

decsystem10
**GETTING STARTED
WITH DECSYSTEM-10**

March 1975

Additional copies of this manual may be ordered from: Software Distribution Center,
Digital Equipment Corporation, Maynard, Ma. 01754 Order Code: DEC-10-XGSDA-A-D

digital equipment corporation · maynard. massachusetts

1st Edition March 1975

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this manual.

The software described in this document is furnished to the purchaser under a license for use on a single computer system and can be copied (with inclusion of DIGITAL's copyright notice) only for use in such system, except as may otherwise be provided in writing by DIGITAL.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

Copyright © 1975 by Digital Equipment Corporation

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

CDP	DIGITAL	KA10	QUICKPOINT
COMPUTER LAB	DNC	K110	RAD-8
COMSYST	EDGRIN	LAB-8	RSTS
COMTEX	EDUSYSTEM	LAB-8/e	RSX
DDT	FLIP CHIP	LAB-K	RTM
DEC	FOCAL	OMNIBUS	RT-11
DECCOMM	GLC-8	OS/8	SABR
DECSYSTEM-10	IDAC	PDP	TOPS10
DECTAPE	IDACS	PHA	TYPESET-8
DIBOL	INDAC	PS/8	TYPESET-10
			UNIBUS

DIVIDERS

**DIVIDER 1 - INTRODUCING YOURSELF TO THE DECsystem-10
SAYING GOODBYE TO THE DECsystem-10
TYPING AN EASY COMMAND**

**DIVIDER 2 - ENTERING YOUR PROGRAM INTO THE COMPUTER
USING THE SOS EDITOR
TYPING MISTAKES (OR FORGET I SAID THAT)
DELETE AN ENTIRE LINE
DELETE ONE OR MORE CHARACTERS
WHAT DID I TYPE**

**DIVIDER 3 - EXECUTING YOUR PROGRAM
STOPPING YOUR PROGRAM**

**DIVIDER 4 - EDITING YOUR PROGRAM
STARTING THE SOS EDITOR
PRINTING A LINE
INSERTING A LINE
DELETING A LINE
REPLACING A LINE
CHANGING A LINE
SAVING THE FILE**

**DIVIDER 5 - WORKING WITH FILES
LISTING THE NAMES OF YOUR FILES
DELETING A FILE
GETTING HELP FROM TOPS10
COPYING ANOTHER USER'S FILE
RUNNING ANOTHER USER'S PROGRAM**

**DIVIDER 6 - USING BASIC
STARTING BASIC
ENTERING YOUR PROGRAM
SAVING YOUR PROGRAM
RUNNING YOUR PROGRAM
EDITING YOUR PROGRAM
LISTING YOUR PROGRAM
RUNNING AN EXISTING PROGRAM
LEAVING BASIC AND LOGGING OFF
RETURN TO MONITOR MODE**

CONTENTS

	Page
Note to the Reader	v
Preface	vii
Getting Started with DECsystem-10	3
Project-Programmer Numbers	3
Passwords	3
Terminals	3
Introducing Yourself to DECsystem-10	4
Saying Goodbye to DECsystem-10	6
Using Two Easy Commands	6
Entering Your Program	11
Files and Programs	11
Using the SOS Editor	12
Correcting Typing Mistakes	15
Executing Your Program	19
Checking Your Program	20
Stopping Your Program	21
Editing Your Program	25
Correcting Your Program	26
Starting the SOS Editor	27
Printing a Line	27
Inserting a Line	28
Deleting a Line	28
Replacing a Line	28
Changing a Line	29
Saving the File	29
Re-executing Your Programs	29
Typing Out Your Program	30
Working with Files	35
Listing the Names of Your Files	35
Deleting a File	35
Getting HELP from TOPS10	36
Copying Another User's File	37
Renaming a File	37
Running Another User's Program	38
Using Basic	41
Starting Basic	41
Entering Your Program	41
Saving Your Program	43
Running Your Program	43
Editing Your Program	44
Replacing Your Program	44
Re-running Your Program	45
Listing Your Program	46
Running an Existing Program	47
Leaving BASIC and Logging Out	47
Appendix - TOPS-10 Commands	49

NOTE TO THE READER

GETTING STARTED WITH DECsystem-10 is a manual that introduces you to the DECsystem-10 and the TOPS-10 Operating System. This manual has colored dividers that mark its sections; each divider lists procedures for accomplishing specified tasks. For more detailed information, read the sections themselves; they describe the tasks more fully.

We hope that as you read this manual you are sitting at a computer terminal and executing each boxed example yourself. The adage "to do is to learn" applies especially to computers. The fastest—and the surest—way to learn TOPS-10 is to use it!

We use several conventions within this manual which may be foreign to you. Some of these conventions are defined below, others are defined within the text.

Convention	Meaning
	Whenever this symbol appears, we are indicating that you should press the key on the terminal keyboard labeled RETURN. This symbol is called a carriage return.
ESC	Whenever this symbol appears, we are indicating that you should press the key on the terminal keyboard labeled ESC. This key is called the ESCape key, and when it is pressed the system will print a \$. Note that on some terminals this key is labeled ALT (for ALternate) MODE or PREFIX.
•	This symbol is called a circumflex and is sometimes printed as ↑ (up-arrow). Whenever it appears, we are indicating that you should press the key on the terminal keyboard labeled CTRL. This key is called the ConTROl key.
	Whenever this symbol appears, we are indicating that you should press the key on the terminal keyboard labeled RUBOUT. On some terminal keyboards, this key is labeled DELETE. When this key is pressed, the most recently typed character is deleted.
—	Portions of an example which are underlined are the portions of the example that you must type. The portions not underlined are printed by the system.
UPPER CASE LETTERS	The portions of an example that are typed as uppercase letters are necessary portions of that example or command line. You must type the letters exactly as you see them in this manual. In the example: .EXECUTE filename.extension  you are to type the word in upper case letters exactly as it appears.

Convention**Meaning**

**lower
case
letters**

The portions of an example or command line description that are typed in lower case are variable portions that you must supply. The letters that you supply for a variable do not necessarily have to substitute letter-for-letter. To supply a variable for "name" you could use a word of n characters in length; you do not have to supply a word with exactly four characters. If at any time a character length limitation is in effect, we will tell you the legal range. In the example:

`.EXECUTE filename.extension` ↵

you are to type the word in uppercase letters exactly as it appears; you are to substitute a file name and a file name extension for the lower case letters.

△

Wherever this symbol appears within an example, we are indicating that you should press the space bar on the terminal keyboard.

TAB

Wherever this symbol appears, we are indicating that you should press the TAB key on the terminal keyboard. If the terminal you are using does not have a TAB key, press the CTRL key simultaneously with the I key (CTRL/I).

PREFACE

GETTING STARTED WITH DECsystem-10 is written for the person who has never used a computer before and for the person who has never used a DECsystem-10 before, but who has used other computer systems. Those persons who fall into the first category should read the entire manual; those falling into the second category might find that reading only the dividers is sufficient.

It is not necessary that other books or manuals be read before *GETTING STARTED WITH DECsystem-10*. However, upon finishing this manual some of you may wish to read *INTRODUCTION TO DECsystem-10 SOFTWARE* (DEC-10 MZDC-D), *DECsystem-10 OPERATING SYSTEM COMMANDS MANUAL* (DEC-10-OSMA-A-D), and *BEGINNER'S GUIDE TO MULTI-PROGRAM BATCH* (DEC-10-OMPBA-B-D).

INTRODUCING YOURSELF TO DECsystem-10

In order to begin using the system, do the following:

1. After you turn the terminal on, press the CTRL key. While it is still in the down position, press the C key. This is called typing a CTRL/C.
2. After you see the period (.), type the word LOGIN and press the RETURN key.
3. After you see the #, type your project-programmer number and press the RETURN key.
4. After you see the word PASSWORD:, type your password and press the RETURN key. Your password will not be printed.

Steps 1 through 4 above are called the logging-in procedure. Below is an example of the logging-in procedure.

```
.LOGIN )  
JOB 32      R5725D SYS #40/2 ITY73  
#27,4072 )  
PASSWORD: )  
1131      21-JAN-75      TUE
```

If the terminal you are using is not a local terminal (hooked directly to the computer), you must make connection with the computer through the use of a telephone. The procedure to establish this connection is:

1. Turn on the terminal.
2. Check the speed setting of the terminal (most terminals are labeled either 10 CPS, 15 CPS and 30 CPS or 110 baud, 150 baud, and 300 baud.) It should initially be set to either 10 CPS or 110 baud.
3. Dial the phone number to the computer.
4. Wait for a steady tone (sometimes this is a high pitched beep).
5. Lay the receiver in the slots provided for it either on the terminal or on the acoustic coupler.
6. Wait for the carrier detect light to come on.
7. If you see nothing printed on the terminal, type two CTRL/Cs until you see either a period or the words PLEASE LOGIN OR ATTACH.
8. Following the logging-in procedure listed above.

SAYING GOODBYE TO DECsystem-10

When you are finished using the system, do the following:

1. After you see the period, type K/F and press the RETURN key.

This procedure is called the logging-off procedure. Below is an example of the logging-off procedure.

```
.K/F )  
JOB 32, USER [27,4072] LOGGED OFF TTY73      1131  21-JAN-75  
SAVED ALL FILES (140 BLOCKS)  
RUNTIME 1.54 SEC
```

TYPING AN EASY COMMAND

To get today's date and time, type the word DAYTIME after you see the period. Press the RETURN key.

```
.DAYTIME )  
21-JAN-75 11:32:10
```

GETTING ACQUAINTED WITH DECsystem-10

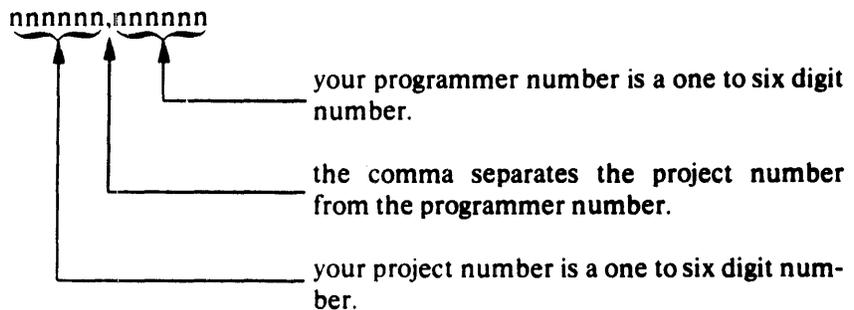
Before you can use the system, there are several questions for which you need the answers. The questions to ask someone at your installation are:

1. What project-programmer number should I use?
2. What terminal can I use?
3. Is there a document detailing the terminal's use? (If there is no such document, you can ask someone at your installation to assist you, or you can experiment with the terminal.)

Your project-programmer number and your password uniquely identify you to the system. A terminal is the piece of equipment that you use to communicate with the system.

Project-Programmer Numbers

A project-programmer number is written in the form:



The project-programmer number used as an example within this manual is 27,4072. Sometimes project-programmer numbers are typed within square brackets or parentheses, like [27,4072]. In later sections of this manual where square brackets are required, we will point out that fact. In some examples, a project-programmer number will be represented by the letters *ppn* for abbreviation purposes.

