



**AgEcon** SEARCH  
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

*The World's Largest Open Access Agricultural & Applied Economics Digital Library*

**This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.**

**Help ensure our sustainability.**

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

[aesearch@umn.edu](mailto:aesearch@umn.edu)

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

## **Historic, Archive Document**

Do not assume content reflects current scientific knowledge, policies, or practices.



aQA76  
.9  
.V5S44  
1984

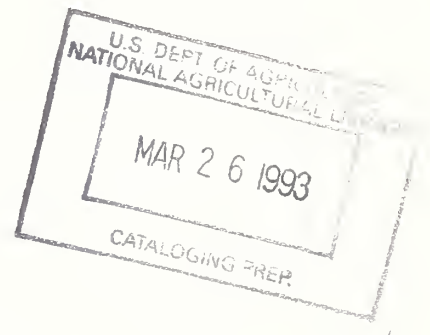
tates  
ent of  
re

Economic  
Research  
Service

International  
Economics  
Division

# Search Command for Character Strings and Words in Computer Files on CMS (Conversational Monitor System)

Ralph M. Seeley



**United States  
Department of  
Agriculture**



**National Agricultural Library**

SEARCH COMMAND FOR CHARACTER STRINGS AND WORDS IN COMPUTER FILES ON CMS (CONVERSATIONAL MONITOR SYSTEM). By Ralph M. Seeley. Trade Policy Branch, International Economics Division, Economic Research Service, U.S. Department of Agriculture. May 1984. ERS Staff Report No. AGES840416.

ABSTRACT

The SEARCH command locates a string of characters in a specified set of computer files. The user may restrict the search to find a freestanding word or variable. The command records each line containing the character string or variable, along with the line number and file name. The record of lines found may be further searched for additional strings or variables. Lines containing or beginning with each string or variable may be kept or deleted. The menu-driven command records user input. SEARCH is comparable to the UNIX operating system's GREP, and runs on the Conversational Monitor System (CMS).

Keywords: character string search, word and variable search, locate, find, CMS, GREP, UNIX operating system, computer program debugging, computer file management, data base maintenance.

ACKNOWLEDGMENTS

The author thanks Charles Hallahan, Kenneth Koester, William Kost, and Robert Rovinsky for their editorial comments. An introduction to the VM/370 Control Program by Peter Kostik is appreciated. Thanks are owed to the Data Services Center for granting permission to use its minicomputer, which was a prerequisite for this and related work.

Use of brand names (or company names) in this publication is for identification only and does not imply endorsement by the U.S. Department of Agriculture.

\* \* \* \* \*  
\* This paper was reproduced for limited distribution to the \*  
\* research community outside the U.S. Department of \*  
\* Agriculture \*  
\* \* \* \* \*

CONTENTS

SUMMARY.....	iii
INTRODUCTION.....	1
USERS' GUIDE TO THE SEARCH COMMAND.....	1
Initial Search String.....	2
Identification of Files to Search.....	2
Initial Search Technique.....	4
Editor.....	5
Further Processing Menu.....	6
Sample Session.....	7
Sample Search Results.....	9
PROGRAMMERS' GUIDE TO THE SEARCH COMMAND.....	9
Search Command Limitations.....	9
Program Use.....	10
Program Modification.....	12
Narrative Program Explanation.....	12
Program Flowcharts.....	13
Program Listings.....	27
REFERENCES.....	47
GLOSSARY.....	48
INDEX.....	49

## SUMMARY

The search command locates a consecutive group of characters (a character string) in a specified set of computer files. The search may be restricted to locating freestanding words or variables. The menu-driven command records lines in which the strings or variables are found, their line numbers and file names, and the user input. The output may be viewed, printed, and further searched for additional strings or variables. Lines containing or beginning with each string or variable may be kept or deleted.

Use of the command proceeds in six steps. Steps five and six may be repeated for additional searches through the same output.

1. Enter the command name, S.
2. Enter the initial search string or variable. All characters entered, including blanks, are part of the string or variable. If nothing is typed before the enter key is pressed, a help document will be displayed at the terminal.
3. Enter the identifier of the set of files to be searched. The identifier consists of filename, filetype, and filemode. Example: MAXIM FORTRAN A. The identifier may include the "wild card" symbols \* (representing zero or more of any characters) and % (representing one of any character). Example: \* DAT% A. The default identifier is \* FORTRAN A. The identifier SEARCH RESULTS allows resumed work with the last search output. An identifier with filetype SFILES may be used, referring to a special file containing the identifiers of other files to be searched.
4. Enter the initial search technique. S simply searches for the string. V searches for the string as a freestanding variable or word. O also simply searches for the string. It can be faster than S. O should only be used for filetypes ASM370S, ASSEMBLE, AMSERV, COBOL, COPY, DIRECT, FORTRAN, MACLIB, MACRO, PLI, PLIOPT, UPDATE, and UPDTxxxx. X stops the program.
5. View or modify the output file while it is in the editor. The editor is automatically entered after every search. Leave the editor by pressing program function key 3.
6. Enter a further processing command. X stops the program. P prints the output, with a filetype reflecting the initial search string, if possible. L returns the display to the editor (step five). F creates an SFILES list from the output. The commands D, C, S, and V search through the output file, while ignoring the search documentation, line numbers, and file identifiers. These commands require entering of a new search string. D deletes any lines which contain the new search string. C deletes any lines beginning with the new search string. S keeps only lines containing the new search string. V keeps only lines containing the new search string where it appears as a freestanding variable or word.





## INTRODUCTION

This paper describes a computer command which allows searching for a variable, word, number, or other character string in a specified group of files. It is available to users of computers which run the CMS (Conversational Monitor System) operating system (3), and who wish to search for character strings through complicated files, large files, or large numbers of files. 1

The search routine can cut the time required to find where a variable is assigned a value, for example, from several hours to several minutes. Moreover, during those several minutes, the researcher can be occupied with other activities. A copy of the material found may easily be printed and saved for reference. The search command is far less prone to error than the researcher. A subset of the lines found can be generated by directing the routine to search for other character strings in the output and keep or delete lines in which the strings are found. A record of such processing is automatically kept in the output file and displayed at the terminal. When the output is printed, its name reflects the search string.

The search command described in this article was originally devised to aid debugging of the IIASA (International Institute for Applied Statistical Analysis) world agricultural model. There are several reasons for which such a search capability is essential for the IIASA project. The first is the sheer size of the model. It consists in all of roughly 260 subroutines, or 46,000 lines of FORTRAN computer code, specially written for the model. There is little documentation available on the model at the program level. There are few explanatory comments within the model. Many variables are passed to as many as 50 or more subroutines.

This writer has felt the need for such a search program for several years. However, doing such searching would be expensive on a mainframe computer, and slow on a microcomputer or word processor. Moreover, the latter two devices provide inadequate storage space. Access to the Data Service Center's minicomputer has made use of the search command practical for work on the IIASA model. The utility of a procedure allowing searching for a string across a group of files was verified during the use of the GREP command from the Berkeley version of the UNIX operating system (1). The GREP command, although not designed to search for a word or variable as such, is a valuable debugging tool.

## USERS' GUIDE TO THE SEARCH COMMAND

The beginning user of the search command may wish to start by reading the summary at the front of this article. The files to be searched must be directly accessible in the Conversational Monitor System (CMS), in which the command operates. After each of the following input lines is entered, the program prompts for the next input using a menu. If there are tens of

---

1/ Underscored numbers in the parentheses refer to items in the references.

thousands of lines through which to search, a few minutes of processing could follow entering of the search technique. The user initiates the search by entering the abbreviated command name, S.

### Initial Search String

The program responds to entering of its abbreviated name by prompting for the initial search string. This search string may include leading, trailing, or embedded blanks. To erase an extra blank, use either the DEL or ERASE EOF key (this paper assumes use of a terminal similar to the International Business Machines Corporation 3270 series; see reference 2). If no search string is typed before the ENTER key is pressed, a help document will be displayed at the terminal. The text of this help document may be seen in the later section of this article listing the program file SRCHHELP EXEC.

The user who has become familiar with the search command may wish to enter the search string on the same line as the command name. The command name is followed by a single blank and then the search string. As described in the preceding paragraph, the search string may contain blanks. After the single blank separating the command name S from the search string, all blanks typed are part of that string.

