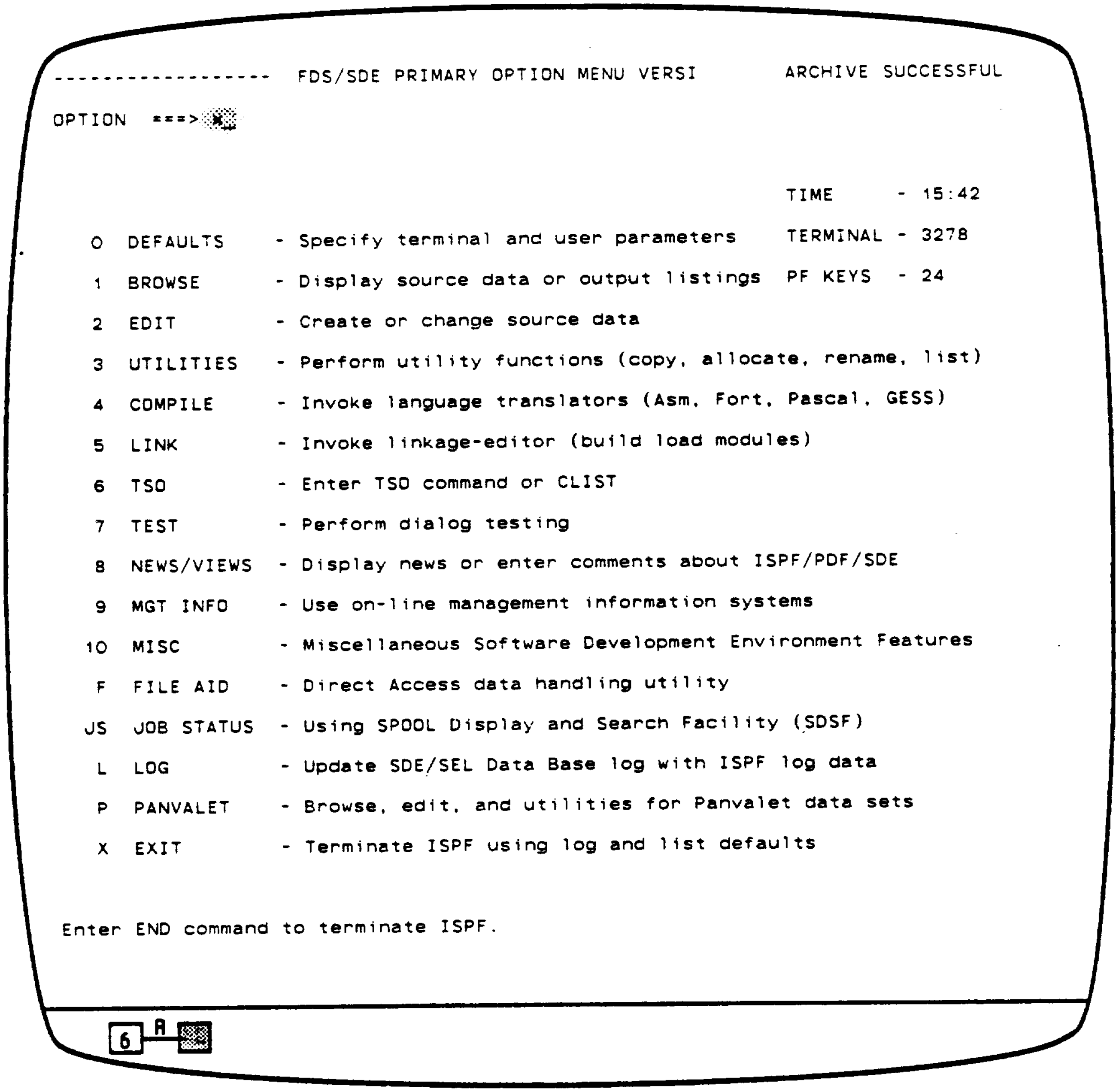
Scenario Step 8: Updating SDE log and exiting the SDE



```
------------------ FDS/SDE PRIMARY OPTION MENU VERSI ARCHIVE SUCCESSFUL
OPTION  ===> _
                                                  USERID   - GJZZP
                                                  TIME     - 15:42
   0  DEFAULTS    - Specify terminal and user parameters   TERMINAL - 3278
   1  BROWSE      - Display source data or output listings PF KEYS  - 24
   2  EDIT        - Create or change source data
   3  UTILITIES   - Perform utility functions (copy, allocate, rename, list)
   4  COMPILE     - Invoke language translators (Asm, Fort, Pascal, GESS)
   5  LINK        - Invoke linkage-editor (build load modules)
   6  TSO         - Enter TSO command or CLIST
   7  TEST        - Perform dialog testing
   8  NEWS/VIEWS  - Display news or enter comments about ISPF/PDF/SDE
   9  MGT INFO    - Use on-line management information systems
  10  MISC        - Miscellaneous Software Development Environment Features
   F  FILE AID    - Direct Access data handling utility
  JS  JOB STATUS  - Using SPOOL Display and Search Facility (SDSF)
   L  LOG         - Update SDE/SEL Data Base log with ISPF log data
   P  PANVALET    - Browse, edit, and utilities for Panvalet data sets
   X  EXIT        - Terminate ISPF using log and list defaults

 Enter END command to terminate ISPF.