Passwords

A password allows you to access the system. This password must be typed in order for the system to be sure (1) that you are who you say you are and (2) that you are authorized to use the system. Each user has his/her own secret password. To keep these passwords secret, the password is not printed when you type it at the terminal.

Terminals

The third piece of information you need before using the system is information about the terminal you will be using. You should find out how to turn the terminal on and off. You should notice the positioning of some of the keys. Notice particularly, where the following keys are located:

the CTRL key

This key is used to type a control character and is further described below.

the RUBOUT key	This key is used to delete characters and is further described on page 10. (On some terminals this key is labeled DELETE.)
the SPACE bar	This bar, when pressed, moves the carriage over one space, creating a blank space.
the SHIFT key	This key, when pressed, allows you to type any character appearing on the upper half of a key.
the ESC key	This key is labeled on some terminals as ALT MODE or PREFIX.

When you are getting acquainted with the DECsystem-10, it is always good practice to use a hard-copy terminal—one that prints on paper everything that you and the system communicate to each other. If you desire, you can tear off the paper and save it for reference.

A video display terminal does not have a printing mechanism and does not use paper. Instead, it displays the conversation between you and the system on a television-like screen. Obviously, you cannot save this type of communication.

INTRODUCING YOURSELF TO DECsystem-10

Once you have turned on the terminal, press the CTRL key. While it is still in the down position, press the C key. This is called typing a CTRL/C and †C is printed at the terminal. CTRL/C is only one of several control characters that help you run the system.

- †C Type a CTRL/C.
- The system prints a period.

When you type a CTRL/C, you are interrupting the system, telling it to stop what it is doing and listen to you. When it is interrupted, the system prints a period.

The period is used by the system as a prompt character. A prompt character indicates to you that the system has finished what you last told it to do, and it is now ready for you to tell it to do something else.

NOTE

Sometimes two CTRL/Cs are needed to interrupt the system. If you type one or two CTRL/Cs and you do not get a period, ask another user who is familiar with the DECsystem-10 to help you.

After you see the . on your terminal, you have to identify yourself to the system. Type the word LOGIN directly after the period. Just as the system uses a period to tell you that it has finished a function, you must use a carriage return to tell it that you have finished a function. You tell it by pressing the RETURN key. (In this manual, a carriage return is indicated by ↵.)

The word LOGIN is a command word. The TOPS-10 Operating System recognizes a great many command words; within this manual we will discuss quite a few commands, but not all of them. The entire list of recognizable command words and their meanings can be found in Appendix A.

You type all command words to the system directly after the period prompt character. After you type these command words, you must press the RETURN key. With some command words the system expects you to supply an argument before you press the RETURN key.

command argument)

An argument supplies a quantity, time, location, device name, or other qualifying data that the system must know in order to perform your command.

After you type the word LOGIN and press the RETURN key, the system prints a message similar to this one:

```
.LOGIN)
JOB 23      R5725D SYS #40/2 TTY73
#
```

The first line of the message identifies you as being job number 23; you are using the monitor named R5725D; you are using system number 40/2; the terminal that you are using is called TTY73. The numbers and letters that appear on your terminal will resemble these, but they will not be identical. It is not necessary that you know what all of these numbers and letters mean, but you should know that they are printed by the system.

The second line of the system message is a number sign (#). It is printed to prompt you to type your project-programmer number. After typing your project-programmer number, press the RETURN key.

```
.LOGIN)
JOB 23      R5725D SYS #40/2 TTY73
#27,4072)
PASSWORD:)
```

The system prints the word PASSWORD followed by a colon. After the colon, you are to type your password. As you type each letter of your password, nothing is printed at the terminal; but the letters that you type are passed to the system even though they are not printed. After typing your password, press the RETURN key.

```
.LOGIN)
JOB 28      R5725D SYS #40/2 TTY73
#27,4072)
PASSWORD: )
1134      21-JAN-75      TUE
```

The system will print the message of the day. The message that you see may be somewhat longer than this, but its first line will resemble the above. If you do not see this message, but you see instead ?LGNIET INVALID ENTRY - TRY AGAIN, retype your project-programmer number and password. If you again receive the error message, check with someone at your installation.

SAYING GOODBYE TO DECsystem-10

When you are finished using the system, you cannot just stand up and walk away from your terminal. You must first type the characters K/F after the period. Then you must press the RETURN key. If you do not log-off the system in this manner, the terminal will not be free for the next user.

```
.K/F )  
JOB 28, USER (27,4072) LOGGED OFF TTY73      1134  21-JAN-75  
SAVED ALL FILES (140 BLOCKS)  
RUNTIME 1.47 SEC  
.
```

The K portion of the command terminates your communication with the system. The /F portion of the command is called a switch and indicates to the system that you want it to save any work that you did using the system. If you do not log-off the system in this manner, someone could come along and modify your files, copy your files, or do work on the system and charge it to your project-programmer number. After you press the RETURN key, the system prints a log-off message.

This message indicates that you just logged off the system. You were job number 28; you logged in with 27,4072 as your project-programmer number; the terminal you were using was labeled TTY73; you logged off on January 21, 1975. The second line of the log-off message indicates how many seconds the system actually worked for you.

To continue reading this manual and typing the examples which follow, you will have to log in again.

USING TWO EASY COMMANDS

Before you do anything with the system, try typing a few easy commands. There are several easy commands, but we will discuss only two of them here.

To have the system tell you the current date and time of day, type the word DAYTIME after you see the period. Press the RETURN key,

```
.DAYTIME )  
21-JAN-75 11:35:23
```

The diagram shows two arrows pointing from labels to the output of the DAYTIME command. One arrow points from the label "the current date" to the date "21-JAN-75". The other arrow points from the label "the current time" to the time "11:35:23".

The date is typed by the system in the form:

day of month - month - year

The time of day is typed by the system in the form:

hours : minutes : seconds

Hours are represented using a 24-hour clock. The time that the system typed in the preceding example 11:35:23 is 23 seconds after 11:35. Twelve noon is represented as 12:00; twelve midnight is represented as 00:00:00; three thirty in the morning is represented as 03:30:00.

By typing only three letters and pressing the RETURN key, it is possible to have the system type pages of information. After you see the period, type the letters SYSTAT and press the RETURN key.

```
.SYSTAT )
STATUS OF      R5725D SYS #40/2 AT 11:35:31 ON 21-JAN-75
UPTIME 4:30:08+4:28:28, 35+60%NULL TIME = 33+60%IDLE + 2+0%LOST
31 JOBS IN USE OUT OF 45. 31 LOGGED IN, 3 DETACHED OUT OF 45. (LOGMAX)

JOB   WHO      LINE#   WHAT   SIZE(K)  STATE  RUN TIME
1     [OPR]     DET    DAEMON  7+SPY    SL SW      30
2     [OPR]       1    FR5725  35      TI SW    16:54
3     [OPR]    PIJ14  SYSINF  8        HB SW      1
4     [OPR]     DET    SAVER   6+SPY    HB SW      1
5     60,60    131    DCTMH   16      TC SW      2
6     [OPR]    P0J14  LPTSPL  3+4      DI        4:01
7     TC
```

The information that the system typed for you is system status information concerning every person that is currently using the system. Within the first table that the system typed, look down the second column from the left, find the word [SELF] - that is you! It tells you your job number in the first column (the same number you got when you logged in.) Furthermore, it tells you that you are using the SYSTAT program, which is the name of the program that is printing all of this system status information for you.

In the example above, we stopped the system from printing information by typing a CTRL/O. You do not have to type a CTRL/O, you can wait until the system is finished typing. You can type a CTRL/O any time that the system is printing, and it will not harm anything.

ENTERING YOUR PROGRAM INTO THE COMPUTER

NOTE

If you want to use BASIC programming language, you do not have to read this section. Turn to the divider labeled USING BASIC to learn how to get your program into the computer.

Choose a name for the file which will contain your program. If the program you are going to put into the file is a FORTRAN program, add .FOR to the end of the name; if it is a COBOL program, add .CBL to the end of the name; if it is an ALGOL program, add .ALG to the end of the name. .FOR, .CBL, and .ALG are called file name extensions.

USING THE SOS EDITOR

1. After you see the period, type R SOS and press the RETURN key.
2. The SOS editor will print the word FILE followed by a colon. After the colon, type the name of your file, a period, and the file name extension. Press the RETURN key.
3. The system will print INPUT: followed by the name of your file.
4. The number 00100 is printed by the system. After this line number, type the first line of your program and press the RETURN key.
5. After you see 00200, which is the second line number, type the second line of your program, press the RETURN key.
6. Continue entering your program in this manner:
 - a. wait until a line number is printed,
 - b. type a line of your program, and
 - c. press the RETURN key.
7. Type the last line of your program and press the ESC key instead of the RETURN key.
8. After you see an asterisk (*), type the letter E and press the RETURN key.

TYPING MISTAKES (OR FORGET I SAID THAT)

On this divider, you will learn to correct mistakes made BEFORE pressing the RETURN key. To correct mistakes made AFTER pressing the RETURN key, see the section "Editing your Program" on pages 24 and 25.

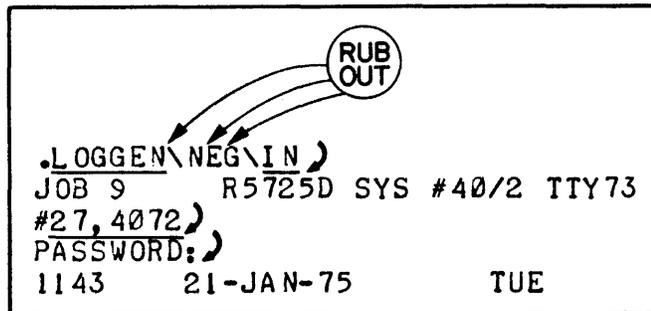
DELETE AN ENTIRE LINE

```
.LOGGEN↑U
LOGIN )
JOB 28      R5725D SYS #40/2 TTY73
#27,4072 )
PASSWORD: )
1137      21-JAN-75      TUE
```

A CTRL/U, printed as ↑U, erases an entire line. To type a CTRL/U, press the key labeled CTRL; while you hold it down, type a U.

DELETE ONE OR MORE CHARACTERS

```
.LOGGEN\NEG\IN )
JOB 9      R5725D SYS #40/2 TTY73
#27,4072 )
PASSWORD: )
1143      21-JAN-75      TUE
```



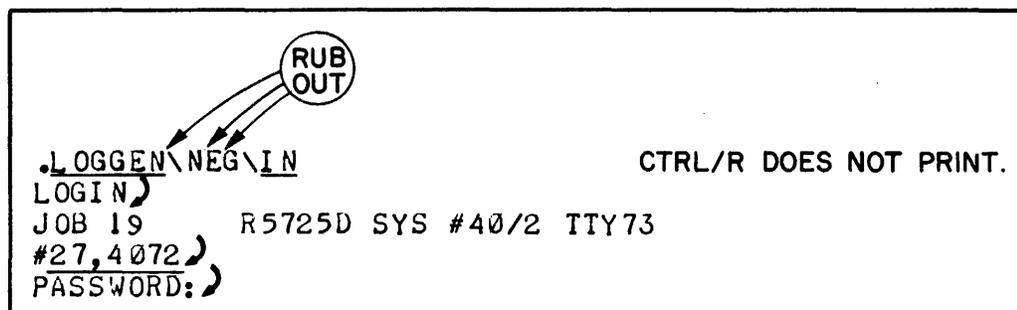
You can delete one or more characters by pressing the RUBOUT key once for each character you want to be deleted. Within this example, we wanted to delete G E N. In order to do so, we pressed the RUBOUT key three times, then we typed the correct letters (i.e., I N.)