### Identification of Files to Search

A brief description of the file identification syntax under CMS will be given here (4). Three pieces of information serve to identify a CMS file. The first is the filename. For the purposes of the IIASA model, the filename might be the name of a subroutine or data file. Second is the filetype. The filetype indicates the use for the file and generally the file format. Examples include FORTRAN (for a file containing a FORTRAN program) and DATA (for a file holding data input to or output from a program). The filemode is the third piece of information identifying a file. It is a letter which tells where the file is stored. The filemode is most often A, which points to the user's prime storage area. Users familiar with microcomputers will find that filemodes are comparable to disk drive letters on microcomputers.

### Simple Identifier

After the command name and search string are entered, the program prompts for an indication of the set of files over which to search (referred to here as the fileset identifier). There are five options at this point. The first is to simply enter the identifier (ID) of a particular file; for example, MAXIM FORTRAN A.

### Group Identifier

In the S command, the characters \* and % may be used as part of a filename, filetype, or filemode to specify groups of files. The \* represents any combination of zero or more valid characters. The % represents any one valid character. Valid characters are letters, numbers, and the characters on the next line:

\$ - + : \_

Thus, the fileset identifier \* FORTRAN A represents all FORTRAN files on the A disk. The fileset identifier \*S%100 DATA A\* would encompass, among others, the file IDs SY100 DATA A1,

FAST100 DATA A0, and MS9100 DATA A4. A copy of the list of existing files which match the fileset identifier is alphabetized and used for the search.

- Default Identifier** If an incomplete fileset identifier is entered, the remainder will be supplied from default values. If nothing at all is typed before the ENTER button is pressed (null input), the file set to be searched is composed completely of the defaults. The default filename is \*, while the default filetype is FORTRAN, and the default filemode is the mode of the disk on which the S command resides, usually A for device 191.
- Resumed Search** There are two other ways to designate files to process with the S command. The first is to resume working with the most recent search output. This may be done by giving the fileset identifier SEARCH RESULTS.
- List of Identifiers (SFILES List)** The fifth type of fileset identifier is the identifier of an SFILES list. The SFILES list is a special sort of file. It contains a list of file IDs. If the fileset identifier refers to an SFILES list, the files named in the list will be searched. The benefit of applying such a list can be a savings in time and effort for the user. With an SFILES list, the user can specify precisely which files to search, even if the files do not have IDs that fit a pattern.
- Benefit of SFILES List** For example, a researcher has stored on a computer 200 files which contain FORTRAN programming language subroutines. The researcher wants to examine the use of certain variables in the subroutines. The researcher soon discovers that the only subroutines relevant to the examination are the 50 in which the FORTRAN statement, COMMON /WORK/, appears. The researcher creates an SFILES list containing IDs of files in which that statement appears. The researcher may now search for the various variables through only 50 files instead of 200. Searching is quicker and reading the output is easier because there is less likelihood of finding irrelevant lines.
- Format of SFILES List** In the SFILES list, file IDs must be entered, one file ID per line, with no blank lines. The file IDs need not start in any particular column. In general, it is better not to include filemodes in the SFILES list if the user wants the option of changing disk modes. If no mode is entered for a file in the SFILES list, the default is the current mode of the disk on which the S program resides. If the default is not desired, the mode may be specified as part of the fileset identifier. If the filetype is also missing, the default is FORTRAN. The program alphabetizes a copy of the SFILES list to be used for the search.
- Fileset Identifier For SFILES List** The filetype of the list must be SFILES. The fileset identifier may include the mode of the SFILES file. A second filemode may be entered in the fileset identifier to indicate the filemode of the files to be searched. The defaults for both of these filemodes are the filemode of the S program. For example, if the S program resides on disk B, but the WORK SFILES list is

created on disk A and the files to be searched are located on disk J, then the fileset identifier would be WORK SFILES A J.

#### Creation of SFILES List

The SFILES list may be created by typing of file IDs. Alternatively, it may be created through use of the search command. In the example above explaining possible benefits of using a list, the researcher wishes to create a list of files in which the FORTRAN statement, COMMON /WORK/, appears. The researcher can begin to create the list by searching for the statement through all 200 of the files. (Later sections in this article explain more specifically how the search would be carried out.) In this example, the best approach would probably be to search for the word WORK, then further search for the word COMMON, and finally delete all lines beginning with a C (FORTRAN comment lines). Searching in more than one step allows for the fact that a varying number of blanks is permitted between the words and the slashes. In any case, the researcher need only ensure that all of the desired lines have been found, but that no irrelevant lines have been found. The command F in the further processing menu may then be used to copy the file IDs from the output into an SFILES list.

#### Initial Search Technique

After the choice of a search string and a fileset identifier comes the selection of a search technique. There are three relevant goals: speed, opportunity for further searching, and limited output. There are several parameters: the likelihood of encountering more than 20 lines per file which contain the search string; any need for the string strictly as a variable; the filetype; and any further processing desired. Also relevant is the prior use of editors other than XEDIT (7), such as SPF, the System Productivity Facility (8). The time required for searching is in part a function of the number of files to be searched, the total number of lines to be searched, the search string, and the search technique. The S search technique is the default.

#### S Technique

The S search technique searches for a string. It gives line numbers no matter which filetype or which editor was used. The speed of the S technique is largely a function of the factors listed in the preceding paragraph. As each file is searched, its ID is displayed at the terminal. Lines found as a result of the search are not displayed during the search, but are shown when the search is completed and the output file is made visible in the editor.

#### V Technique

The V technique records lines in which the search string appears, neither preceded nor followed by letters or numbers. What are permitted to precede or follow the string are algebraic symbols, blanks, the beginning or end of a line, the dollar sign, punctuation, and so on. This definition is approximately equivalent to that for a variable in FORTRAN and similar programming languages. The programmers' section of this article lists some exceptions. The definition is also roughly the same as the pattern of characters which distinguishes a word in English text. The speed of the V technique is in part a function of the number of times that the

search string is found, but is preceded or followed by a letter or number. If the user's goal is finding a string as a variable, then it may be best to start searching with the V technique because the volume of output generated by the search is less with this approach. Otherwise, searching for a string which appears in many lines can create an output file too large to be edited. As each file is searched, its ID is displayed at the terminal. Lines found as a result of the search are not displayed during the search until the editor is entered.

#### O Technique

The O technique is the fastest when no more than 20 lines containing the search string are likely to be found per file. The output is displayed on the screen as it is found. If more than a screenload is found per file, the screen must be cleared for the program to proceed. The clearing may be accomplished by pressing of the CLEAR button. Otherwise, the screen will be automatically cleared after one minute. Thus, the speed of the technique is highly dependent on the number of times that more than 20 occurrences of the search string will be found per file, given that the researcher is unwilling to divert attention from other work while the search is taking place just to press the CLEAR button. In ideal circumstances, the O technique is almost twice as fast as the S technique. The O technique provides line numbers only for the following filetypes: ASM370S, ASSEMBLE, AMSERV, COBOL, COPY, DIRECT, FORTRAN, MACLIB, MACRO, PLI, PLIOPT, UPDATE, and UPDTxxxx. The O search technique should not be used for other filetypes. Line numbers displayed by the O search technique are preceded by zeros and followed by one zero. For example, line number 137 would appear as 01370.

#### X Technique

The search string and filesset identifier as received by the command are typed below the search technique menu, permitting the user to verify that continuing with the search is appropriate. If not, typing X in response to the request for a choice of search technique stops the program.

#### Breaking Search

To break during the middle of a search, press the PA1 key repeatedly until the message VM READ appears at the lower right corner of the terminal screen. Next, type HX and press the ENTER key. The message CMS will appear, after which any CMS command may be entered.

#### Editor

After the initial search takes place, the editor XEDIT (7) is entered automatically. The user may view or edit the output file at this point.

#### Program Function Keys

Twelve program function (PF) keys are preset for the editor. PF key 1 returns to a single screen after the use of a split screen, initiated with PF 2. PF 2 gives a split screen. A second file may be edited on one of the two split screens if the user types XEDIT and a file ID on the command input line. PF key 3 saves the output file and returns the user to the further processing menu. PF 4 repeats the last XEDIT command. PF 5 makes visible the XEDIT line numbers for the output file. This enables the use of XEDIT line edit commands, mainly for

deleting lines. PF 6 hides the line numbers made visible by PF 5, enabling display of an entire FORTRAN statement. PF key 7 displays the previous screen and PF 8 the next screen. PF 9 makes the display jump to the top of the file. PF key 10 causes display of the left side of the file. PF 11 moves the display toward the right of the file. PF 12 causes the display of the end of the file. An abbreviated list of PF key settings is displayed on the third line of the XEDIT screen when the output file is viewed. When editing ceases after pressing of the PF 3 key, the output is saved and the further processing menu is displayed.