```

In a minute or two, the message ARCHIVE SUCCESSFUL will be displayed in the short message area.

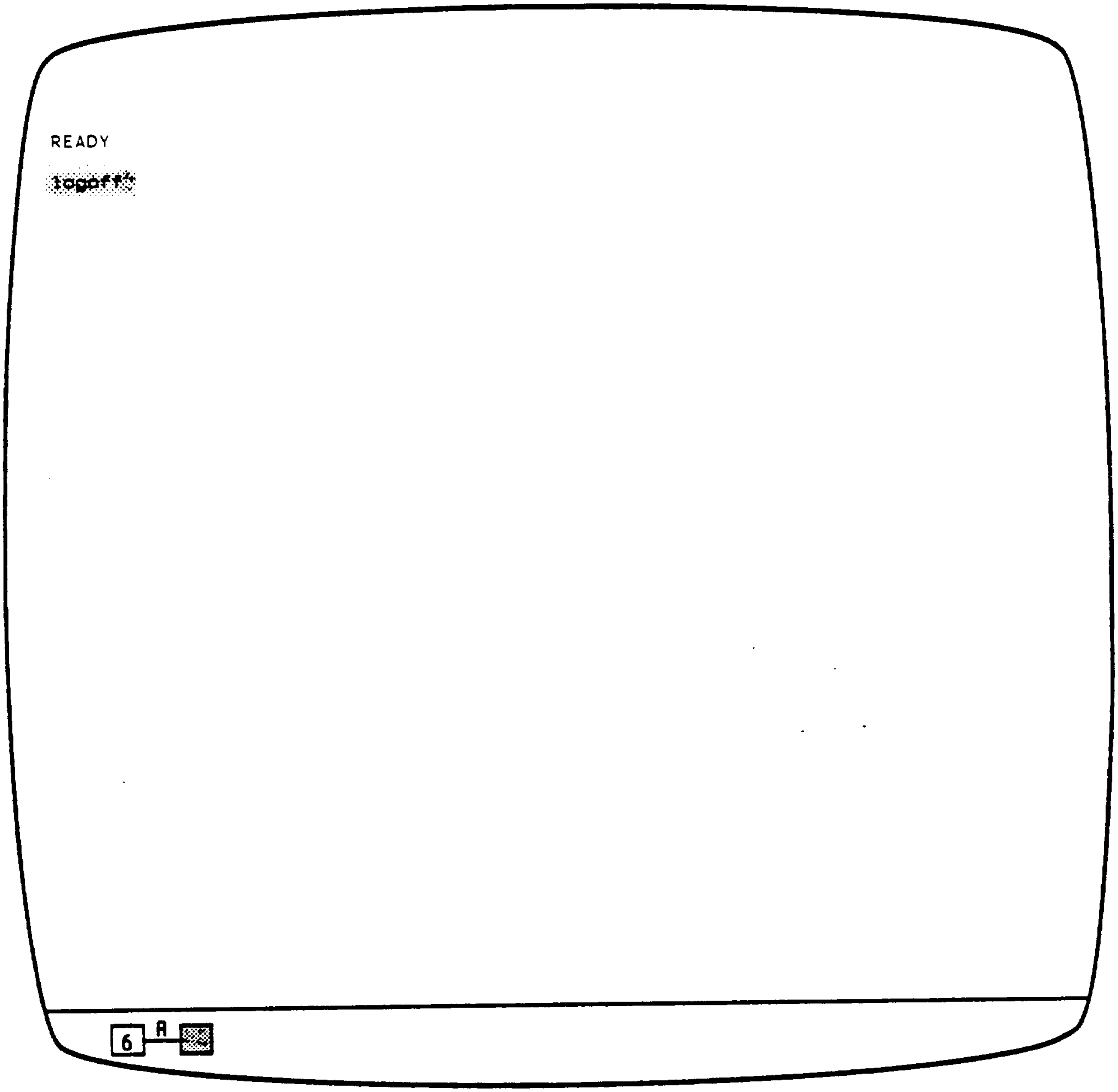
Scenario Step 8: Updating SDE log and exiting the SDE



```
-----------------  FDS/SDE PRIMARY OPTION MENU VERSI      ARCHIVE SUCCESSFUL
OPTION  ===> X

                                                          TIME     - 15:42
   0  DEFAULTS     - Specify terminal and user parameters  TERMINAL - 3278
   1  BROWSE       - Display source data or output listings PF KEYS  - 24
   2  EDIT         - Create or change source data
   3  UTILITIES    - Perform utility functions (copy, allocate, rename, list)
   4  COMPILE      - Invoke language translators (Asm, Fort, Pascal, GESS)
   5  LINK         - Invoke linkage-editor (build load modules)
   6  TSO          - Enter TSO command or CLIST
   7  TEST         - Perform dialog testing
   8  NEWS/VIEWS   - Display news or enter comments about ISPF/PDF/SDE
   9  MGT INFO     - Use on-line management information systems
  10  MISC         - Miscellaneous Software Development Environment Features
   F  FILE AID     - Direct Access data handling utility
  JS  JOB STATUS   - Using SPOOL Display and Search Facility (SDSF)
   L  LOG          - Update SDE/SEL Data Base log with ISPF log data
   P  PANVALET     - Browse, edit, and utilities for Panvalet data sets
   X  EXIT         - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

Type X as the option to exit. The screen will clear; and soon . . .

Scenario Step 8: Updating SDE log and exiting the SDE



```
READY
logoff
```

. . . the TSO READY acknowledgement will appear. Now, type in logoff and you will be off the system. Best of luck in your software development adventures . . . aided by the SDE.

STANDARD BIBLIOGRAPHY OF SEL LITERATURE

The technical papers, memorandums, and documents listed in this bibliography are organized into two groups. The first group is composed of documents issued by the Software Engineering Laboratory (SEL) during its research and development activities. The second group includes materials that were published elsewhere but pertain to SEL activities.