WHAT DID I TYPE

To see the corrected line as the system sees it, you can type a CTRL/R before you press the RETURN key.

```
.LOGGEN\NEG\IN
LOGIN )
JOB 19      R5725D SYS #40/2 TTY73
#27,4072 )
PASSWORD: )
```

CTRL/R DOES NOT PRINT.



If the line is as you wish it, press the RETURN key after the system prints the corrected line.

ENTERING YOUR PROGRAM

Files and Programs

Before you enter a program into the computer, it is wise to know the definition of the word program and how it is stored in the computer.

In the first portion of this manual, you typed several TOPS-10 Commands: LOGIN, DAYTIME, and SYSTAT. When you pressed the RETURN key, the system executed the command.

A program is executed also, but in a slightly different manner. A program is made up of statements. Each statement tells the system to do something, but that something is not done when you press the RETURN key. It is done after:

1. You have typed all statements in the program into a file and after,
2. You have told the system to translate all of the statements in the program to machine language.

If the system is to remember and later execute your program, it must be told to enter the program into a file. A file is an ordered set of data that is named and then stored in the system. Usually your file and program have the same name. For instance, if you had a program named SORT, it could be stored in a file named SORT. You could also store the program SORT in a file called WONDER or any other file you wish. (The only restriction is that the file name must not be longer than six characters.)

Always keep in mind that you are responsible for how much of your work the system remembers and for how much it forgets. If, for example, you tell the system to store SORT for you, it will always be there when you want it; but if you tell the system to delete SORT it will be erased and gone forever. If you try to summon it, the system will act as though it has never heard of it. You will learn later that you can perform a bit of "black magic" to retrieve these lost files, but we will not discuss this in this manual. (For further information refer to the DECsystem-10 Operating System Commands Manual, the FAILSAFE program.)

You can use the DECsystem-10 to solve a problem. Your problem will usually be represented as a program written in a programming language such as ALGOL, BASIC, or FORTRAN. Before you can tell the system to execute your program, you must enter your program into a file. To accomplish this, you are assisted by a program that is called an editor. The name of the editor used in this manual is SOS.

Since you can have more than one file stored within the system, you must decide on a name for each of your files. This name will uniquely identify each of your files from one another.

The name can be anything that you want it to be as long as it is not longer than six characters. The extension to that file name usually represents the programming language in which you typed your program, but it can be any four characters you wish as long as the first character is always a period. You can use one of the following extensions.

EXTENSION	TYPE OF PROGRAM
.ALG	an ALGOL program
.BAS	a BASIC program
.CBL	a COBOL program
.FOR	a FORTRAN program
.MAC	a MACRO program

NOTE

If you are using BASIC, you do not have to use the SOS editor; you can type a program directly to BASIC. Turn to the divider labeled USING BASIC for instructions.

USING THE SOS EDITOR

To find the sum of two numbers, we will use two programs: one written in FORTRAN and the other in ALGOL; either one solves the problem. You may type the one you choose. If you wish, you may try both.

Whatever your choice of language, you will be using an editor for assistance in typing the program. To request the SOS editor's services, type R, a space, and then the command word SOS. Finally, press the RETURN key.

.R SOS ↵

The SOS editor will respond by typing the word FILE followed by a colon. After the colon, type the name and extension of your file. Press the RETURN key.

.R SOS ↵

FILE: PROG1.FOR ↵

The SOS editor will type the word INPUT: followed by the name of your file.

.R SOS ↵

FILE: PROG1.FOR ↵
 INPUT: PROG1.FOR
 00100

This line indicates to you that the SOS editor has received your request for assistance, and it is ready to start helping you with the file named PROG1.FOR.

HINT

If SOS responds with EDIT: instead of INPUT:, it means that you already have a file with that name and extension. If this happens, type EQ and press the RETURN key. Start the edit process over again by typing R SOS. This time when the SOS editor requests the name of your file and the file name extension, type in a file name different from the one you typed before. You can use the same file name extension if you desire.

Directly after the SOS editor types INPUT: PROG1.FOR, it will print 00100. For example:

```
R SOS
FILE: PROG1.FOR
INPUT: PROG1.FOR
00100
```

00100 is called a line number. There is a line number for each line of your program starting with 00100 and increasing by 100 to the end of your program. Line numbers go by hundreds because later you may want to insert new lines without altering those lines you have already typed.

After the SOS editor prints the first line number of your program, type the first line of your program and press the RETURN key. The SOS editor will automatically print the next line number, after which you type the next line of your program and press the RETURN key. You should continue doing this until you type your entire program. After you type the last line of your program, press the ESC key and not the RETURN key.

Please type the problems exactly as you see them in this manual, check your lines carefully before you press the RETURN key. Correct any mistakes by using the RUBOUT key or the CTRL/U (you should not retype the line numbers) feature as explained on the previous divider. (Do not press CTRL/C). If you make a typing mistake, but you do not discover it until after you have pressed the RETURN key, just keep right on typing—we will show you how to correct the mistakes a little bit later. When typing the program below, you can type a tab to space the carriage over eight spaces. To type a tab, you press the key labeled CTRL and at the same time press the I key. This is called typing a CTRL/I which results in a tabulation.

NOTE

On some terminals there is a key labeled TAB, which is usually on the left side of the keyboard. You can press the TAB key to get the same effect as typing a CTRL/I.

Here is a FORTRAN program that solves our addition problem.

```

.R SOS )
FILE: PROG1.FOR )
INPUT: PROG1.FOR
00100 (TAB) TYPE(1,1)
00200 10 (TAB) FORMAT( 'WHAT ARE THE TWO NUMBERS? ')
00300 (TAB) ACCEPT(20, X, Y)
00400 20 (TAB) FORMAT(2F)
00500 (TAB) ANSWER=X+Y
00600 (TAB) TYPE(3,1, ANSWER)
00700 30 (TAB) FORMAT( 'THE SUM OF THE NUMBERS IS ', F7.2 )
00800 (TAB) ENDS
*ES )
      USER PRESSES
      THE ESC KEY.
[DSKC: PROG1.FOR ]

```

If you realize that you have made a typing mistake, turn to page 23 to learn how to correct your program. If you are not aware of any mistakes, turn to page 17 to learn how to execute your program.

Here is an ALGOL program that solves our addition problem.

```

.R SOS )
FILE: PROG2.ALG )
INPUT: PROG2.ALG
00100 BEGIN (TAB) INTEGER S, C, A;
00200 | THIS PROGRAM WILL ADD TWO NUMBERS AND;
00300 | PRINT THE RESULT;
00400 (TAB) WRITE( " (2C) WHAT ARE THE NUMBERS TO ADD (1C) " );
00500 (TAB) READ( S );
00600 (TAB) READ( C );
00700 (TAB) A:=C+S;
00800 (TAB) WRITE( " (2C) THE SUM OF " );
00900 (TAB) PRINT( S, 3, 3 );
01000 (TAB) WRITE( " AND " );
01100 (TAB) PRINT( C, 3, 3 );
01200 (TAB) WRITE( " IS: " );
01300 (TAB) PRINT( A, 3, 3 );
01400 (TAB) WRITE( " (2C) " );
01500 ENDS
*E )
      USER PRESSES THE ESC KEY.
[DSKC: PROG2.ALG ]

```

If you realize that you have made a typing mistake, turn to page 23 to learn how to correct your program. If you are not aware of any mistakes, turn to page 17 to learn how to execute your program.

CORRECTING TYPING MISTAKES

Use the RUBOUT key to erase one or more characters in the line you are currently typing. Each time you press this key, the last character you typed is erased. The system responds by typing a backslash when you press the RUBOUT key for the first time. After the backslash the system will print each character that you deleted. When you have finished deleting and you start typing, the system will print another backslash. In other words, the system will print a backslash on either side of the character(s) you deleted.

NOTE

On some terminals, the RUBOUT key may be labeled DELETE.

Consider the following:



00600 TY Y \ Y \ PE 30, ANSWER

When typing the word TYPE, you hit the Y key twice by mistake. Since you noticed your mistake the moment that you made the mistake, you were able to utilize the RUBOUT key, pressing it once to delete one character. The system printed a backslash followed by the letter you deleted (which was Y). After you typed the P, the system printed another backslash indicating the end of deletion, and it printed the character that you just typed.



00700 PRINT "COMPUTERS ARE" \ "ERA SRA \ERS ARE"

In line 700, after you typed the second quotation mark, you noticed that you had misspelled the word COMPUTERS. You deleted eight characters (a space and a TAB count as one character) by pressing the RUBOUT key eight times. The system printed a backslash on either side of the eight deleted characters that you deleted. You continued typing the line, before the system printed the first character you typed, it printed the second backslash indicating the end of the deletion process.

You should remember several things when you use the RUBOUT key. First, you cannot correct characters on a line once you have pressed the RETURN key. Second, in addition to correcting

lines in a program, you can use the RUBOUT key to correct errors you make when typing commands. For example, if you mistype the LOGIN command, and you notice the error before you hit the RETURN key, you can use the RUBOUT key to correct your error.

NOTE

If you do not notice your mistake while typing a command, and you press the RETURN key, you will usually receive an error message.

```
.LOGGEN )  
?LOGGEN?
```

.

?LOGGEN? is an error message; the system is telling you that it did not understand what you told it to do. After the error message, the system prints a period after which you can type the correct command.

EXECUTING YOUR PROGRAM

To execute your program, follow the procedure listed below:

1. Type the command word EXECUTE after the period prompt character.
2. Follow the command word EXECUTE with a space, the name of your file, a period, and your file's file name extension.
3. Press the key labeled RETURN.

If your program did not execute properly and you received an error message, turn to the divider labeled EDITING YOUR PROGRAM. Some program errors are caused by typing mistakes.

STOPPING YOUR PROGRAM

To stop your program during its execution, follow the procedure listed below:

1. Press the key on the terminal keyboard labeled CTRL.
2. While the key labeled CTRL is in the down position, press the C key twice.

The above procedure produces a double CTRL/C and stops the execution of your program.

EXECUTING YOUR PROGRAM

Now that your program has been typed into a file, it must be translated by the appropriate language translator (compiler) into a machine language program. After this translation, the program can be executed.

Before your program is translated, it is called a source language program, and the file containing it has an extension indicating the language in which the program is written (e.g., .FOR indicates a FORTRAN program.) After your program is translated, a new file containing the translation of your program is created. The name of this file is the same as the name of the untranslated file, but this file has an extension of .REL (indicating a machine language program.)

In order to translate and execute your program, type the command word EXECUTE followed by the name and extension of your file. Press the RETURN key.

```
.EXECUTE filename.extension ↵
```

Note that the extension .FOR causes the FORTRAN compiler to translate your program, and .ALG causes the ALGOL compiler to translate your program.