#### Further Processing Menu

The further processing menu is displayed when the editor is left. The several menu choices will now be described.

- Processing Command X The first option is X. If X is entered, the user immediately exits the search program.
- Processing Command P The command P prints the output file in the state in which it was last viewed. The filetype is changed to the first word of the initial search string, if that word contains no invalid characters. This temporary name change allows easier reference to the printed output. When the printout is renamed, the filename of the printout includes a colon (:). If the output file has lines wider than 132 characters, the right-most characters will not appear in the printout (the printout will be truncated). When the printout is truncated, the filename of the printout includes a minus sign (-).
- Processing Command L The L command moves the display directly back into the editor, where the user may look at or edit the output file.
- Processing Command F The F command copies file IDs from the output search results into an SFILES list. The command prompts for input of a filetype and filemode for the SFILES list being created. The user is also given an opportunity enter a comment line to document the list. When the command is done, the further processing menu is again displayed.
- Further Searching Next are the commands which allow further processing, or searching, in the output file. All four of these commands ignore the documentation section, file IDs, and line numbers during searching. They delete file IDs which no longer have associated lines of text below them; thus, there is a minimum of distracting output for the user to examine. Each of the four commands requires a search string. This string may be entered following the command and one blank. As with the initial search string, any blanks entered after the command name and the single blank following it are counted as part of the search string. If the string is not entered on the same line as the command, the program will prompt for it. Excess blanks may be removed with the DEL or ERASE EOF keys. Care is needed in entering the search string to avoid removing more lines than desired. When processing for each command is completed, the display returns to the editor.

- Processing Command D The first search command in the further processing menu is D. It keeps only those lines which do not contain the specified string.
- Processing Command C The C command deletes lines which begin in column one with the specified string. It can be used to delete FORTRAN comments, which begin with a C in column 1. In this application, the command and string would be entered as C C.
- Processing Command S The S command searches for the specified string and keeps all lines which contain it. This command is analogous to the initial search technique, S.
- Processing Command V Last is V, similar to the V initial search technique. It keeps all lines which contain the search string, not delimited by letters or numbers.

Sample Session

A example of a search session is shown next. The user's entries are shown in lower case, and the program's responses in upper case. Although the session on the terminal would include clearing of the screen, the log presented here is continuous to allow a compact record.

```

s
PLEASE ENTER THE SEARCH STRING. ALL CHARACTERS ENTERED,
INCLUDING BLANKS, ARE PART OF THE STRING. PRESSING ENTER WITHOUT
ENTERING ANY CHARACTERS CAUSES DISPLAY OF HELP DOCUMENT.

common
PLEASE ENTER FILESET IDENTIFIER. EXAMPLE: READ* FORTRAN J
DEFAULT FILENAME: *          DEFAULT FILETYPE: FORTRAN
DEFAULT FILEMODE: MODE OF S PROGRAM, USUALLY A

TO RESUME WORK ON LAST SEARCH OUTPUT, ENTER SEARCH RESULTS.

TO USE A LIST OF FILE ID'S, ENTER THE IDENTIFIER FOR THE
DATASET CONTAINING THE LIST. ITS FILETYPE MUST BE SFILES.
A FILEMODE MAY BE USED TO SHOW WHERE THE SFILES LIST
RESIDES. A SECOND FILEMODE MAY BE ENTERED TO SHOW WHERE
THE FILES TO BE SEARCHED ARE KEPT. DEFAULTS FOR BOTH
FILEMODES ARE THE MODE OF THE S PROGRAM.
EXAMPLE: EXCHANGE SFILES A J

*read*
PLEASE CHOOSE ONE OF THE STRING SEARCH TECHNIQUES BELOW. NULL
INPUT GIVES DEFAULT OF S. (EXPLANATIONS IN PARENTHESES):
S (SEARCH FOR A STRING. LINE NUMBERS SHOWN FOR ALL FILETYPES)
V (SEARCH FOR A VARIABLE; THAT IS, A STRING NOT DELIMITED BY
LETTERS OR NUMBERS. LINE NUMBERS SHOWN FOR ALL FILETYPES)
O (SEARCH FOR A STRING. FOR STRINGS WITH FEW LINES FOUND PER
FILE, IT IS FASTER THAN THE PRECEDING TECHNIQUES. LINE
NUMBERS NOT SHOWN FOR SEVERAL FILETYPES, INCLUDING DATA.)
X (EXIT FROM PROGRAM.)

STRING: 'COMMON'          FILE SET: *READ* FORTRAN A

v
_____ EXREAD FORTRAN A1

```



\_\_\_\_ INTREAD FORTRAN A1  
\_\_\_\_ MREAD FORTRAN A1  
\_\_\_\_ READN FORTRAN A1  
\_\_\_\_ READO FORTRAN A1  
\_\_\_\_ READ100 FORTRAN A1  
\_\_\_\_ READ231 FORTRAN A1  
\_\_\_\_ TESTREAD FORTRAN A1

ONE OF THESE COMMANDS (EXPLANATIONS IN PARENTHESES) MAY BE USED:

X (EXIT FROM THE SEARCH PROGRAM)  
P (PRINT THE FILE NOW)  
L (LOOK AT THE FILE AGAIN)  
F (CREATE SFILES LIST USING FILE ID'S IN SEARCH RESULTS)  
(THE COMMANDS BELOW MAY BE FOLLOWED BY A BLANK AND A STRING)  
D (DELETE LINES CONTAINING A SPECIFIED STRING)  
C (DELETE LINES BEGINNING IN COLUMN 1 WITH A SPECIFIED STRING)  
S (KEEP LINES CONTAINING A SPECIFIED STRING)  
V (KEEP LINES CONTAINING A SPECIFIED STRING NOT PRECEDED OR  
FOLLOWED BY LETTERS OR NUMBERS; THAT IS, A VARIABLE)

SEARCH FILE SET: \*READ\* FORTRAN A  
SEARCH VARIABLE: 'COMMON'

c c

ONE OF THESE COMMANDS (EXPLANATIONS IN PARENTHESES) MAY BE USED:

X (EXIT FROM THE SEARCH PROGRAM)  
P (PRINT THE FILE NOW)  
L (LOOK AT THE FILE AGAIN)  
F (CREATE SFILES LIST USING FILE ID'S IN SEARCH RESULTS)  
(THE COMMANDS BELOW MAY BE FOLLOWED BY A BLANK AND A STRING)  
D (DELETE LINES CONTAINING A SPECIFIED STRING)  
C (DELETE LINES BEGINNING IN COLUMN 1 WITH A SPECIFIED STRING)  
S (KEEP LINES CONTAINING A SPECIFIED STRING)  
V (KEEP LINES CONTAINING A SPECIFIED STRING NOT PRECEDED OR  
FOLLOWED BY LETTERS OR NUMBERS; THAT IS, A VARIABLE)

SEARCH FILE SET: \*READ\* FORTRAN A  
SEARCH VARIABLE: 'COMMON'  
DELETE LEADING STRING: 'C'

s /

ONE OF THESE COMMANDS (EXPLANATIONS IN PARENTHESES) MAY BE USED:

X (EXIT FROM THE SEARCH PROGRAM)  
P (PRINT THE FILE NOW)  
L (LOOK AT THE FILE AGAIN)  
F (CREATE SFILES LIST USING FILE ID'S IN SEARCH RESULTS)  
(THE COMMANDS BELOW MAY BE FOLLOWED BY A BLANK AND A STRING)  
D (DELETE LINES CONTAINING A SPECIFIED STRING)  
C (DELETE LINES BEGINNING IN COLUMN 1 WITH A SPECIFIED STRING)  
S (KEEP LINES CONTAINING A SPECIFIED STRING)  
V (KEEP LINES CONTAINING A SPECIFIED STRING NOT PRECEDED OR  
FOLLOWED BY LETTERS OR NUMBERS; THAT IS, A VARIABLE)