SEL-ORIGINATED DOCUMENTS

SEL-76-001, Proceedings From the First Summer Software Engineering Workshop, August 1976

SEL-77-001, The Software Engineering Laboratory, V. R. Basili, M. V. Zelkowitz, F. E. McGarry, et al., May 1977

SEL-77-002, Proceedings From the Second Summer Software Engineering Workshop, September 1977

SEL-77-003, Structured FORTRAN Preprocessor (SFORT), B. Chu and D. S. Wilson, September 1977

SEL-77-004, GSFC NAVPAK Design Specifications Languages Study, P. A. Scheffer and C. E. Velez, October 1977

SEL-78-001, FORTRAN Static Source Code Analyzer (SAP) Design and Module Descriptions, E. M. O'Neill, S. R. Waligora, and C. E. Goorevich, February 1978

SEL-78-003, Evaluation of Draper NAVPAK Software Design, K. Tasaki and F. E. McGarry, June 1978

SEL-78-004, Structured FORTRAN Preprocessor (SFORT) PDP-11/70 User's Guide, D. S. Wilson and B. Chu, September 1978

SEL-78-005, Proceedings From the Third Summer Software Engineering Workshop, September 1978

SEL-78-006, GSFC Software Engineering Research Requirements Analysis Study, P. A. Scheffer and C. E. Velez, November 1978

SEL-78-007, Applicability of the Rayleigh Curve to the SEL Environment, T. E. Mapp, December 1978

SEL-78-302, *FORTRAN Static Source Code Analyzer Program (SAP) User's Guide (Revision 3)*, W. J. Decker and W. A. Taylor, July 1986

SEL-79-001, *SIMPL-D Data Base Reference Manual*, M. V. Zelkowitz, July 1979

SEL-79-002, *The Software Engineering Laboratory: Relationship Equations*, K. Freburger and V. R. Basili, May 1979

SEL-79-003, *Common Software Module Repository (CSMR) System Description and User's Guide*, C. E. Goorevich, A. L. Green, and S. R. Waligora, August 1979

SEL-79-004, *Evaluation of the Caine, Farber, and Gordon Program Design Language (PDL) in the Goddard Space Flight Center (GSFC) Code 580 Software Design Environment*, C. E. Goorevich, A. L. Green, and W. J. Decker, September 1979

SEL-79-005, *Proceedings From the Fourth Summer Software Engineering Workshop*, November 1979

SEL-80-001, *Functional Requirements/Specifications for Code 580 Configuration Analysis Tool (CAT)*, F. K. Banks, A. L. Green, and C. E. Goorevich, February 1980

SEL-80-002, *Multi-Level Expression Design Language-Requirement Level (MEDL-R) System Evaluation*, W. J. Decker and C. E. Goorevich, May 1980

SEL-80-003, *Multimission Modular Spacecraft Ground Support Software System (MMS/GSSS) State-of-the-Art Computer Systems/Compatibility Study*, T. Welden, M. McClellan, and P. Liebertz, May 1980

SEL-80-005, *A Study of the Musa Reliability Model*, A. M. Miller, November 1980

SEL-80-006, *Proceedings From the Fifth Annual Software Engineering Workshop*, November 1980

SEL-80-007, *An Appraisal of Selected Cost/Resource Estimation Models for Software Systems*, J. F. Cook and F. E. McGarry, December 1980

SEL-80-104, *Configuration Analysis Tool (CAT) System Description and User's Guide (Revision 1)*, W. Decker and W. Taylor, December 1982

SEL-81-008, Cost and Reliability Estimation Models (CAREM) User's Guide, J. F. Cook and E. Edwards, February 1981

SEL-81-009, Software Engineering Laboratory Programmer Workbench Phase 1 Evaluation, W. J. Decker and F. E. McGarry, March 1981

SEL-81-011, Evaluating Software Development by Analysis of Change Data, D. M. Weiss, November 1981

SEL-81-012, The Rayleigh Curve As a Model for Effort Distribution Over the Life of Medium Scale Software Systems, G. O. Picasso, December 1981

SEL-81-013, Proceedings From the Sixth Annual Software Engineering Workshop, December 1981

SEL-81-014, Automated Collection of Software Engineering Data in the Software Engineering Laboratory (SEL), A. L. Green, W. J. Decker, and F. E. McGarry, September 1981

SEL-81-101, Guide to Data Collection, V. E. Church, D. N. Card, F. E. McGarry, et al., August 1982

SEL-81-102, Software Engineering Laboratory (SEL) Data Base Organization and User's Guide Revision 1, P. Lo and D. Wyckoff, July 1983

SEL-81-104, The Software Engineering Laboratory, D. N. Card, F. E. McGarry, G. Page, et al., February 1982