EXECUTING YOUR FORTRAN PROGRAM

<pre>.EXECUTE PROG1.FOR ↵ FORTRAN: PROG1 MAIN. LINK: LOADING [LNKXCT PROG1 EXECUTION] WHAT ARE THE TWO NUMBERS? 23△6789 ↵ THE SUM OF THE NUMBERS IS 6812.00 END OF EXECUTION CPU TIME: 0.09 ELAPSED TIME: 11.77 EXIT</pre>	<p>FORTTRAN translates the file. The file is loaded.</p> <p>Execution begins.</p>
--	---

EXECUTING YOUR ALGOL PROGRAM

```
.EXECUTE PROG2.ALG )
```

```
ALGOL: PROG2
```

```
LINK: LOADING
```

```
[LNKXCT PROG2 EXECUTION]
```

```
ALGOL translates the file.
```

```
The file is loaded.
```

```
Execution begins.
```

```
WHAT ARE THE NUMBERS TO ADD
```

```
23 )  
345 )
```

```
THE SUM OF 23.000 AND 345.000 IS: 368.000
```

```
END OF EXECUTION - 2K CORE
```

```
EXECUTION TIME: 0.07 SECS.
```

```
ELAPSED TIME: 14.68 SECS.
```

```
.
```

CHECKING YOUR PROGRAM

If you are curious and want to know how your program is doing, you can check its progress even when it is in the middle of execution. To do this, press the CTRL key and, while it is still in the down position, press the T key. This process is called typing a CTRL/T. The response from a CTRL/T shows the status of your program, the amount of machine time that you have used, and the amount of time since you last typed a CTRL/T.

```
DAY: :3:56 RUN: 15.61 RD:1505 WR:250 PROG1 2K+5K RN PC:400230
```

The response from the CTRL/T indicates that you typed a CTRL/T (which does not produce a printed character) while PROG1.FOR was running. Your program was in the RUN (RN) state and you typed a CTRL/T 3 minutes and 56 seconds earlier.

Typing a CTRL/T does not interfere with the running of your program in any way. However, if your program is typing information on your terminal at the same time that you type a CTRL/T, the response from CTRL/T will be mixed in with the information from your program.

```
.LOGIN )
```

```
JOB 4 R5725D SYS #40/2 TTY73
```

```
#DAY: 0.80 RUN: 0.16 RD:5 WR:0 LOGIN 2K+11K SL PC:405365
```

```
27,4072 )
```

```
PASSWORD: )
```

```
1213 21-JAN-75
```

```
TUE
```

STOPPING YOUR PROGRAM

Sometimes unexpected things happen in a program, and it may never complete execution. For instance:

- The program may be written to get information from another file; during its execution, however, you remember that the other file does not exist.

-OR-

- You may decide that you would like to stop the execution of your program because you want to do something else.

No matter what your reason is for stopping your program, you follow the same procedure: press the CTRL key; while it is still in the down position, press the C key twice. You will then see the period (.) prompt character which means that your program has stopped execution and that you can now type any TOPS-10 command you wish.

```
.EXECUTE PROG1.FOR
LINK:   LOADING
[LNKXCT PROG1 ETC
.
```


EDITING YOUR PROGRAM

STARTING THE SOS EDITOR

To use the SOS editor in order to change your program, do the following:

1. After you see the period (.) prompt character, type the letter R, a space, and the command word SQS. Press the RETURN key.
2. After you see the word FILE:, type the name and extension of the file that contains the program you want to change. Press the RETURN key. You should see EDIT:filename.extension, where filename.extension is the name and extension of your file. The name and extension of your file are followed by a period prompt character.

NOTE

If you see INPUT:filename.ext instead of EDIT:filename.ext, press the ESC key. After you see the asterisk (*), type EG and press the RETURN key.

Check the file name that you typed. If you find that you mistyped it, start again at step 1.

If both the name and the extension were typed correctly, perhaps the file has a name different from the one you remember; or perhaps the file has been deleted. Type the word DIRECT immediately after the period and press the RETURN key. The DIRECT command will print on your terminal a list of all of the files within your disk area. If your file is not listed, you will have to reenter your program into a file.

After you see the asterisk (*), use any of the commands below to change your file.

PRINTING A LINE

To print a line of your file, type P and the number of the line you want to be printed. Press the RETURN key.

```
*P1300 )  
01300      C = SQRT (A ++2 +
```

INSERTING A LINE

To insert a new line in your file, type the letter I and the line number you want the new line to have. Press the RETURN key. After the SOS editor prints a line number, type the new line of your program. Press the RETURN key.

```
*I1350 )  
01350      C = SQRT (A**2 + B**2)
```

DELETING A LINE

To delete a line in your file, type the letter D and the number of the line you want deleted. Press the RETURN key.

```
*D1300 )  
1 LINES (01300/1) DELETED
```

REPLACING A LINE

To automatically delete a line in your file and insert a new line in its place, type the letter R followed by the number of the line you want to replace. Press the RETURN key. After the SOS Editor prints the line number, type the replacement line and press the RETURN key.

```
*R1400 )  
01400 (TAB) THIS LINE HAS BEEN REPLACED )  
1 LINES (01400/1) DELETED
```

CHANGING A LINE

To replace an existing group of characters within a line with a new group of characters, type the letter S and the existing characters; press the ESC key. Type the new characters; press the ESC key. Type the number of the line that you are changing; press the RETURN key.

```
*P1600 )  
01600 THIS IS A COM COMMENT  
*SΔC O M Δ $ Δ $ 1 6 0 0 )  
01600 THIS IS A COMMENT
```

SAVING THE FILE

To finish using the SOS editor and save the edited file, type E and press the RETURN key.

```
*E )  
[DSKC: TEST3 ]  
.
```

EDITING YOUR PROGRAM

Once you have typed your program into a file, you can use the SOS editor to eliminate any errors in it. This section shows you the entire process of editing, re-executing, and typing out a FORTRAN program.

If you typed either PROG1.FOR or PROG2.ALG incorrectly (and consequently, the program would not execute properly), turn to page 27 for directions on editing those errors out of your program. After editing your program to eliminate any errors, type:

```
.EXECUTE programname.extension)
```

CHANGING YOUR FORTRAN PROGRAM

If your FORTRAN/ALGOL program is typed correctly, you can change it. Type everything that is underlined in the example below to make your program called PROG1.FOR add three numbers.

```
.R SOS )  
  
FILE: PROG1.FOR )  
EDIT: PROG1.FOR  
*P200 )  
00200 10 FORMAT ( ' WHAT ARE THE TWO NUMBERS? ' )  
*SIWOSTHREES$200 )  
00200 10 FORMAT ( ' WHAT ARE THE THREE NUMBERS? ' )  
*P300  
00300 ACCEPT 20, X, Y  
*S,ΔY$,ΔY,ΔZ $300  
00300 ACCEPT 20, X, Y, Z  
*P500 )  
00500 ANSWER=X+Y  
*SY$Y+Z $500  
00500 ANSWER=X+Y+Z  
*E )  
  
[ DSKC: PROG1.FOR [ 27, 4072 ] ]  
  
.
```

To see what the changed program looks like, type the following line after the period prompt character:

```
.TYPE PROG1.FOR )
```

CHANGING YOUR ALGOL PROGRAM

Type everything that is underlined in the example below to make your program called PROG2.ALG add three numbers.

```

.R SOS )

FILE: PROG2.ALG )
EDIT: PROG2.ALG
*P100 )
00100 BEGIN INTEGER S, C, A;
*SASB,ΔAS100 )
00100 BEGIN INTEGER S, C, B, A;
*P200 )
00200 !THIS PROGRAM WILL ADD TWO NUMBERS AND:
*STWQ$THREE$200 )
00200 !THIS PROGRAM WILL ADD THREE NUMBERS AND:
*I650 )
00650 (TAB) READ (3); )
*P700 )
00700 A: =C+S;
*S: S+B; $700 )
00700 A: =C+S+B;
*I1150 )
01150 (TAB) WRITE ("AND "); )
*I1160 )
01160 (TAB) PRINT (B,3,3); )
*E )

Press the "ESC" key.

[DSKC: PROG2.ALG[27,4072 ] ]
.

```

To see what the changed program looks like, type the following line after the period prompt character:

.TYPE PROG2.ALG)

Now that your changed program is in the file, run it to find any errors. Type EXECUTE and the name and extension of the file containing your program; press the RETURN key.

Note that the extension .FOR causes the FORTRAN compiler to translate your program; the extension .ALG causes the ALGOL compiler to translate your program.

If your program did not execute properly because of errors, read the following section for help. If your program is exactly like the FORTRAN and/or ALGOL programs in this manual and will not execute properly, ask someone familiar with FORTRAN and/or ALGOL to check it over.

CORRECTING YOUR PROGRAMS

There may be several typing errors in your program. You can eliminate these errors with the help of the SOS editor.

STARTING THE SOS EDITOR

After you see the period prompt character, type R, a space, and the word SOS; press the RETURN key.

```
.R SOS )
```

After you see the word FILE:, type the name of the file to be edited. Follow the file name by a period, then by the extension. Press the RETURN key; the SOS editor will print the word EDIT: followed by the file name and extension; and on the next line it will print an asterisk. For this example, we are using the file name PROG2 with the extension .ALG.

```
.R SOS )
```

```
FILE: PROG2.ALG )  
EDIT: PROG2.ALG  
*
```

NOTE

If you see INPUT: instead of EDIT:, press the ESC key. After you see the asterisk (*), type EQ and press the RETURN key.

Check the name and extension that you typed. If either was mistyped, you can start the procedure at step 1.

If neither the name nor the extension was mistyped, perhaps the file has a name different from the name you remember; or perhaps the file had been deleted.

Type the word DIRECT immediately after the period and press the RETURN key. The DIRECT command will list the names of the files within your disk area. If your file is not listed, you will have to retype your program into a file.

After you see the asterisk, you can use any of the following commands to edit your file.

PRINTING A LINE

Print the line in which you think that there is an error. To print a line, type the letter P directly after the asterisk prompt character followed by the number of the line you want printed. Press the RETURN key. The command P1300 prints the contents of the line numbered 1300.

```
*P1300 )  
01300 BEGIN INTEGER S,C;
```

In line number 100, the integer A was omitted between the comma and the semicolon.

To print a group of lines, type the letter P directly after the asterisk prompt character. Follow the letter P with the number of the first line to be printed, a colon, and the number of the last line to be printed. Press the RETURN key. The command P500:700 prints the lines 500 through 700.

```

*P500;700 )
00500      RED (S);
00600      READ READER (C);
00700      A := C + C;

```

We see several errors: an A has to be inserted into line number 500; in line number 700 the second C must be changed to an S; line number 600 has two READs. Refer to the section CHANGING A LINE to correct the error in line 500. Refer to the section REPLACING A LINE to correct the error in line numbered 600. Refer to the next section, INSERTING A LINE, to correct the error in line number 700.

INSERTING A LINE

To insert a line into your file, type the letter I followed by the number of the line you want to insert. Press the key labeled RETURN. The command inserts the line 750. After SOS types the new line number, type the line to be inserted; press the key labeled RETURN.

```

*I750 )
00750      A := C + S;

```

Under some circumstances SOS types another line number after you press the key labeled RETURN. When this happens, press the key labeled ESC; SOS will then print an asterisk.