SEARCH FILE SET: \*READ\* FORTRAN A  
SEARCH VARIABLE: 'COMMON'  
DELETE LEADING STRING: 'C'  
SEARCH STRING: '/'

x

Sample  
Search  
Results

The search results produced by the preceding session are shown next. The output file name when printed (SEARCH: COMMON A1) reflects the initial search string.

```
SEARCH FILE SET:          *READ* FORTRAN A
SEARCH VARIABLE:          'COMMON'
DELETE LEADING STRING:    'C'
SEARCH STRING:            '/'
```

```
-----
READN FORTRAN A1
16      COMMON /CONT/ NCOM,NCLASS,NIYRS
88      COMMON /CONT/ NCOM,NCLASS,NIYRS
90      COMMON /POLPAR/ XMAX(31),XTAR(31),XMIN(31),XBMAX(31),
96      COMMON /CONST/ N,NRANK,ICOR(31),AO(31,31),GO(31,31),
101     COMMON /ACT/ X(31),XD(31),XB(31),XBD(31),P(31),PD(31),
105     COMMON /PARDET/ COMMOD(31,3),XMAXC(31,2),XTARC(31,2),
424     COMMON /CONST/ NN,NRANK,ICOR(31),AO(31,31),GO(31,31),
430     COMMON /CONT/ NCOM,NCLASS,NIYRS
-----
READO FORTRAN A1
15      COMMON /CONT/ NCOM,NCLASS
17      COMMON /PARDET/ COMMOD(31,3),XMAXC(31,2),XTARC(31,2),
94      COMMON /CONT/ NCOM,NCLASS
96      COMMON /POLPAR/ XMAX(31),XTAR(31),XMIN(31),XBMAX(31),
102     COMMON /CONST/ N,NRANK,ICOR(31),AO(31,31),GO(31,31),
107     COMMON /ACT/ X(31),XD(31),XB(31),XBD(31),P(31),PD(31),
111     COMMON /PARDET/ COMMOD(31,3),XMAXC(31,2),XTARC(31,2),
423     COMMON /CONST/ NN,NRANK,ICOR(31),AO(31,31),GO(31,31),
429     COMMON /CONT/ NCOM,NCLASS
-----
READ100 FORTRAN A1
15      COMMON /POLPAR/ XMAX(31),PRM(31),XMIN(31),XBMAX(31),
22      COMMON /CONST/ N,NRANK,ICOR(31),AO(31,31),GO(31,31),
27      COMMON /ACT/ X(31),XD(31),XB(31),XBD(31),P(31),PD(31),
31      COMMON /PARDET/ COUNTR(3),COMMOD(31,3),XMAXC(31,2),
-----
READ231 FORTRAN A1
6       COMMON /WORK/ DMMY(90000)
7       COMMON /WORLD / TWP(31),TWST(31),TWTF(31),TWPOP,GWP,GWP70,
9       COMMON /WORLD/ PTOTYPE,PSTART,PJTADJ,POVAL,CSWPF(31)
```

PROGRAMMERS'  
GUIDE TO THE  
SEARCH COMMAND

This section contains technical notes on use of the search command and modification of the programs behind it. The discussion will be most useful to programmers who are familiar with CMS (Conversational Monitor System), CP (Control Program), XEDIT (a CMS editor), and EXEC2 (a language permitting execution of CMS and CP commands from a special type of file). (See references 3, 5, 7, and 6, respectively.)

Search Command  
Limitations

The program requires read/write (R/W) access to an "A" disk. This disk need not be the 191 device. A temporary disk may be created and then accessed as the "A" disk.

The program is not intended to work with initial search strings which contain the @ (at) symbol. The @ symbol is used internally to delimit strings in the initial search routines.

The program truncates printout to a width of 132 characters. This is the maximum length allowed by many printers. The

initial search technique O will not work correctly on lines wider than 132 characters. The initial search techniques S and V will not work correctly on files with lines wider than 249 characters.

The program will not operate properly if it generates an output file too large to be handled by the editor, XEDIT.

If the initial search technique O is used and unnumbered lines are found, further searching in the output file will generally be inaccurate. That is, the unnumbered lines may be kept when they should be deleted or they may be deleted when they should be kept.

When the initial search technique O is used, the user's reader files will be in nohold status after the search. This is not normally a problem.

The V initial search technique and the V further processing command are not currently set up to work with search strings and files containing lower case letters.

#### Program Use

A few aspects of using the search command are most easily explained using more technical language than in the user's guide. Fortunately, these matters are not too frequently encountered.

#### Command Name

The command currently has the abbreviated name S because it is easy to type. Renaming the S EXEC file is sufficient to change the command name. Other possible mnemonic names are SRCH and SEARCH. If the command name SRCH is used, then all of the files making up the search routine may be found with the command FLIST SRCH\* \*XE\*.

#### O Initial Search Technique

A CMS file will not have line numbers useable by the O initial search technique unless the following conditions held when it was last saved: serialization was on, record length was at least 8 greater than truncation length, and record format was fixed. The editor SPF does not in general add line numbers to a file in a form that can be used by XEDIT. This lack of line numbers shows up when lines are inserted into a file using SPF. Not only is line number information not available in the initial search output, but unnumbered lines may be lost in any further searching. This is because unnumbered lines found with the O technique begin in column one of the output file. Normally, found lines begin in column seven of the output, while columns one through five are reserved for line numbers. A line which begins with an underscore in column one is treated as a file identifier. To the extent that it is possible for a found line to begin with an underscore in column one, further searching is uncertain.

#### V Search Techniques

The V initial search technique and the V further processing command both use the same criteria for seeking a string. To be found, the string must not be immediately preceded or followed by a letter or number. This definition includes all

occurrences of a FORTRAN variable that would be recognized by a compiler, as well as some spurious occurrences in comment cards and in some literals (two examples are 'XMAX' and 7H/XMAX/ ), usually in FORMAT statements. Deletion of all lines containing a single quote (using further processing command D) is possible, but the gains from this approach need to be balanced against the possibility of removing direct access read or write statements, which use single quotes.

Searching with the V command gives results which appear like searching for a word in English text. Words in English are normally delimited by blanks, punctuation, or the beginning or end of a line. The V command cannot find a word that is hyphenated and which runs from the end of one line to the beginning of the next line. With this exception, the V command should be reliable in finding "words."

#### Record Format

The S and V initial search techniques are set up to generate output with variable-length records. The number of blocks required to store the output is thus in general reduced. Therefore, there is less likelihood of creating an output file too large to be edited. The O initial search technique creates an output file with fixed-length records, 132 characters wide.

#### Exceeding Storage

The initial search techniques can produce an output file too large to be edited. If this happens, the search program will stop with an error message. The user may then attempt to increase allowed storage using the CP DEFINE STORAGE command. If storage is successfully increased, the user must enter the command IPL CMS and again press ENTER after the message WELCOME TO CMS appears on the terminal screen. If increasing storage doesn't solve the problem, there are three options. First, consider using the V initial search technique where appropriate instead of the S or O techniques. It should give less output. Second, partition the set of files to be searched and only search a fraction of them at a time using two or more SFILES lists. Finally, consider whether the search string could be chosen differently, to give less output on the initial search. For example, if the FORTRAN statement BLOCK DATA is sought, the user could search either for BLOCK or for DATA. From prior experience, the user may expect to find far fewer occurrences of the word BLOCK than of the word DATA. In this case, it would be advisable to initially search for BLOCK and further search for DATA. The search is best carried on in two steps as indicated because the FORTRAN programming language permits a varying number of blanks between the words BLOCK and DATA.

#### Editor

The XEDIT environment is initialized by a profile, SRCHPROF XEDIT, set up for the search command. The PF keys and other parameters are set to commands which give the environment some resemblance to that of the SPF editor. The most notable difference is for the PF keys 5 and 6. PF 5 causes the display of XEDIT line numbers for the output file and changes the VERIFY field to keep the screen full of information without wrap-around of lines. It also changes the setting for PF 11, which scrolls to the right of the output file. Thus, either

before or after pressing PF 5, scrolling once to the right puts column 132 at the right-hand edge of the screen. This setting permits viewing of the entire width of a printable file (132 characters) while giving maximum overlap with the initial screen, which starts with column 1 at the left-hand edge. PF 6 returns to the initial settings.

#### Program Modification

The search routine is written in a modular fashion, to the extent feasible. Therefore, adding to or modifying the program should not be too difficult for the programmer familiar with EXEC2. Obvious candidates for addition are variations on the further searching programs SRCHSTRG EXEC and SRCHVAR EXEC. It will be evident from examination of the places where SRCHSTRG is executed in S EXEC that addition of another search criterion using SRCHSTRG EXEC would require insertion of only a few lines in S EXEC. Additions would be required in three places to: note the new command in the further searching menu; direct the program to a label for the new command when the command is requested; and, following the new label, execute SRCHSTRG with the appropriate parameters.

It might be desirable to modify the program so that it would allow the user to just view (browse) the output file rather than edit it. In that case, use of the editor could be made an option in the further processing menu. Browsing does not require a large storage capacity, even when a large file is viewed. Also, browsing a large file is faster than editing it.