SEL-81-106, Software Engineering Laboratory (SEL) Document Library (DOCLIB) System Description and User's Guide, W. Taylor and W. J. Decker, May 1985

SEL-81-107, Software Engineering Laboratory (SEL) Compendium of Tools, W. J. Decker, W. A. Taylor, and E. J. Smith, February 1982

SEL-81-110, Evaluation of an Independent Verification and Validation (IV&V) Methodology for Flight Dynamics, G. Page, F. E. McGarry, and D. N. Card, June 1985

SEL-81-203, Software Engineering Laboratory (SEL) Data Base Maintenance System (DBAM) User's Guide and System Description, P. Lo, June 1984

SEL-81-205, Recommended Approach to Software Development, F. E. McGarry, G. Page, S. Eslinger, et al., April 1983

SEL-82-001, Evaluation of Management Measures of Software Development, G. Page, D. N. Card, and F. E. McGarry, September 1982, vols. 1 and 2

SEL-82-003, Software Engineering Laboratory (SEL) Data Base Reporting Software User's Guide and System Description, P. Lo, September 1982

SEL-82-004, Collected Software Engineering Papers: Volume 1, July 1982

SEL-82-007, Proceedings From the Seventh Annual Software Engineering Workshop, December 1982

SEL-82-008, Evaluating Software Development by Analysis of Changes: The Data From the Software Engineering Laboratory, V. R. Basili and D. M. Weiss, December 1982

SEL-82-102, FORTRAN Static Source Code Analyzer Program (SAP) System Description (Revision 1), W. A. Taylor and W. J. Decker, April 1985

SEL-82-105, Glossary of Software Engineering Laboratory Terms, T. A. Babst, F. E. McGarry, and M. G. Rohleder, October 1983

SEL-82-306, Annotated Bibliography of Software Engineering Laboratory Literature, D. N. Card, Q. L. Jordan, and F. E. McGarry, November 1985

SEL-83-001, An Approach to Software Cost Estimation, F. E. McGarry, G. Page, D. N. Card, et al., February 1984

SEL-83-002, Measures and Metrics for Software Development, D. N. Card, F. E. McGarry, G. Page, et al., March 1984

SEL-83-003, Collected Software Engineering Papers: Volume II, November 1983

SEL-83-006, Monitoring Software Development Through Dynamic Variables, C. W. Doerflinger, November 1983

SEL-83-007, Proceedings From the Eighth Annual Software Engineering Workshop, November 1983

SEL-83-104, Software Engineering Laboratory (SEL) Data Base Retrieval System (DARES) User's Guide, T. A. Babst, W. J. Decker, P. Lo, and W. Miller, August 1984

SEL-83-105, Software Engineering Laboratory (SEL) Data Base Retrieval System (DARES) System Description, P. Lo, W. J. Decker, and W. Miller, August 1984

SEL-84-001, Manager's Handbook for Software Development, W. W. Agresti, F. E. McGarry, D. N. Card, et al., April 1984

SEL-84-002, Configuration Management and Control: Policies and Procedures, Q. L. Jordan and E. Edwards, December 1984

SEL-84-003, Investigation of Specification Measures for the Software Engineering Laboratory (SEL), W. W. Agresti, V. E. Church, and F. E. McGarry, December 1984

SEL-84-004, Proceedings From the Ninth Annual Software Engineering Workshop, November 1984

SEL-85-001, A Comparison of Software Verification Techniques, D. N. Card, R. W. Selby, Jr., F. E. McGarry, et al., April 1985

SEL-85-002, Ada Training Evaluation and Recommendations From the Gamma Ray Observatory Ada Development Team, R. Murphy and M. Stark, October 1985

SEL-85-003, Collected Software Engineering Papers: Volume III, November 1985

SEL-85-004, Evaluations of Software Technologies: Testing, CLEANROOM, and Metrics, R. W. Selby, Jr., May 1985

SEL-85-005, Software Verification and Testing, D. N. Card, C. Antle, and E. Edwards, December 1985

SEL-86-001, Programmer's Handbook for Flight Dynamics Software Development, R. Wood and E. Edwards, March 1986

SEL-86-002, General Object-Oriented Software Development, E. Seidewitz and M. Stark, July 1986

SEL-86-003, Flight Dynamics System Software Development Environment Tutorial, J. Buell and P. Myers, July 1986

SEL-RELATED LITERATURE