DELETING A LINE

Line 700 should now be removed from your file, since line number 750 has been inserted into the file. To delete a line in your file, type the letter D followed by the line number of the line you wish to be deleted. Press the key labeled RETURN. The command D700 will delete line number 700 from your file.

```

*D700 )
1 LINES (00700/1) DELETED

```

REPLACING A LINE

In line 600, our error was typing READ twice. The R (for replace) command deletes the line and automatically accepts text for replacement.

To replace a line within your file, type the letter R followed by the number of the line you want to replace. Press the key labeled RETURN. After SOS types the line number, type the new contents of the line and press the key labeled RETURN. The command R600 replaces the line 600.

```

*R600 )
00600      READ (C);
1 LINES (00600/1) DELETED

```

CHANGING A LINE

In line number 500, the letter A has to be inserted between the letters E and D. To replace existing characters within a line with a new group of characters—(1)Type the letter S followed by the characters you wish to be replaced; (2)Press the ESC key; (3)Type the replacement characters; (4)Press the ESC key; (5)Type the number of the line that you want to change; (6)Press the RETURN key. The command SED\$EAD\$500 replaces the existing characters ED within line 500 with new characters EAD.

```
*P500 )  
00500          RED (S);  
*SED$EAD$500 )  
00500          READ (S);
```

SAVING THE FILE

To finish using the SOS editor and to save the edited version of the file, type the letter E and press the RETURN key.

```
*E )           The system types the name of your  
              file.  
[DSKC: TEST6 ]
```

RE-EXECUTING YOUR PROGRAMS

To execute your FORTRAN program a second time, follow the example below.

```
.EXECUTE PROG1.FOR )  
FORTRAN: PROG1  
MAIN.  
LINK:   LOADING  
[LNKXCT PROG1 EXECUTION]  
WHAT ARE THE THREE NUMBERS?  
234567 )  
345 )  
THE SUM OF THE NUMBERS IS 1146.00  
  
END OF EXECUTION  
CPU TIME: 0.09  ELAPSED TIME: 16.22  
EXIT
```

To execute your ALGOL program a second time, see the following example.

```

.EXECUTE PROG2.ALG )
ALGOL: PROG2
LINK:   LOADING
(LNKXCT PROG2 EXECUTION)

WHAT ARE THE NUMBERS TO ADD
2345 )
654 )
3456 )

THE SUM OF 2345.000 AND 654.000 AND 3456.000 IS: 6455.000

END OF EXECUTION - 2K CORE
EXECUTION TIME: 0.08 SECS.
ELAPSED TIME: 16.62 SECS.

```

Type two CTRL/Cs to stop the execution of your program.

TYPING OUT YOUR PROGRAM

To get a final copy of your FORTRAN program, type the word TYPE after the period prompt character. Follow the word TYPE with one space, followed by the file name and extension of your program. Press the RETURN key.

```

.TYPE PROG1.FOR )
00100      TYPE 10
00200      10      FORMAT ( ' WHAT ARE THE THREE NUMBERS? ' )
00300      ACCEPT 20, X, Y, Z
00400      20      FORMAT (2F)
00500      ANSWER=X+Y+Z
00600      TYPE 30, ANSWER
00700      30      FORMAT ( ' THE SUM OF THE NUMBERS IS ',F7.2)
00800      END

```

```

.TYPE PROG2.ALG )
00100 BEGIN INTEGER S, C, B, A;
00200 !THIS PROGRAM WILL ADD THREE NUMBERS AND;
00300 !PRINT THE RESULT;
00400 WRITE (" [2C] WHAT ARE THE NUMBERS TO ADD [1C] ");
00500 READ (S);
00600 READ (C);
00650 READ (B);
00700 A:=C+S+B;
00800 WRITE (" [2C] THE SUM OF ");
00900 PRINT (S,3,3);
01000 WRITE (" AND ");
01100 PRINT (C,3,3);
01150 WRITE (" AND ");
01160 PRINT (B,3,3);
01200 WRITE (" IS: ");
01300 PRINT (A,3,3);
01400 WRITE (" [2C] ");
01500 END

```


WORKING WITH FILES

LISTING THE NAMES OF YOUR FILES

To obtain a list of your file names, do the following:

1. After you see the period prompt character, type the word **DIRECT** and press the key labeled **RETURN**.

The names of each file, along with some other information, is printed on your terminal. The following example is typical of a TOPS-10 response to a direct command.

```
.DIRECT )  
PROG1  REL      2 <055>  21-JAN-75  DSKC:  [27,4072 ]  
TEST1           1 <055>  21-JAN-75  
TEST2           1 <055>  21-JAN-75  
PROG1  FOR      1 <055>  21-JAN-75  
PROG2  ALG      1 <055>  21-JAN-75  
PROG2  REL      2 <055>  21-JAN-75  
TOTAL OF 8 BLOCKS IN 6 FILES ON DSKC: [27,4072 ]  
.
```

DELETING A FILE

To remove a file that you no longer want in your disk area, do the following:

1. After the period prompt character, type the word **DELETE**. Type a space followed by the name of the file you wish to delete. Press the **RETURN** key.

The system will print the name(s) of the file(s) it deleted and the number of blocks in your directory which were freed by this deletion. The following example is typical of a TOPS-10 response to a **DELETE** command.

```
.DELETE TEST1 )  
FILES DELETED:  
TEST1  
01 BLOCKS FREED  
.
```

GETTING HELP FROM TOPS-10

To have the system help you with programs follow the procedure below:

1. Type the word HELP after the period prompt character.
2. Type a space followed by an asterisk (*).
3. Press the RETURN key.

The system will print a list of all program names with which it can help you. Look down the list, find the program with which you want help, and follow the procedure below.

1. Type the word HELP followed by a space.
2. Type the program name with which you want help.
3. Press the key labeled RETURN.

COPYING ANOTHER USER'S FILE

You can copy a file from another user's directory into your directory by following the procedure below:

1. Type the word COPY followed by a space.
2. Type the file name and extension of the file which will be put into your area (the destination file.)
3. Type an equal sign.
4. Type the file name and extension of the other user's file.
5. Type a space followed by the other user's project-programmer number in square brackets.
6. Press the RETURN key.

RUNNING ANOTHER USER'S PROGRAM

To execute a program which resides in another user's directory area, follow the procedure below.

1. Type the word RUN followed by a space.
2. Type the name and extension of the other user's file.
3. Type the other user's project-programmer number in square brackets.
4. Press the key labeled RETURN.

WORKING WITH FILES

You have an area on the disk that is uniquely yours - a space reserved for the storage of your files. Identified by your project-programmer number, this area is known as your directory, and it contains all the information about the files you create.

LISTING THE NAMES OF YOUR FILES

Now that your programs have been entered into the computer and executed, you have four files in your disk area. To obtain a listing of these file names, type the word `DIRECT` immediately following the period prompt character; press the key labeled `RETURN`.

```
.DIRECT )
PROG1  REL      2  <055>  21-JAN-75  DSKC:  [27,4072 ]
TEST2
PROG1  FOR      1  <055>  21-JAN-75
PROG2  ALG      1  <055>  21-JAN-75
PROG2  REL      2  <055>  21-JAN-75
      TOTAL OF 7 BLOCKS IN 5 FILES ON DSKC: [27,4072 ]
.
```

When you edit your file for the first time, SOS changes your original, unedited file into a backup file. In this change process, SOS will change the file's extension to `.Qxx`, where `xx` are the last two letters of the file's original extension. SOS creates this backup file so that you will always have an unaltered copy of your original file.

The `DIRECT` command output also includes the number of blocks occupied by the file, and its creation date. In the upper-right corner of the printout, you will find the name of the disk storing your files along with your project-programmer number.

DELETING A FILE

If there is a file in your directory that you no longer need, you may delete it. To delete one or more files you use the `DELETE` command. For example, to delete the file called `TEST2`, you can issue the following command:

```
.DELETE TEST2 )
FILES DELETED:
TEST2
01 BLOCKS FREED
.
```

The system types the words FILES DELETED: in response to the DELETE command, and it then lists the name(s) of the file(s) it deletes from your area. The system prints a line telling you how many blocks were freed in your directory because of this deletion.

When the .DIRECT command is issued again, you can see that TEST2 has been deleted from your directory.

```
.DIRECT )
PROG1 REL 2 <055> 21-JAN-75 DSKC: [27,4072]
PROG1 FOR 1 <055> 21-JAN-75
PROG2 ALG 1 <055> 21-JAN-75
PROG2 REL 2 <055> 21-JAN-75
TOTAL OF 6 BLOCKS IN 4 FILES ON DSKC: [27,4072]
```

GETTING HELP FROM TOPS-10

The system can print out a list of various data which can help you in your programming tasks. You can have this information printed by using the HELP command. To get a list of the available subjects with which TOPS-10 can help you, type the command word HELP, a space, and an asterisk after the period prompt character.

```
.HELP * )
HELP IS AVAILABLE FOR THE FOLLOWING:
ALCOMP ALLOT ALQUIK BACKUP BASIC BLISW BOOT11 BOOTGT
BSC CDP SPL CDRSTK CHANGE COBDDT COBOL COBRG CREDIR
CREF DDT DELFIL DIRECT DISMOU DSKRAT FAILSA FGEN
FILCOM FILE FORDDT FORTRA FRS FUDGE2 GASP
GRIPE HELP INI INITHA ISAM
LINK LOGIN LOGOUT
:
```

The system will print a list of all the programs with which it can help you. Find on this list the program on which you would like assistance, then type:

HELP name)

where name is the command or program name you have chosen from the list typed by the HELP * command.

.HELP LOGIN)

COMMAND FORMAT:

LOGIN PROJ, PROG /SWITCH/SWITCH

WHERE SWITCHES ARE:

CORE: N SET THE CORE LIMIT TO N K.

DSKFUL: PAUSE PRINT A MESSAGE AND ALLOW A CONTINUE
WHEN THE DSK GETS FULL.

DSKFUL: ERROR IF THE DISK FILL UP GIVE THE USER
PROGRAM AN OUTPUT ERROR. (DEFAULT)

DSKPRI: N SET THE DISK PRIORITY TO N. N+0

COPYING ANOTHER USER'S FILE

You can copy one of your files into another file in your area. You do this by typing the command word COPY, the name of the new file, an equal sign, and the name of the file being copied.

.COPY NEW = OLD)

You can also copy a file from another user's directory into your own. You do this by typing the command word COPY, the name of the file to be in your area, an equal sign, the name of the other user's file, and his project-programmer number.

.COPY MINE = GRADES.FOR[27,4020])

Some of the files cannot be copied; if this is the case, you will receive an error message.

RENAMING A FILE

You can rename any file in your directory. You do this by typing the command RENAME followed by the new file name, an equal sign, and the old file name.

```
.RENAME TESTER = MINE )  
FILES RENAMED:  
MI NE
```

.

RUNNING ANOTHER USER'S PROGRAM

You can execute a program that resides in another user's directory area. You can do this by typing the line:

RUN progname [ppn]

where progname is the file name and file name extension of the other user's program. [ppn] is the other user's project-programmer number.

USING BASIC

If you want to run a BASIC program, you can enter it directly to BASIC; you do not have to use any Editor.