For increased efficiency, the program could be rewritten in assembly language. Another possible change would be to consolidate all of the files which make up the command into one large EXEC file. In that case, the current EXEC files could be called as subroutines. The greatest effort would be required to include the XEDIT macro command SRCHNFNF in the one main EXEC file without a reduction in search speed. The program could easily be augmented to save a backup of the output search results before any further searching command is executed. This would permit the user to retrace one step if a further search command removed too many lines. The V initial search technique and the V further processing commands can be upgraded to handle lower case letters. The tests on the variables &PREFIX and &SUFFIX in SRCHLOCV EXEC and SRCHVAR EXEC would have to be expanded to include lower case letters. It is expected that the Data Services Center will implement several enhancements to the current program.

#### Narrative Program Explanation

The structure of the program behind the search command will be described in three stages. First is a narrative explanation. Next is a flowchart. Finally, the program itself is shown. The program begins by choosing a list of files to be examined. Each file is searched for the desired string or variable and those lines containing it are stored in an output file, preceded by the file ID. The output file may be further processed to keep or delete found lines which contain another specified string. Further searching generally pares the size of the output file. Only documentation lines are added by

further searching. It is possible that all of the lines found during the initial search in a particular file may be removed by further searching. When this happens, the program removes that file's ID from the output. Removal of file IDs which are no longer relevant makes the output more compact and easier to read. A record of processing commands and search strings is placed in the output file. The further processing menu is reentered until the user desires to exit from the program.

### Program Flowcharts

The following pages contain flowcharts of the programs which make up the search command. The flowcharting symbols and usages are standard (9). The boxes in the flowcharts typically represent from 1 to 5 program statements. Menus and reminders displayed by the program at the terminal are not included in the flowcharts. Many of the checks for erroneous terminal input are also left out of the flowcharts.