Agresti, W. W., Definition of Specification Measures for the Software Engineering Laboratory, Computer Sciences Corporation, CSC/TM-84/6085, June 1984

Agresti, W. W., Tutorial: New Paradigms for Software Development. New York: IEEE Computer Society Press, July 1986

Agresti, W. W., V. E. Church, D. N. Card, and P. L. Lo, "Designing With Ada for Satellite Simulation: A Case Study," Proceedings of the First International Symposium on Ada for the NASA Space Station, June 1986

[2]Agresti, W. W., F. E. McGarry, D. N. Card, et al., "Measuring Software Technology," Program Transformation and Programming Environments. New York: Springer-Verlag, 1984

[3]Bailey, J. W., and V. R. Basili, "A Meta-Model for Software Development Resource Expenditures," Proceedings of the Fifth International Conference on Software Engineering. New York: IEEE Computer Society Press, 1981

Basili, V. R., "SEL Relationships for Programming Measurement and Estimation," University of Maryland, Technical Memorandum, October 1979

[3]Basili, V. R., "Models and Metrics for Software Management and Engineering," ASME Advances in Computer Technology, January 1980, vol. 1

Basili, V. R., Tutorial on Models and Metrics for Software Management and Engineering. New York: IEEE Computer Society Press, 1980 (also designated SEL-80-008)

[1]Basili, V. R., "Quantitative Evaluation of Software Methodology," Proceedings of the First Pan-Pacific Computer Conference, September 1985

[3]Basili, V. R., and J. Beane, "Can the Parr Curve Help With Manpower Distribution and Resource Estimation Problems?", Journal of Systems and Software, February 1981, vol. 2, no. 1

[3]Basili, V. R., and K. Freburger, "Programming Measurement and Estimation in the Software Engineering Laboratory," Journal of Systems and Software, February 1981, vol. 2, no. 1

[1]Basili, V. R., and N. M. Panlilio-Yap, "Finding Relationships Between Effort and Other Variables in the SEL," Proceedings of the International Computer Software and Applications Conference, October 1985

[2]Basili, V. R., and B. T. Perricone, "Software Errors and Complexity: An Empirical Investigation," Communications of the ACM, January 1984, vol. 27, no. 1

[3]Basili, V. R., and T. Phillips, "Evaluating and Comparing Software Metrics in the Software Engineering Laboratory," *Proceedings of the ACM SIGMETRICS Symposium/Workshop: Quality Metrics*, March 1981

[1]Basili, V. R., and C. L. Ramsey, "ARROWSMITH-P--A Prototype Expert System for Software Engineering Management," *Proceedings of the IEEE/MITRE Expert Systems in Government Symposium*, October 1985

Basili, V. R., and R. Reiter, "Evaluating Automatable Measures for Software Development," *Proceedings of the Workshop on Quantitative Software Models for Reliability, Complexity and Cost*, October 1979

[2]Basili, V. R., R. W. Selby, and T. Phillips, "Metric Analysis and Data Validation Across FORTRAN Projects," *IEEE Transactions on Software Engineering*, November 1983

[1]Basili, V. R., and R. W. Selby, Jr., "Calculation and Use of an Environments's Characteristic Software Metric Set," *Proceedings of the Eighth International Conference on Software Engineering*, August 1985

Basili, V. R., and R. W. Selby, Jr., *Comparing the Effectiveness of Software Testing Strategies*, University of Maryland Technical Report, TR-1501, May 1985

Basili, V. R., R. W. Selby, Jr., and D. H. Hutchens, "Experimentation in Software Engineering," *IEEE Transactions on Software Engineering*, July 1986

[2]Basili, V.R., and D. M. Weiss, *A Methodology for Collecting Valid Software Engineering Data*, University of Maryland, Technical Report TR-1235, December 1982

[1]Basili, V. R., and D. M. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEEE Transactions on Software Engineering*, November 1984

[3]Basili, V. R., and M. V. Zelkowitz, "The Software Engineering Laboratory: Objectives," *Proceedings of the Fifteenth Annual Conference on Computer Personnel Research*, August 1977

Basili, V. R., and M. V. Zelkowitz, "Designing a Software Measurement Experiment," *Proceedings of the Software Life Cycle Management Workshop*, September 1977