STARTING BASIC

After you see the period prompt character, type R BASIC and press the RETURN key.

```
.R BASIC ↵
```

```
READY, FOR HELP TYPE HELP.
```

ENTERING YOUR PROGRAM

If you want to enter a new BASIC program, do the following:

1. After you see the words READY, FOR HELP TYPE HELP, type the command word NEW and press the RETURN key.
2. After you see the words

```
NEW FILE NAME--
```

type a name for your program and press the RETURN key.

3. After you see the word READY, begin typing in your program. Start each line with a line number, type the contents of the line, and press the RETURN key at the end of each line.

```
NEW ↵  
NEW FILE NAME--ADD1.BAS ↵
```

```
READY
```

SAVING YOUR PROGRAM

Once you have finished entering your program, type the word SAVE and press the RETURN key.

```
SAVE ↵
```

```
READY
```

RUNNING YOUR PROGRAM

To run your program, type the word RUN and press the RETURN key.

```
READY  
RUN ↵
```

```
ADD1          11:51          24-JAN-75
```

```
⋮
```

```
TIME: 0.23 SECS.
```

```
READY
```

EDITING YOUR PROGRAM

To edit your BASIC program, type the number of the line you want to change. Then type the new contents of that line and press the RETURN key.

```
260 INPUT )  
355 PRINT )
```

REPLACING YOUR PROGRAM

After you have finished editing your program, type the word REPLACE and press the RETURN key.

```
REPLACE )  
READY
```

LISTING YOUR PROGRAM

To list the entire program, type the word LIST and press the RETURN key.

```
LIST )
```

RUNNING AN EXISTING PROGRAM

If you want to run an existing program, do the following:

1. After you see the word READY, type the word OLD, a space, and the name of the existing BASIC program. Press the RETURN key.
2. After you see the word READY, type the word RUN and press the RETURN key.

```
OLD NAME.BAS )  
READY  
RUN )
```

LEAVING BASIC AND LOGGING OFF

To leave BASIC and log off the system, type the word BYE and press the RETURN key.

```
BYE )  
JOB 4, USER [27,4072.] LOGGED OFF TTY73 1318 21-JAN-75  
SAVED ALL FILES (80 BLOCKS)  
RUNTIME 51.87 SEC
```

TO RETURN TO MONITOR MODE

To return control to the Monitor, type the command MON and press the RETURN key:

```
MON )
```

USING BASIC

If you want to enter and execute a BASIC program, you can type in your program directly to BASIC; you do not have to use any special editor as BASIC has a built-in editor. The following sections describe how to enter a BASIC program, execute it, and edit it.

STARTING BASIC

After you see the period prompt character, type the letter R followed by a space. Type the word BASIC and press the key labeled RETURN.

```
.R BASIC )
```

```
READY, FOR HELP TYPE HELP.
```

Once BASIC types the message, READY, FOR HELP TYPE HELP, you may use any of the BASIC commands discussed in this section. You may not, however, use any of the TOPS-10 commands. When you give the R BASIC command, TOPS-10 turns you over completely to BASIC.

You can interrupt BASIC in one of two ways. You can type the word MON which is a BASIC command that will return you to TOPS-10 or you can type the BASIC command BYE which logs you off the system, so that you have to log in all over again.

ENTERING YOUR PROGRAM

After BASIC types the words READY, FOR HELP TYPE HELP, you are ready to enter your file. type the word NEW which will indicate to BASIC that you are going to enter a new file in the computer. Press the RETURN key.

```
.R BASIC )
```

```
READY, FOR HELP TYPE HELP.  
NEW )
```

BASIC will type on your terminal NEW FILE NAME--, after which you should type the name of your new file. Press the RETURN key.

```
NEW FILE NAME--ADD1.BAS )
```

BASIC will type READY signifying to you that it is ready for you to start entering your file into the computer.

```
READY
```

When you are entering your program into a file, precede every line of your program with a line number. Press the RETURN key at the end of every line.

Below is a BASIC program that adds two numbers. Type the program exactly as we have typed it here. If you make a mistake after you hit the RETURN key, just keep right on typing; we will

show you how to correct those mistakes a little later. If you make a mistake in a line but you notice it before you press the RETURN key, you can use the RUBOUT key.

```
.R BASIC )
READY, FOR HELP TYPE HELP.
NEW )
NEW FILE NAME--ADD1.BAS )
READY
100 (TAB) PRINT "WHAT ARE THE TWO NUMBERS" )
110 (TAB) PRINT "DO YOU WANT TO ADD?" )
120 (TAB) INPUT A )
130 (TAB) INPUT B )
140 (TAB) LET C = A+B )
150 (TAB) PRINT "THE ANSWER IS: " )
160 (TAB) PRINT C )
170 (TAB) PRINT "DO YOU WANT TO ADD ANOTHER?" )
180 (TAB) PRINT "SET OF NUMBERS?" )
190 (TAB) PRINT "TYPE 1 FOR YES; 0 FOR NO." )
200 (TAB) INPUT D )
210 (TAB) IF D = 1 GOTO 100 )
220 (TAB) PRINT "SEE YOU LATER." )
225 (TAB) STOP )
230 (TAB) END )
```

Check your program over at this time: have you made any typing mistakes within your program? If you have, turn to the section EDITING YOUR PROGRAM before you do anything else. If you have not made any typing errors, you can execute your program. Before you can execute a BASIC program you should SAVE that program.

SAVING YOUR PROGRAM

Once you have finished entering your program into the computer, type the word **SAVE** and press the **RETURN** key. BASIC will print the word **READY** as soon as it has finished **SAVing** your program. If you do not save your BASIC program BASIC will not save it in your disk area, and it cannot be executed. So always remember to save your programs immediately after you type them into a file. The visible results of the **SAVing** procedure are as follows (what you type is underlined):

```
SAVE )  
READY
```

RUNNING YOUR PROGRAM

To run (execute) the program that you entered into the computer, type the word **RUN** and press the **RETURN** key. BASIC will then type the name of your program, along with the current time and date. As soon as you see the word **READY** typed on your terminal, BASIC has finished executing your program.

```
RUN )  
ADD1          13:00          21-JAN-75  
  
WHAT ARE THE TWO NUMBERS  
THAT YOU WANT TO ADD?  
 23 )  
 456 )  
THE ANSWER IS:  
 479  
DO YOU WANT TO ADD ANOTHER  
SET OF NUMBERS?  
TYPE 1 FOR YES; TYPE 0 FOR NO.  
 1 )  
WHAT ARE THE TWO NUMBERS  
THAT YOU WANT TO ADD?  
 77654 )  
 987 )  
THE ANSWER IS:  
 8641  
DO YOU WANT TO ADD ANOTHER  
SET OF NUMBERS?  
TYPE 1 FOR YES; TYPE 0 FOR NO.  
 0 )  
SEE YOU LATER.  
  
TIME:  0.41 SECS.  
READY
```

BASIC executed the program named ADD1.BAS. Everything that appeared on your terminal between the current date that BASIC typed and the word TIME: was a part of your program. Everything that BASIC typed between those two lines were lines that you told it to type within the program.

EDITING YOUR PROGRAM

Check your program over. If you have any typing mistakes, retype the erroneous lines correctly, repeating their numbers also. When you finish correcting these lines, replace the unedited program with the one you have just edited by typing the BASIC command word REPLACE, then a carriage return (which is performed by pressing the RETURN key.)

There are other reasons for editing a program; you may, for example, wish to make your program add three instead of only two numbers. The following procedure will make this change:

```
100 (TAB) PRINT "WHAT ARE THE THREE NUMBERS" )  
135 (TAB) INPUT E )  
140 (TAB) LET C = A + B + E )
```

REPLACING YOUR PROGRAM

After you have finished editing your program, type the word REPLACE and PRESS THE RETURN key. BASIC will print the word READY when it has replaced the unedited program with the edited program.

```
REPLACE )  
READY
```

RE-RUNNING YOUR PROGRAM

Whether you have fixed your program or changed it, run it to find out if it works.

```
RUN )
ADD1          13:03          21-JAN-75

WHAT ARE THE THREE NUMBERS
THAT YOU WANT TO ADD?
  2345 )
  123 )
  456 )
THE ANSWER IS:
  2924
DO YOU WANT TO ADD ANOTHER
SET OF NUMBERS?
TYPE 1 FOR YES; TYPE 0 FOR NO.
  0 )
SEE YOU LATER.

TIME:  0.28 SECS.
READY
```

LISTING YOUR PROGRAM

Get a final copy of your corrected program. To list the entire program in your working area, type the word LIST and press the key on the terminal labeled RETURN.

```
LIST )  
  
ADDI          13:05          21-JAN-75  
  
100          PRINT "WHAT ARE THE THREE NUMBERS"  
110          PRINT "THAT YOU WANT TO ADD?"  
120          INPUT A  
130          INPUT B  
135          INPUT E  
140          LET C = A + B + E  
150          PRINT "THE ANSWER IS: "  
160          PRINT C  
170          PRINT "DO YOU WANT TO ADD ANOTHER "  
180          PRINT "SET OF NUMBERS?"  
190          PRINT "TYPE 1 FOR YES; TYPE 0 FOR NO."  
200          INPUT D  
210          IF D = 1 GOTO 100  
220          PRINT "SEE YOU LATER."  
225          STOP  
230          END  
  
READY
```

To list a single line of your program, type the word LIST, followed by a space, followed by the number of the line that you wish to be printed.

LIST line-no.)

Press the key on the terminal labeled RETURN.

```
LIST 130 )
```

```
ADDI          13:06          21-JAN-75  
  
130          INPUT B  
  
READY
```

RUNNING AN EXISTING PROGRAM

To run an existing program you must do two things. First, you must move a copy of the program you want from your storage area into your working area. Then, you must type the word **RUN** and press the key labeled **RETURN**.

To move a copy of an existing program from your storage area to your working area, type the word **OLD**, a space, and the name of the program that you want to execute. Press the key on the terminal labeled **RETURN**.

OLD filename ↵

Here is an example of the execution of an existing program:

READY
RUN ↵

NAME 13:16 21-JAN-75

WHAT IS YOUR NAME?
?SCOTT ↵
SCOTT

TIME: 0.09 SECS.

READY

LEAVING BASIC AND LOGGING OUT

To leave **BASIC** and log out, type the word **BYE** and press the key on the terminal labeled **RETURN**.

Here is an example of logging out in **BASIC**:

BYE ↵
JOB 17, USER [27,4072] LOGGED OFF TTY73 1152 24-JAN-75
SAVED ALL FILES (90 BLOCKS)
RUNTIME 2.37 SEC

**APPENDIX
TOPS10 COMMANDS**

This appendix lists and briefly describes all commands in the TOPS-10 Command Language. Although most of these commands have not been described within this manual, the purpose of their inclusion in this appendix is to make you aware of the full extent and capability of the TOPS-10 Command Language. The DECsystem-10 Operating System Commands Manual describes all of these commands in detail.