Figure 1. Flowchart of Program S EXEC

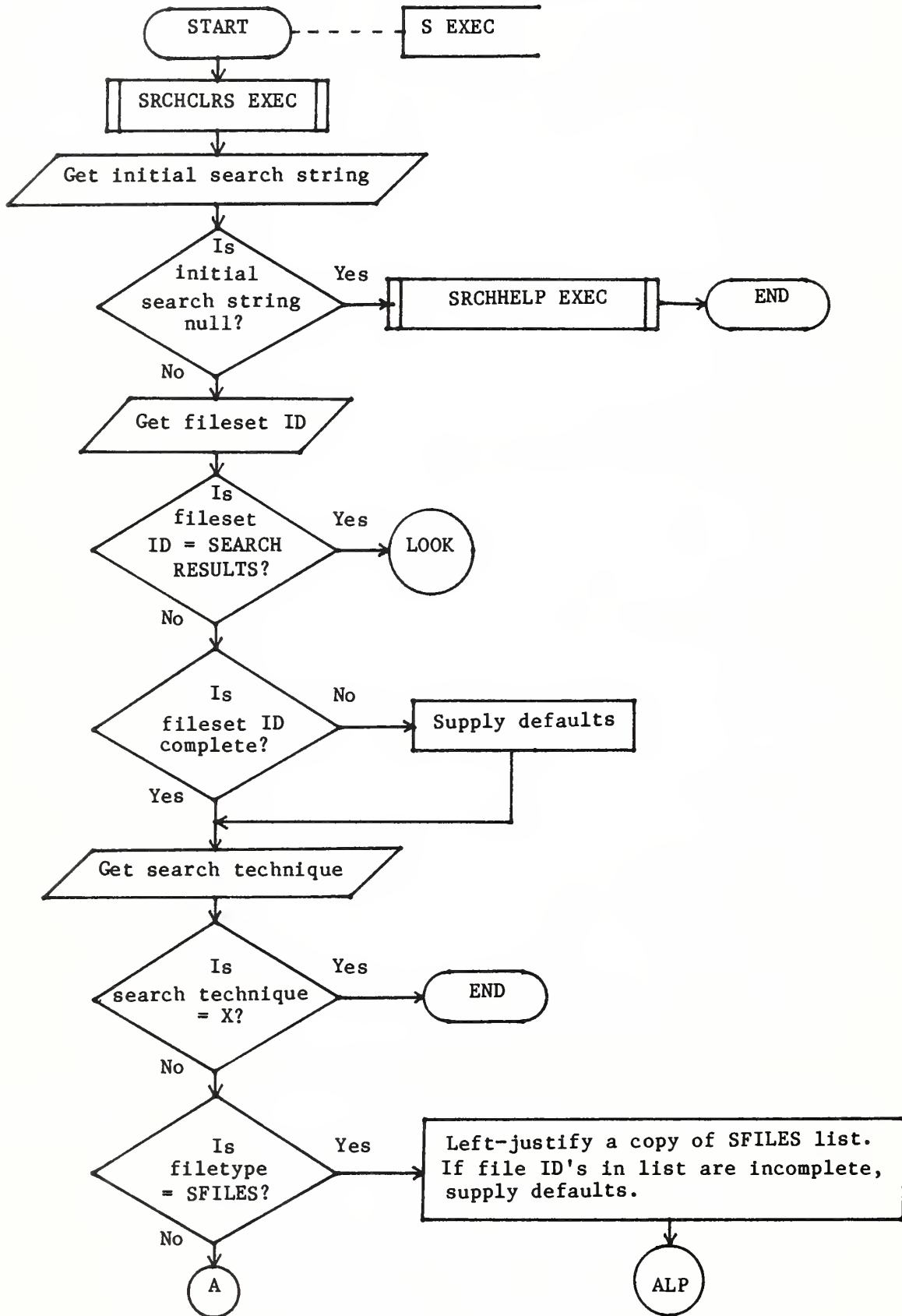


Figure 1. Flowchart of Program S EXEC -- Continued

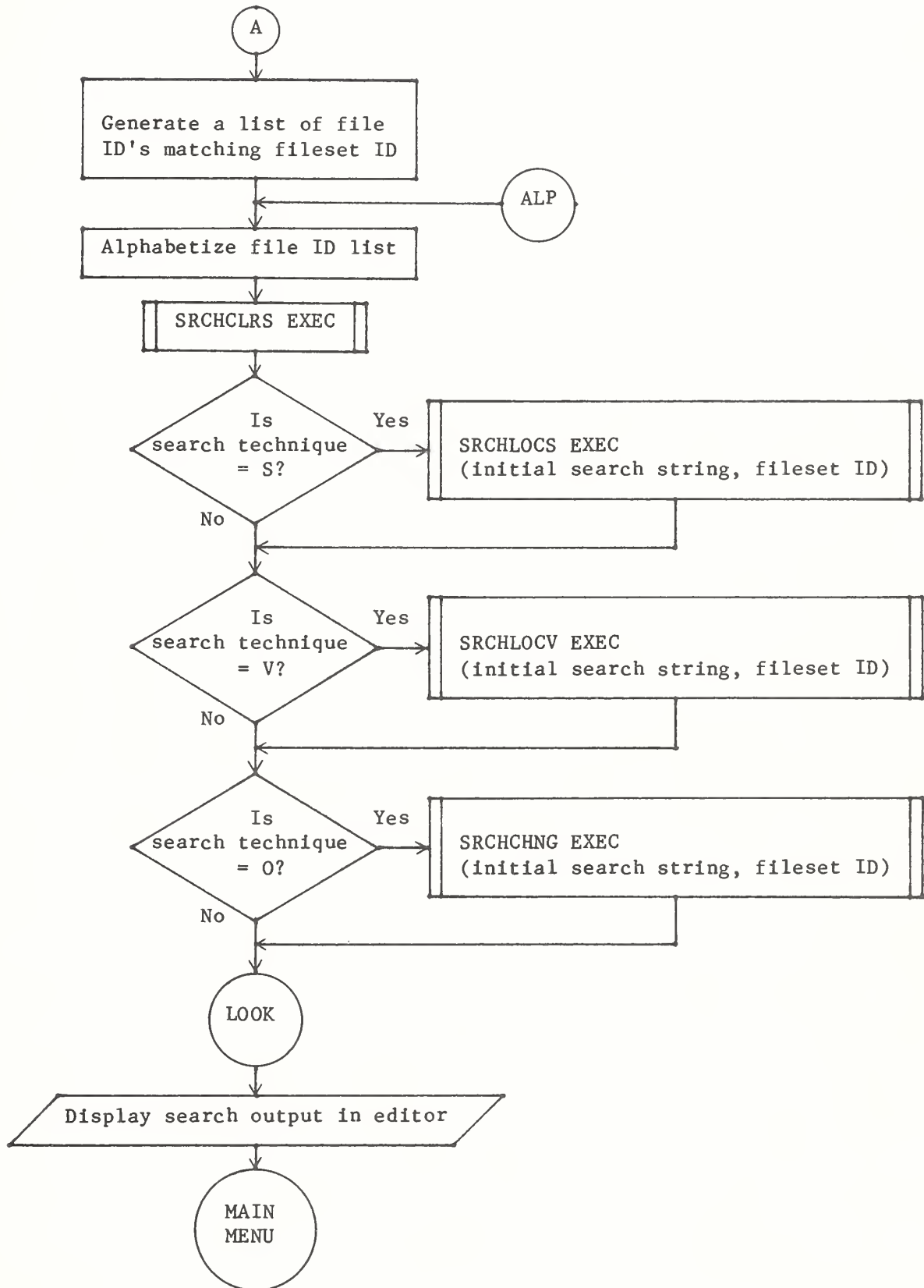




Figure 1. Flowchart of Program S EXEC -- Continued

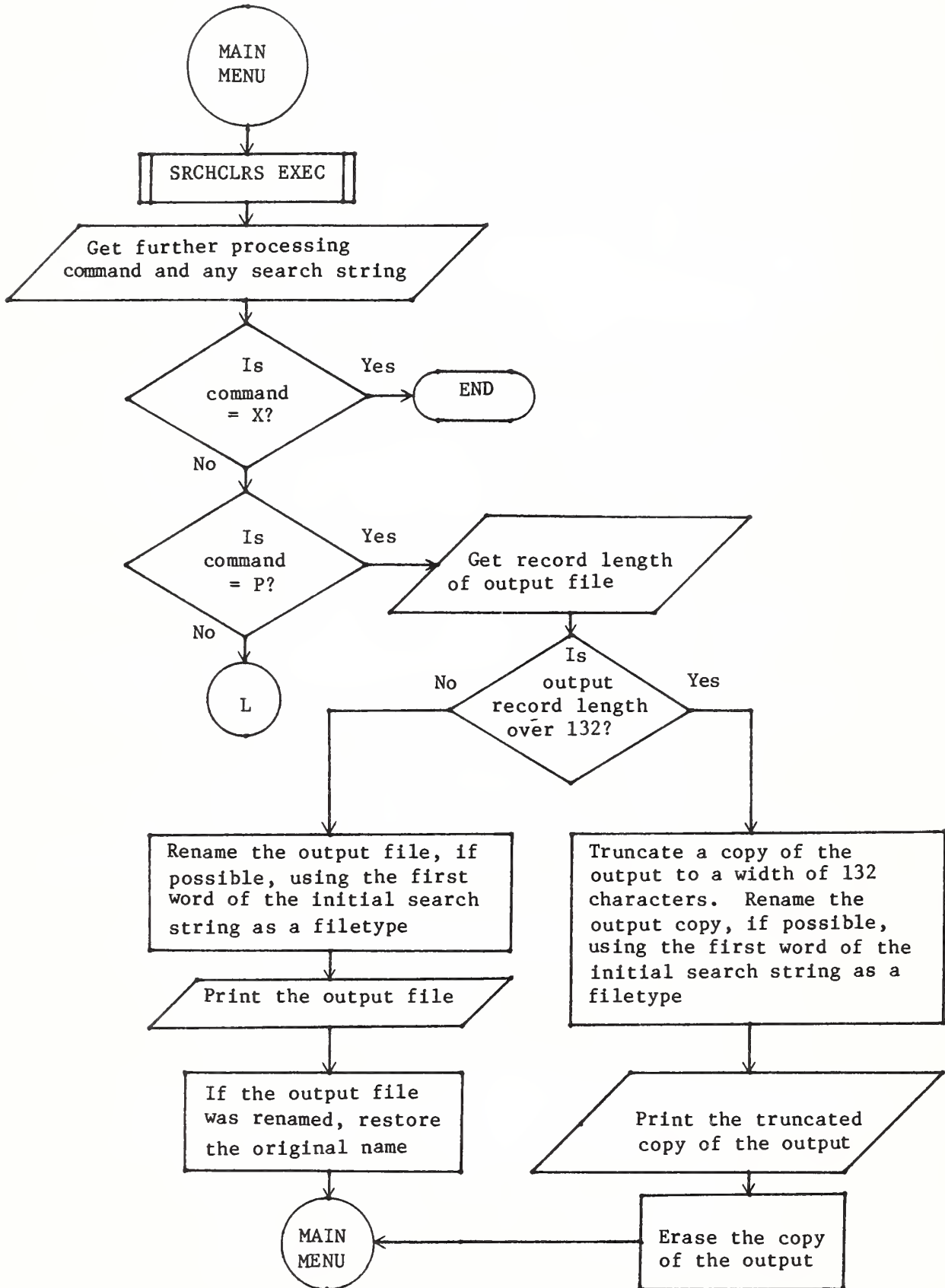