[3]Basili, V. R., and M. V. Zelkowitz, "Operation of the Software Engineering Laboratory," Proceedings of the Second Software Life Cycle Management Workshop, August 1978

[3]Basili, V. R., and M. V. Zelkowitz, "Measuring Software Development Characteristics in the Local Environment," Computers and Structures, August 1978, vol. 10

Basili, V. R., and M. V. Zelkowitz, "Analyzing Medium Scale Software Development," Proceedings of the Third International Conference on Software Engineering. New York: IEEE Computer Society Press, 1978

[1]Card, D. N., "A Software Technology Evaluation Program," Annais do XVIII Congresso Nacional de Informatica, October 1985

Card, D. N., V. E. Church, and W. W. Agresti, "An Empirical Study of Software Design Practices," IEEE Transactions on Software Engineering, February 1986

[1]Card, D. N., G. T. Page, and F. E. McGarry, "Criteria for Software Modularization," Proceedings of the Eighth International Conference on Software Engineering, August 1985

[3]Chen, E., and M. V. Zelkowitz, "Use of Cluster Analysis To Evaluate Software Engineering Methodologies," Proceedings of the Fifth International Conference on Software Engineering. New York: IEEE Computer Society Press, 1981

[2]Doerflinger, C. W., and V. R. Basili, "Monitoring Software Development Through Dynamic Variables," Proceedings of the Seventh International Computer Software and Applications Conference. New York: IEEE Computer Society Press, 1983

Higher Order Software, Inc., TR-9, A Demonstration of AXES for NAVPAK, M. Hamilton and S. Zeldin, September 1977 (also designated SEL-77-005)

[1]McGarry, F. E., J. Valett, and D. Hall, "Measuring the Impact of Computer Resource Quality on the Software Development Process and Product," Proceedings of the Hawaiian International Conference on System Sciences, January 1985

[1]Page, G., F. E. McGarry, and D. N. Card, "A Practical Experience With Independent Verification and Validation," Proceedings of the Eighth International Computer Software and Applications Conference, November 1984

[1]Ramsey, J., and V. R. Basili, "Analyzing the Test Process Using Structural Coverage," Proceedings of the Eighth International Conference on Software Engineering, August 1985

Turner, C., and G. Caron, A Comparison of RADC and NASA/SEL Software Development Data, Data and Analysis Center for Software, Special Publication, May 1981

Turner, C., G. Caron, and G. Brement, NASA/SEL Data Compendium, Data and Analysis Center for Software, Special Publication, April 1981

[1]Weiss, D. M., and V. R. Basili, "Evaluating Software Development by Analysis of Changes: Some Data From the Software Engineering Laboratory," IEEE Transactions on Software Engineering, February 1985

[3]Zelkowitz, M. V., "Resource Estimation for Medium Scale Software Projects," Proceedings of the Twelfth Conference on the Interface of Statistics and Computer Science. New York: IEEE Computer Society Press, 1979

[2]Zelkowitz, M. V., "Data Collection and Evaluation for Experimental Computer Science Research," Empirical Foundations for Computer and Information Science (proceedings), November 1982

Zelkowitz, M. V., and V. R. Basili, "Operational Aspects of a Software Measurement Facility," Proceedings of the Software Life Cycle Management Workshop, September 1977

[1] This article also appears in SEL-85-003, Collected Software Engineering Papers: Volume III, November 1985.

[2] This article also appears in SEL-83-003, Collected Software Engineering Papers: Volume II, November 1983.

[3] This article also appears in SEL-82-004, Collected Software Engineering Papers: Volume I, July 1982.

REFERENCES

1. IBM Corporation, Interactive System Productivity Facility/Program Development Facility, Version 2, Program Reference SC34-2139-0, March 1984

2. Computer Sciences Corporation, CSC/SD-85/6056, Flight Dynamics Systems Software Development Environment (FDS/SDE) User's Guide (Preliminary), J. Buell and W. Miller, December 1985