COMMAND	MEANING
ASSIGN	Allocate an I/O device to a specified job.
ATTACH	Detach the current job and connect the terminal to a detached job.
BACKSPACE	Space a magnetic tape backwards a specified number of files or physical records.
CCONTINUE	Continue a program from the point at which it was interrupted; leave the terminal in monitor mode.
CLOSE	Close all files, preserve logical names and device assignments, and terminate any current I/O.
COMPILE	Produce a relocatable binary (.REL) file for a specified source file.
CONTINUE	Continue a program from the point at which it was interrupted.
COPY	Transfer a file from one I/O device to another.
CPUNCH	Place entries into the card punch output device.
CREATE	Open a file for creation and invoke the LINE EDITOR (LINED).
CREF	Invoke CREF to list on the line printer any cross-referenced files generated by a previous COMPILE, LOAD, EXECUTE or DEBUG command.
CSTART	Begin execution of a program which was either previously loaded with the GET command or interrupted while running; leave the terminal in monitor mode.
D(deposit)	Deposit information in the user's core area.
DAYTIME	Type the current date followed by the current time of day.
DCORE	Invoke DAEMON to write a core-image of the user's area; this file includes all accumulators and relevant job tables.
DDT	Load the user program; start the program using DDT.

COMMAND	MEANING
DEASSIGN	Deassign devices currently ASSIGNED, return these devices to the monitor's pool of available devices and clear any logical names associated with these devices.
DEBUG	Translate a source program; load DDT followed by the user's program with local symbols. Transfer of control is given to DDT.
DELETE	Delete one or more files from disk or DECtape.
DETACH	Disconnect the terminal from the user's job without affecting the status of the job.
DIRECT	List all directory entries.
DISMOUNT	Return a device to the monitor's pool of available devices and remove a file structure from the job search list.
DSK	Type the disk usage for the combined structures of the job both since the last DSK command was issued and since the job was initiated.
DUMP	Invoke DAEMON to write a core image file; invoke DUMP to analyze the file written and to provide printable output.
E(examine)	Examine a core location in the user's core area (high or low segment).
EDIT	Open an already existing line-numbered file on disk for editing with the Line Editor (LINED).
EOF	Write an end-of-file mark on the specified mag tape.
EXECUTE	Translate a source file, load the generated .REL file, and begin execution.
FILE	Provide remote control of DECtape-to-disk and disk-to-DECtape transfers on operator-handled DECtapes.
FINISH	Terminate any I/O currently in progress on a specified device; close the file; release the device; DEASSIGN the device.
FUDGE	Create a library .REL file from a temporary file generated by a previous COMPILE, EXECUTE, or DEBUG command string containing the /FUDGE switch.
GET	Load a core image file from a retrievable storage device; execution will not begin automatically.
HALT	Stop the current job and store the place where interruption occurred in the job data area.
HELP	Type useful information about various system features.
INITIA	Perform standard system initialization for a terminal.

COMMAND	MEANING
JCONTINUE	Continue a job that was stopped because of a CTRL/C.
KJOB	Kill the current job.
LABEL	Write an identifier onto a DECtape.
LIST	Invoke PIP to list the contents of a named source file on the line printer.
LOAD	Translate the specified source file; invoke the linking loader and load the .REL file generated.
LOCATE	Establish the user's job at a specified remote station.
LOGIN	Establish the user's authority to gain access to the system.
MAKE	Create a new file on disk with TECO (text Editor and Corrector.)
MOUNT	Request assignment of a device via the operator.
PJOB	Type the job number to which the user's terminal is attached.
PLEASE	Allow two-way communication between a user and the designated station operator.
PLOT	Place entries in the plotter queue.
PRESERVE	Preserve the specified files with the standard protection code (either 055 or 057) inclusively ORed with 100.
PRINT	Place entries into the line printer queue.
PROTECT	Alter the access protection codes associated with the specified file.
QUEUE	Make an entry in a system queue.
R	Load a core image file from a system device and start the file.
R ALCFIL	Allocate space for a new file or reallocate space for an existing file in one contiguous region on the disk.
R DTCOPY	Invoke the DTCOPY routine which will: <ul style="list-style-type: none"> 1. Copy the entire contents of an input DECtape to an output DECtape. 2. Zero all blocks on an output DECtape and clear the directory. 3. Perform a word-by-word comparison of two DECtapes. 4. Load a bootstrap loader and write it in blocks 0, 1, and 2 of the output DECtape.

COMMAND	MEANING
R DUMP	Provide printable dumps of arbitrary data files in a mode form specified by the user.
R FAILSAFE	Save disk files on magnetic tape and place all or some of these saved files back onto the disk.
R FILCOM	Compare two versions of a file and output any differences.
R FILEX	Transfer files and convert between one core image format to another core image format. Read and write various DECTape formats.
R FUDGE2	Update files containing one or more relocatable binary modules and manipulate the individual modules within these files.
R GLOB	Read multiple binary files and produce an alphabetical cross-referenced list of all global symbols (symbols accessible to other programs) encountered. The GLOB program also searches files in library search mode, checking for global symbols, if the program file was loaded by LINK-10 in library search mode.
R GRIPE	Invoke the GRIPE program which accepts text from a user and records it into a disk file, thereby enabling users to record comments and complaints to be read at a later time by the operating staff.
R OPSER	Invoke the OPSER program which is used to facilitate multiple job control from a single operator terminal by allowing the operator to run several jobs called subjobs from his terminal. The OPSER program acts as the supervisor of the various subjobs by allowing monitor level or user level commands to be passed to all of the subjobs or to selected subjobs.
R QUOLST	Invoke the QUOLST program which informs the user of both the amount of disk space he has used and the amount he has left on each file structure in his search list.
R REATTA	Invoke the REATTA program which allows a user to transfer his job from one terminal to another.
R SETSRC	Invoke the SETSRC program which manipulates the job's search list of the system's search list. A search list is defined to be the order of file structures that are to be searched whenever generic device DSK: is explicitly or implicitly specified by the user. The user can alter the search list defined for him by adding or deleting structures from it by using the SETSRC program.
REASSIGN	Allow one job to pass a device to a second job without having the device go through the monitor device pool.

COMMAND	MEANING
REENTER	Copy the saved program counter value from .JBPC into .JBOPC and start the program at an alternate entry point specified in .JBREN, .JBREN must be set by the user or by the user's program.
RENAME	Change the name of one or more files on disk or DEC-tape.
RESOURCES	Print the names of all available devices (except TTY's and PTY's), all file structures, and all physical units not in file structures (unless they are down or non-existent.)
REWIND	Rewind a magnetic tape or a DECtape.
RUN	Load a core image from a retrievable storage device and start at the location specified within the file (location .JBSA in the job data area location.)
SAVE	Write out a core image of the user's core area on the specified device.
SCHED	Type out the schedule bits as set by the last privileged SET SCHED command.
SEND	Provide a mechanism for one-way interconsole communication. A line of information is transmitted from one terminal to another, with the identification of the terminal sending the information. With remote communications capabilities, SEND is able to differentiate stations.
SET BLOCKSIZE	Set a default blocksize for the specified magnetic tape.
SET BREAK	Set a break address which will cause a break condition when the specified address is read from, written into, and/or fetched. (KI-10 and KL-10 processors only.)
SET CDR	Set the filename for the next card-reader spooling intercept.
SET CPU	Allow a privileged user to change the CPUs on which his job can run.
SET DENSITY	Set a default density for the specified magnetic tape.
SET DSKFUL	Control the treatment of a user's job when it is attempting output when one of two conditions occur. One condition is when there is no available space on the file structure being referenced; the other condition is when the user's quota for that file structure has been exceeded.
SET DSKPRI	Allow a privileged user to set the priority for his job's disk operations.
SET HPQ	Allow a privileged user to place his job in a high-priority run queue.

COMMAND**MEANING**

SET PHYSICAL LIMIT	Set a physical user maximum physical page limit.
SET PHYSICAL GUIDELINE	Set a maximum physical page guideline.
SET SPOOL	Add devices or delete from the current list of devices being spooled for this job.
SET TIME	Set a central processor time limit for a job.
SET TTY	Declare properties of the terminal line on which the command is typed to the scanner service. With hardwired TTY's the system manager can set the default conditions, so that this command is usually not needed. However, the user is likely to use this command on data sets, where the terminal properties cannot be predicted.
SET VIRTUAL LIMIT	Set a user virtual page limit.
SET WATCH	Set the system to print incremental job statistics automatically. This command provides the user with a tool for measuring the performance of a specified program.
SKIP	Space a magnetic tape forward a specified number of files or records or to the logical end of the tape.
SSAVE	This command is the same as the SAVE command except that the high segment if present, will be sharable when it is loaded with the GET command.
START	Begin execution of a program either previously loaded with the GET command or interrupted (i.e., with a CTRL/C).
SUBMIT	Place entries into the input queue for the Batch system.
SYSTAT	Run a system program which prints status information about the system. This information allows a user to determine the load on the system before logging-in.
TECO	Invoke TECO (Text Editor) and open an already existent file on disk for editing.
TIME	Type the total running time since the last TIME command, followed by the total running time used by the job since it was initialized (logged-in), followed by the integrated product of running time and core size (KILO-CORE-SEC=).
TPUNCH	Place entries into the paper-tape punch output queue.
TTY	Declare properties of a terminal (see SET TTY command.)
TYPE	Direct PIP to type the contents of the named source file(s) on the user's terminal.

COMMAND	MEANING
UNLOAD	Rewind and unload a magnetic tape or a DECtape.
USESTAT	Return status information pertinent to the current user. (Identical to a CTRL/T).
VERSION	Print the version number of the program in the user's core area (i.e., the last program run implicitly or explicitly.)
WHERE	Enable the user to determine the station at which a specific peripheral device is located. If the station of a particular terminal is requested, the number returned is the physical location of the terminal, which may or may not be the location of the controlling job. This depends on whether the user changed his job's logical location with the LOCATE command.
ZERO	Clear the directory of the output device.

BASIC STATEMENTS

The BASIC statements used in this manual are briefly described below. More detailed information can be obtained from the BASIC CONVERSATIONAL LANGUAGE MANUAL (DEC-10-LBLMA-A-D).

END

Indicates the end of the program and must be the statement with the highest line number. This statement is always required.

IF *expression*; **GOTO** *line-number*

Transfers control to the statement at the line number specified if the given expression is true.

INPUT *variable, variable, ..., variable*

Allows data to be entered from the user's terminal during the execution of a program.

LET *variable = expression*

Assigns the value of the expression to the right of the equal sign to the specified variable(s).

PRINT { *string* }
{ *variable* }

Types out either text strings (which are enclosed within quotation marks) or the value of an expression.

STOP

Stops the program and its equivalent to GOTO n, where n is the line number of the END statement.

SAMPLE BASIC PROGRAM

```

100 PRINT "WHAT ARE THE TWO NUMBERS THAT YOU"
110 PRINT "WANT TO ADD?"
120 INPUT A
130 INPUT B
140 LET C = A + B
150 PRINT "THE ANSWER IS:"
160 PRINT C
170 PRINT "DO YOU WANT TO ADD ANOTHER SET"
180 PRINT "OF NUMBERS?"
190 INPUT D
200 IF D = 1 GOTO 100
210 PRINT "SEE YOU LATER"
220 STOP
230 END

```

ALGOL STATEMENTS

The ALGOL statements used in this manual are briefly described below. More detailed information can be obtained from the ALGOL PROGRAMMER'S REFERENCE MANUAL (DEC-10-KAZB-D).

BEGIN

Indicates the beginning of a block of compound statements. the block is terminated by an END statement.

COMMENT *text*;

Indicates that the text which follows is to be treated as a comment (the ! symbol may be substituted for the word COMMENT)

END

Indicates the end of a block of compound statements; the block begins with a BEGIN statement.