Figure 1. Flowchart of Program S EXEC -- Continued

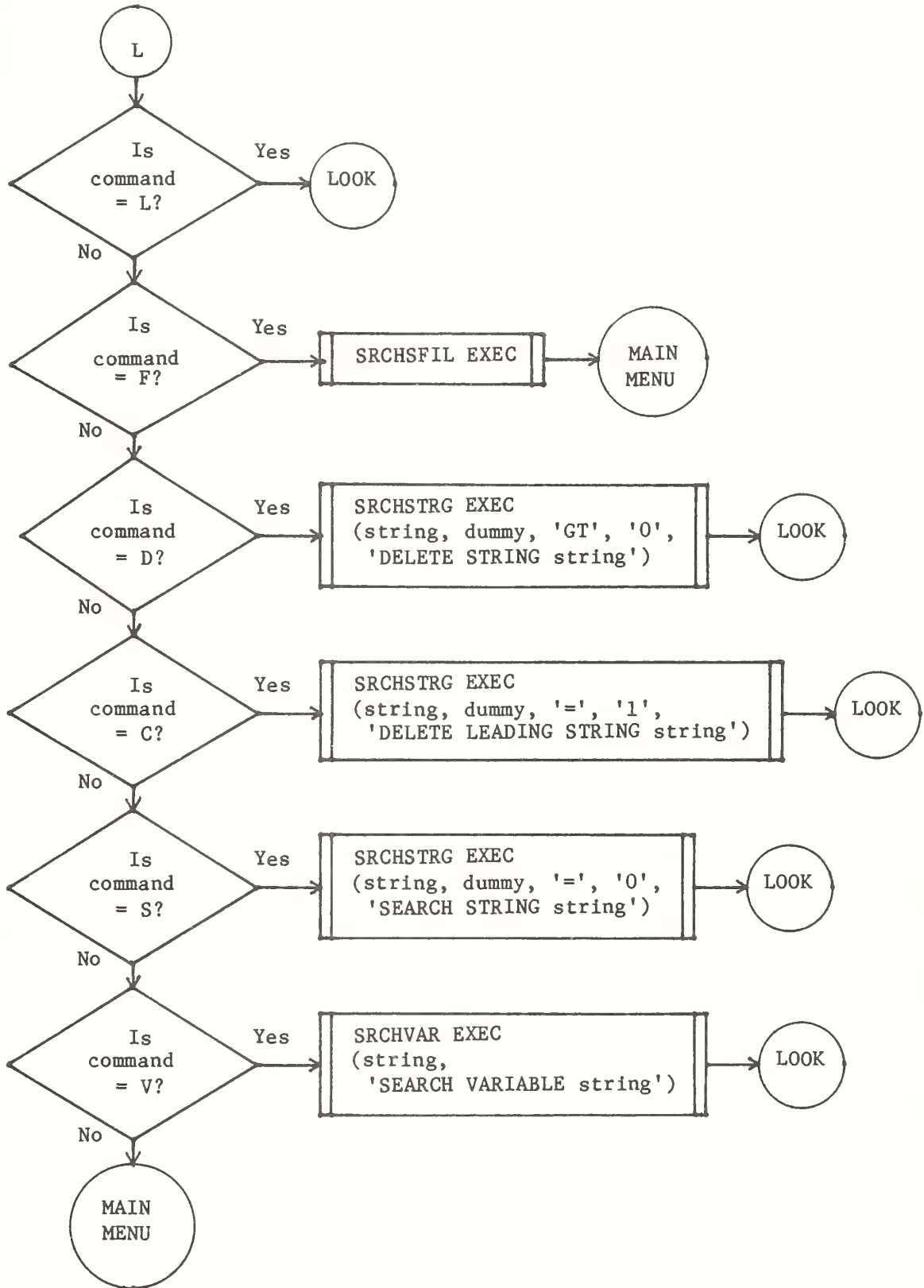


Figure 2. Flowchart of Program SRCHLOCS EXEC

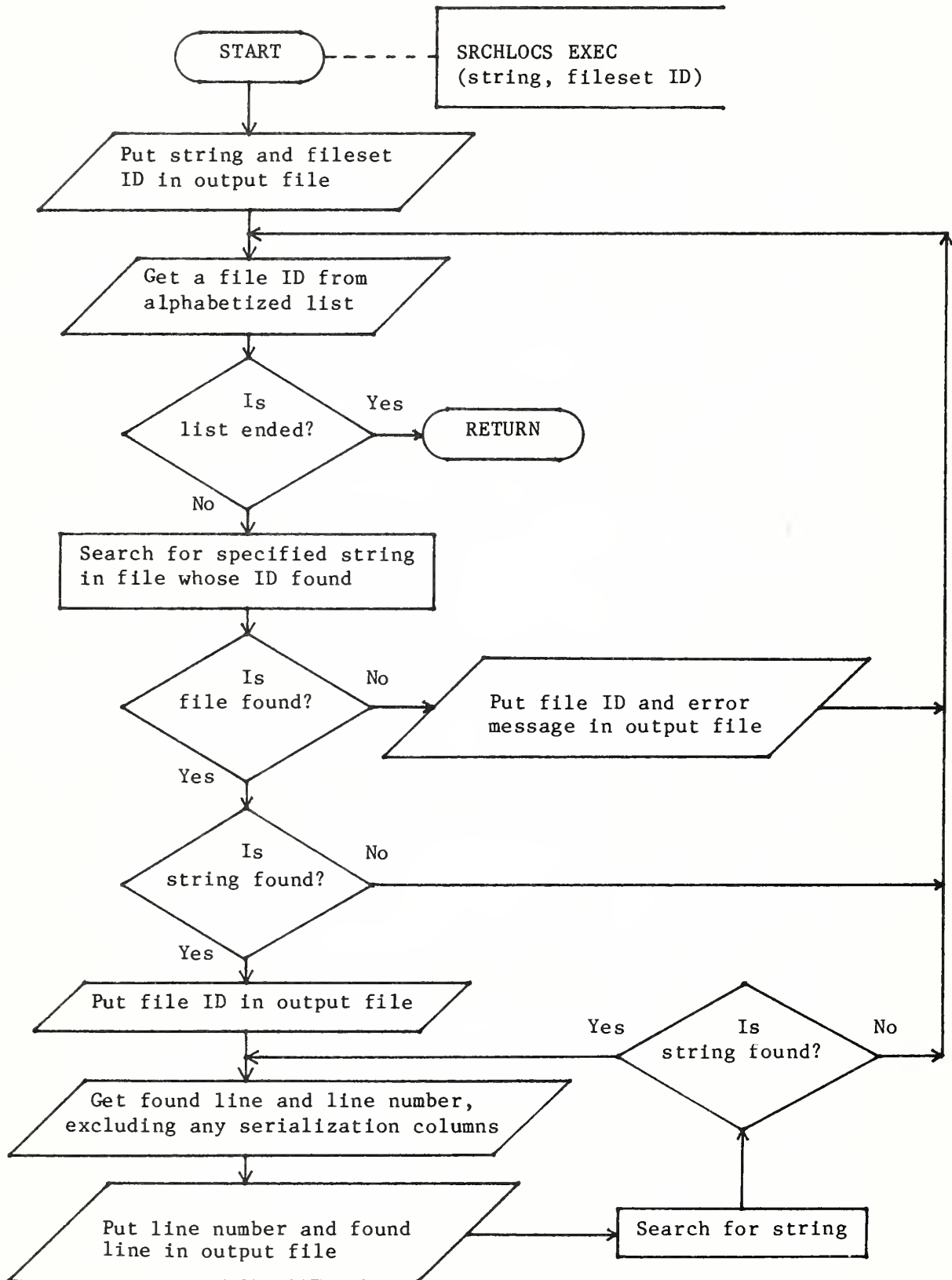


Figure 3. Flowchart of Program SRCHLOCV EXEC

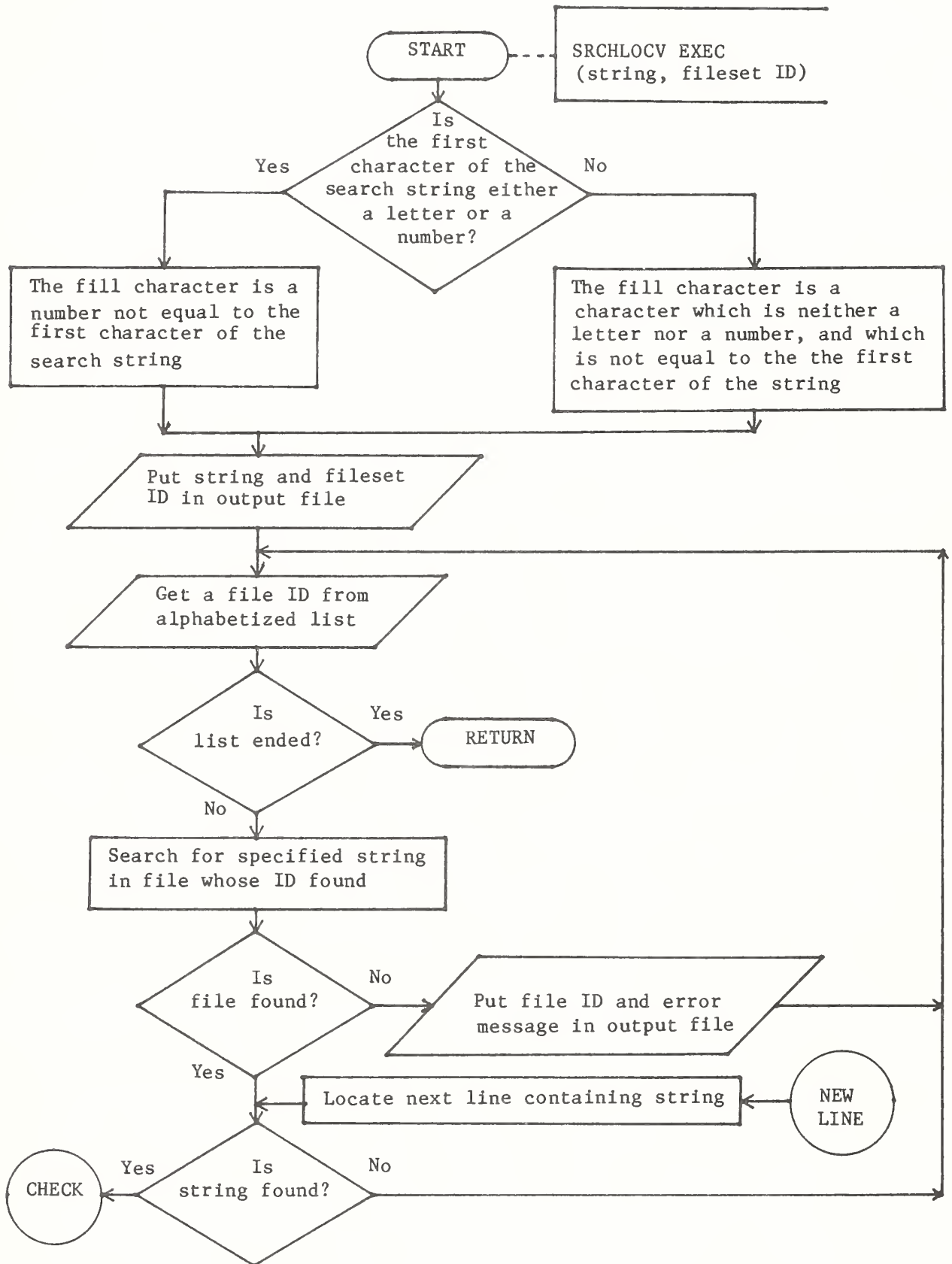


Figure 3. Flowchart of Program SRCHLOCV EXEC -- Continued