INTEGER *variable, variable, ..., variable*;

Specifies a list of variable names which are to have integer values.

PRINT (*variable, format-expression, format-expression*);

Prints the value of the variable specified in the format specified.

READ (*variable*);

Associates a value with the specified variable.

WRITE (*format-specification*);

Outputs data in the format specified.

SAMPLE ALGOL PROGRAM

```

BEGIN
INTEGER S, C, A;
WRITE ("[2C] WHAT ARE THE NUMBERS ?(IC)");
READ (S);
READ (C);
A := C+S;
WRITE ("[2C] THE SUM OF ");
PRINT (S,3,3);
WRITE ("AND");
PRINT (C,3,3);
WRITE ("IS:");
PRINT (A,3,3);
WRITE ("[2C]");
END

```

FORTRAN-10 STATEMENTS

The FORTRAN-10 statements used in this manual are briefly described below. More detailed information may be obtained from the FORTRAN-10 Language Manual (DEC-10-LFORA-B-D).

ACCEPT *statement-label, variable, variable, ..., variable*

Allows for the input of data via the user's terminal during the execution of a program.

END

Indicates the physical end of a source program.

statement-label **FORMAT** (*'text' format-specification*)

Allows for the formatting of input and output data.

TYPE *statement-label, variable, ..., variable*

Outputs the values of the specified variable(s) in the format specified by the indicated FORMAT statement.

SAMPLE FORTRAN PROGRAM

```

TYPE 10
10 FORMAT ('WHAT ARE THE TWO NUMBERS?')
ACCEPT 20, X, Y
20 FORMAT (2F)
ANSWER = X + Y
TYPE 30, ANSWER
30 FORMAT ('THE SUM OF THE NUMBERS IS',F7.2)
END

```

NOTES

REMOTE TERMINAL COMMUNICATIONS

1. Turn on the terminal
2. Check the speed setting of the terminal (most terminals are labeled either 10 CPS, 15 CPS, and 30 CPS or 110 baud, 150 baud, and 300 baud) It should initially be set to either 10 CPS or 110 baud
3. Dial the phone number to the computer.
4. Wait for a steady tone.
5. Lay the phone receiver in the slots provided for it either on the terminal or on the acoustic coupler
6. Wait for the carrier detect light to come on
7. If you see nothing printed on the terminal, type two CTRL/Cs until you see either the prompt character or the words PLEASE LOGIN OR ATTACH
8. Follow Steps 1 through 7 of the LOGGING-IN PROCEDURE.

LOGGING-IN PROCEDURE

1. Type a CTRL/C.
2. After the period prompt character, type LOGIN
3. Press the RETURN key.
4. After the system prints #, type your project-programmer number.
5. Press the RETURN key.
6. After the system types PASSWORD, type your password.
7. Press the RETURN key.

LOGGING-OFF PROCEDURE

1. After the prompt character, type K/F.
2. Press the RETURN key.

-or-

1. When using BASIC, type the command BYE
2. Press the RETURN key.

CREATING A FILE

1. After the prompt character, type R SOS
2. After FILE is printed, type the name of your file
3. The system will print INPUT followed by your file name
4. After the system prints a line number, type a line of your file
5. Press the RETURN key
6. Repeat steps 4 and 5 until you have entered the entire file
7. Press the ESC key.
8. After the system prints a *, type F
9. Press the RETURN key.

DELETE A LINE

1. Type a CTRL/U.
2. The system will advance the carriage down one line and over to the first character position.
3. Type your new line. (HINT: If you are entering a file and the system is printing line numbers for you, do not type the line number over again.)
4. Press the RETURN key.

VIEWING A CORRECTED LINE

1. Correct the line using the RUBOUT key.
2. Before you press the RETURN key, type CTRL/R.
3. View the corrected line. If it is acceptable, press the RETURN key. If it is not acceptable, repeat steps 1 through 3 until the line is acceptable or use the CTRL/U feature

DELETE ONE OR MORE CHARACTERS

1. Press the RUBOUT key once for each character you want to delete.
2. Type the corrected line from the point of deletion
3. Press the RETURN key.

EDITING A PROGRAM

1. After the prompt character, type R SOS
2. After FILE is printed, type the name of the file you want to edit
3. The system will type EDIT followed by your file name
4. After the *, type one of the SOS commands.

SOS COMMANDS

	format	example
Print a Line	*Pline#)	*P1300
Print a Series of Lines	*Pline# - line #)	*P500-700
Insert a Line	*Iline#)	*I750
Delete a Line	*Dline#)	*D800
Replace a Line	*Rline#)	*R900
Changing a Line	*Stex\$tex\$line#)	*Sabc\$adc\$1300
Exit from SOS	*E)	*F

EXIT FROM THE SOS EDITOR

1. If you are in input mode (SOS types line numbers for you), press the ESC key.
2. After the *, type E
3. Press the RETURN key.

-or-

1. If you are in command mode, type E after the *.
2. Press the RETURN key.

EXECUTING A USER PROGRAM

1. After the prompt character, type EXECUTE
2. Type a space followed by the program name.
3. Press the RETURN key.

STOP A USER PROGRAM

1. Type two CTRL/Cs

LISTING A PROGRAM

On a terminal

1. Type the word TYPE, after the prompt character
2. Type the name of the file you want to be listed
3. Press the RETURN key

On the line printer

1. After the prompt character, type Q L P T =
2. Type the name of the file you want to be listed
3. Press the RETURN key

Using BASIC

1. Type the word LIST
2. Press the RETURN key

LISTING NAMES OF FILES IN YOUR DISK AREA

1. After the prompt character, type DIRECT
2. Press the RETURN key

RENAMING A FILE

1. After the prompt character, type the word RENAME.
2. Type a space followed by the new file name
3. Type an equal sign followed by the old file name.
4. Press the RETURN key

INDEX

- .ALG, 12
- ALGOL
 - sample program, 14
 - execute a program in, 20
- ALT MODE key, v
- Asterisk, 23

- .BAS, 12
- BASIC
 - entering a file in, 39, 41
 - listing a program in, 46
 - running a program, 39, 41, 45
 - sample program, 42
 - saving a program, 39, 41
 - using, 39-49
- BASIC commands
 - LIST, 46
 - REPLACE, 44
 - RUN, 39, 41, 45
 - SAVE, 39, 41

- Carriage return, v, 4
- .CBL, 12
- Change
 - a line (in SOS), 24, 29
 - a FORTRAN program, 25
 - an ALGOL program, 26
- Checking a program, 20
- Command argument, 5
- Commands
 - COPY, 34, 37
 - DAYTIME, 6
 - DELETE, 33
 - DIRECT, 33, 35
 - EXECUTE, 17, 19
 - HELP, 34, 36
 - K/F, 6
 - LOGIN, 5
 - RENAME, 37
 - RUN, 34
 - SYSTAT, 7
 - TYPE, 25, 26, 30
- Compiler, 19
- Control characters
 - CTRL/C, 4, 18, 21
 - CTRL/I, 13
 - CTRL/O, 7
 - CTRL/T, 20
- Conventions, v
- COPY command, 34, 37
- Copying a file, 34, 37
- Correcting a program, 26
- CTRL key, v, 3
- CTRL/C, 4, 18, 21
- CTRL/I, 13
- CTRL/O, 7
- CTRL/T, 20

- DELETE command, 33, 35
- Delete
 - a file, 33, 35
 - a line, 24, 28
- DAYTIME command, 6
- DIRECT command, 33, 35

- Entering
 - a file (SOS), 12, 23
 - a BASIC program, 41
- Editing
 - a BASIC program, 44
 - a program, 12, 23, 25, 27
 - a file, 12, 23, 25, 27
- Erase
 - a character, 15
 - a line, 24, 28
- ESCAPE key, v, 4, 13
- EXECUTE command, 17, 19, 20, 34, 38
- Execution
 - of an ALGOL program, 20
 - of another user's program, 34, 38
 - of a BASIC program, 45
 - of a FORTRAN program, 20
 - of a program, 17, 19
 - stopping, 21
- Extensions, file name, 12

- Files
 - definition of, ii
 - delete a, 33, 35
 - editing, 23, 25
 - entering a, 12
 - extensions to file names, 12

list names of, 33, 35
names of, ii
renaming, 37
FORTRAN program, 14, 19
.FOR, 12

HELP command, 34, 36

Insert a line, 23, 28
Interrupting
the system, 4
a program, 21, 28

K/F command, 6

Line numbers
in BASIC, 41
in SOS, 13
List
names of files, 33, 35
of TOPS-10 commands App.
LIST command, 46
LOGIN command, 5
Logging-in, 5
Logging-off, 6

.MAC, 12

Passwords, 3
Period, 4
PPN, 3
PREFIX key, v
Print a line, 23, 27
Program
checking a, 20
definition of, ii
editing a, 23, 25
editing a BASIC, 44
execute a, 17, 19
execute a BASIC, 43, 45
saving a BASIC, 43
stop a, 21, 28
typing a, 30

Project-programmer numbers, 3
Prompt character
system (.), 4
SOS (*), 23

.REL, 19
Rename a file, 37, 38
RENAME command, 37
Replace a line, 24, 28
REPLACE command (BASIC), 42
RUBOUT key, v, 4, 15
RUN command (BASIC), 39, 45
RUN command (TOPS-10), 34

SAVE command (BASIC), 43
SAVE command (TOPS-10), 39
Saving
a file, 24, 29
a BASIC program, 43
SOS commands
change a line (S), 24, 29
delete a line (D), 24, 28
insert a line (I), 23, 28
print a line (P), 23, 27
replace a line (R), 24, 28
save a file (E), 24, 29
SOS editor
start the, 23, 27
line numbers, 13
using, 12, 23, 13, 27
Source language, 19
Stopping a program, 18, 21
SYSTAT command, 7

TAB key, vi, 13
Terminals, 3
TOPS-10 commands App.
Translation, 19
TYPE command, 25, 26, 30
Typing a program, 30
Typing mistakes, 13, 15

READER'S COMMENTS

NOTE: This form is for document comments only. Problems with software should be reported on a Software Problem Report (SPR) form (see the HOW TO OBTAIN SOFTWARE INFORMATION page).

Did you find errors in this manual? If so, specify by page.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

Please indicate the type of user/reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Non-programmer interested in computer concepts and capabilities

Name _____ Date _____

Organization _____

Street _____

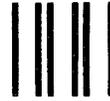
City _____ State _____ Zip Code _____
or
Country

If you do not require a written reply, please check here.

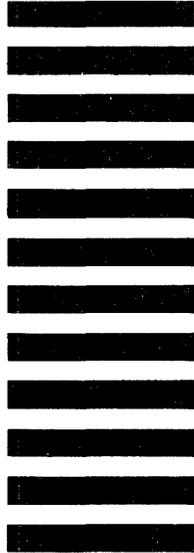
Please cut along this line.

-----Do Not Tear - Fold Here and Tape-----

digital



No Postage
Necessary
if Mailed in the
United States



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE PUBLICATIONS
200 FOREST STREET MR1-2/E37
MARLBOROUGH, MASSACHUSETTS 01752

-----Do Not Tear - Fold Here and Tape-----

Cut Along Dotted Line