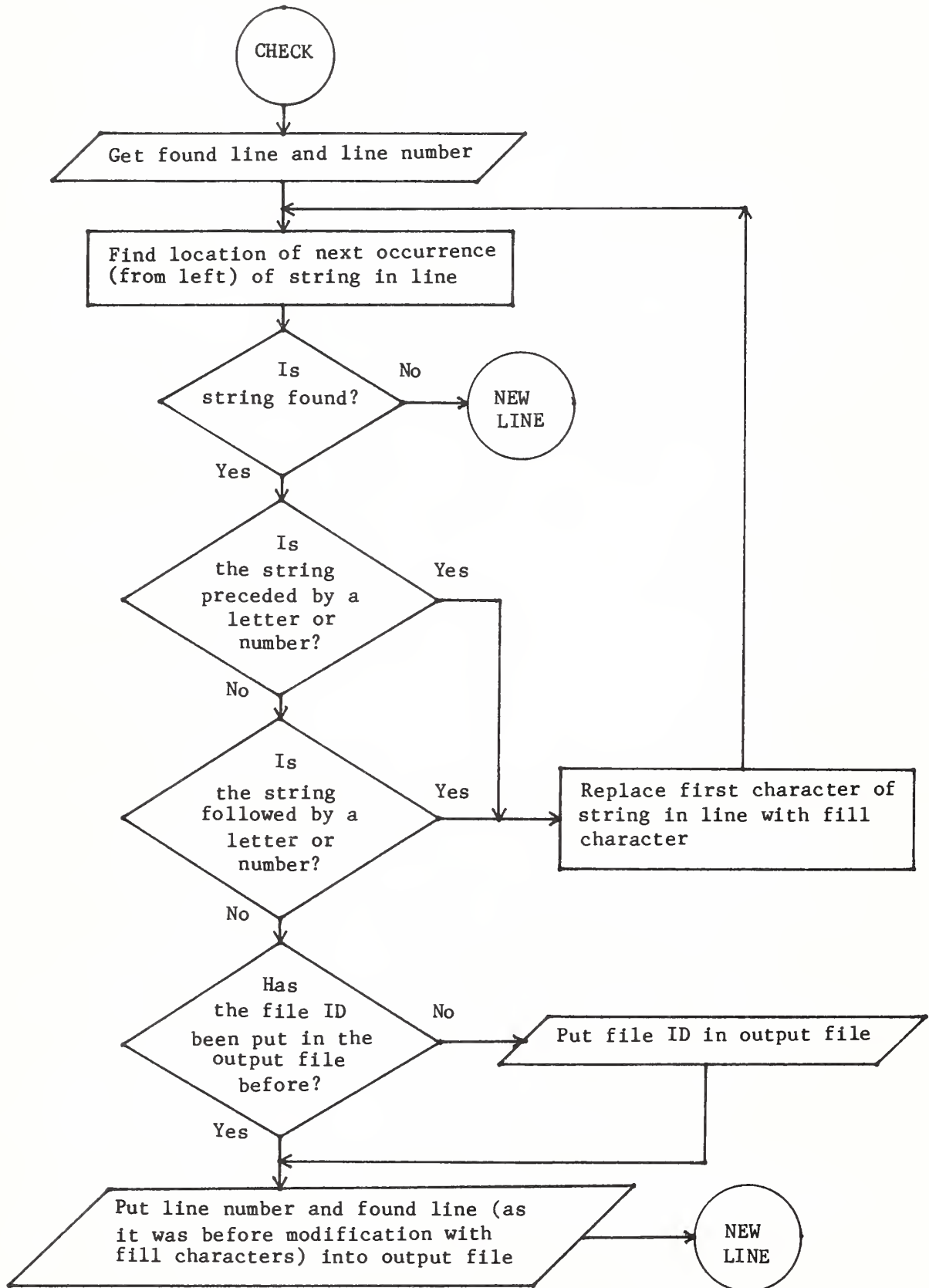


Figure 4. Flowchart of Program SRCHCHNG EXEC

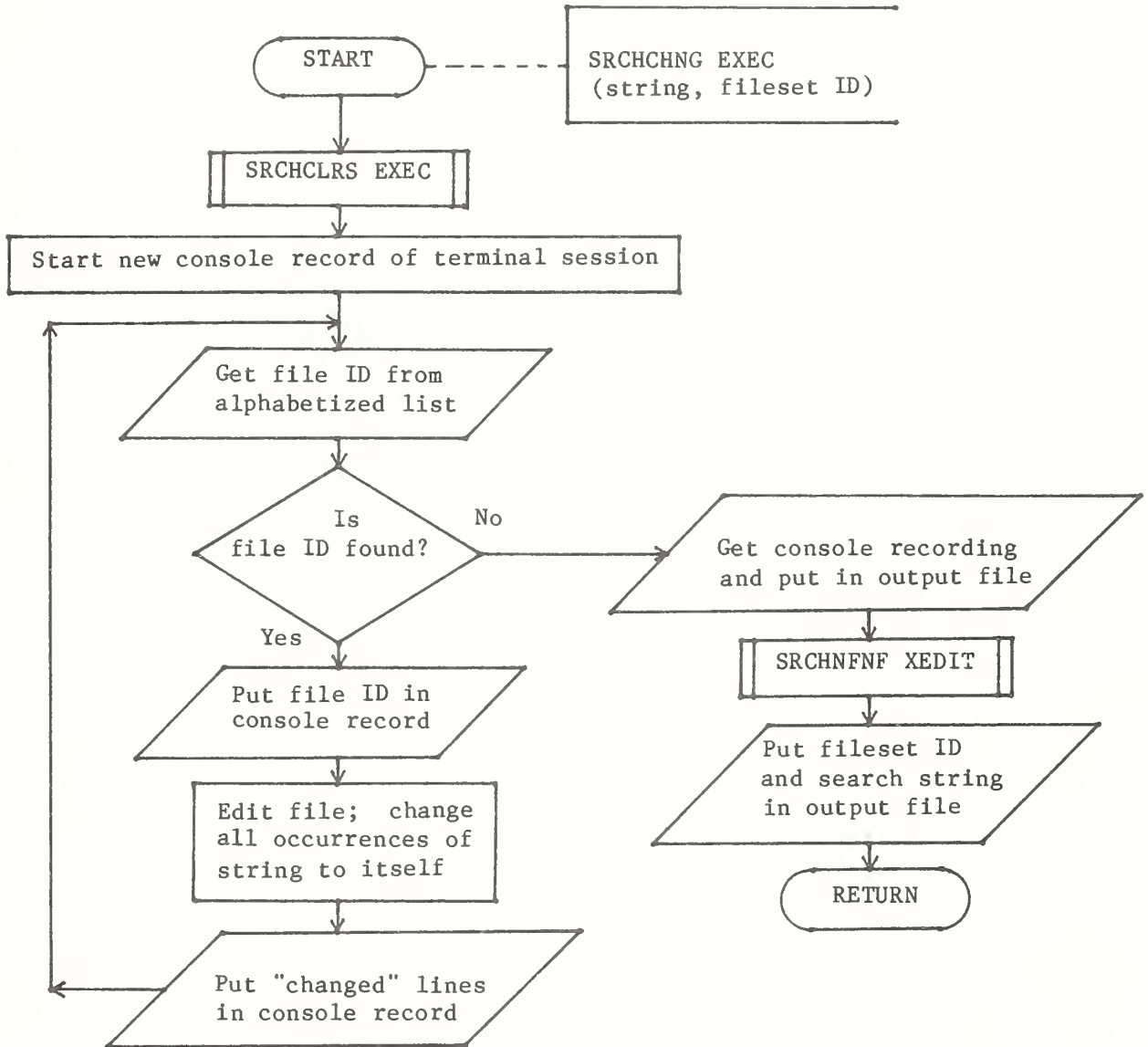


Figure 5. Flowchart of Program SRCHCLRS EXEC

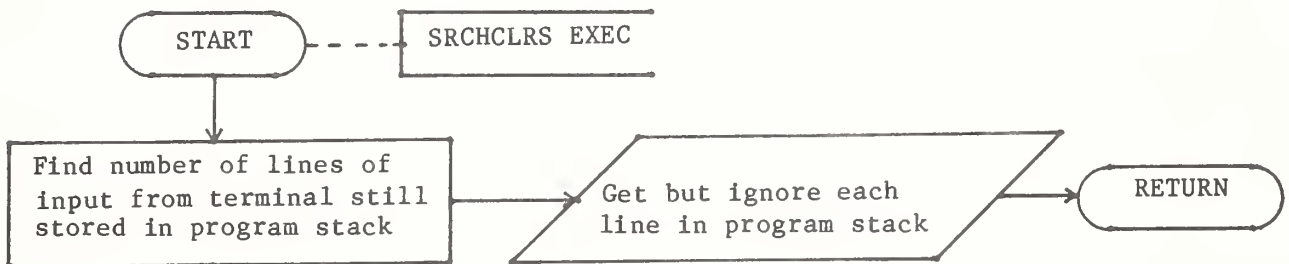


Figure 6. Flowchart of Program SRCHNFNF XEDIT

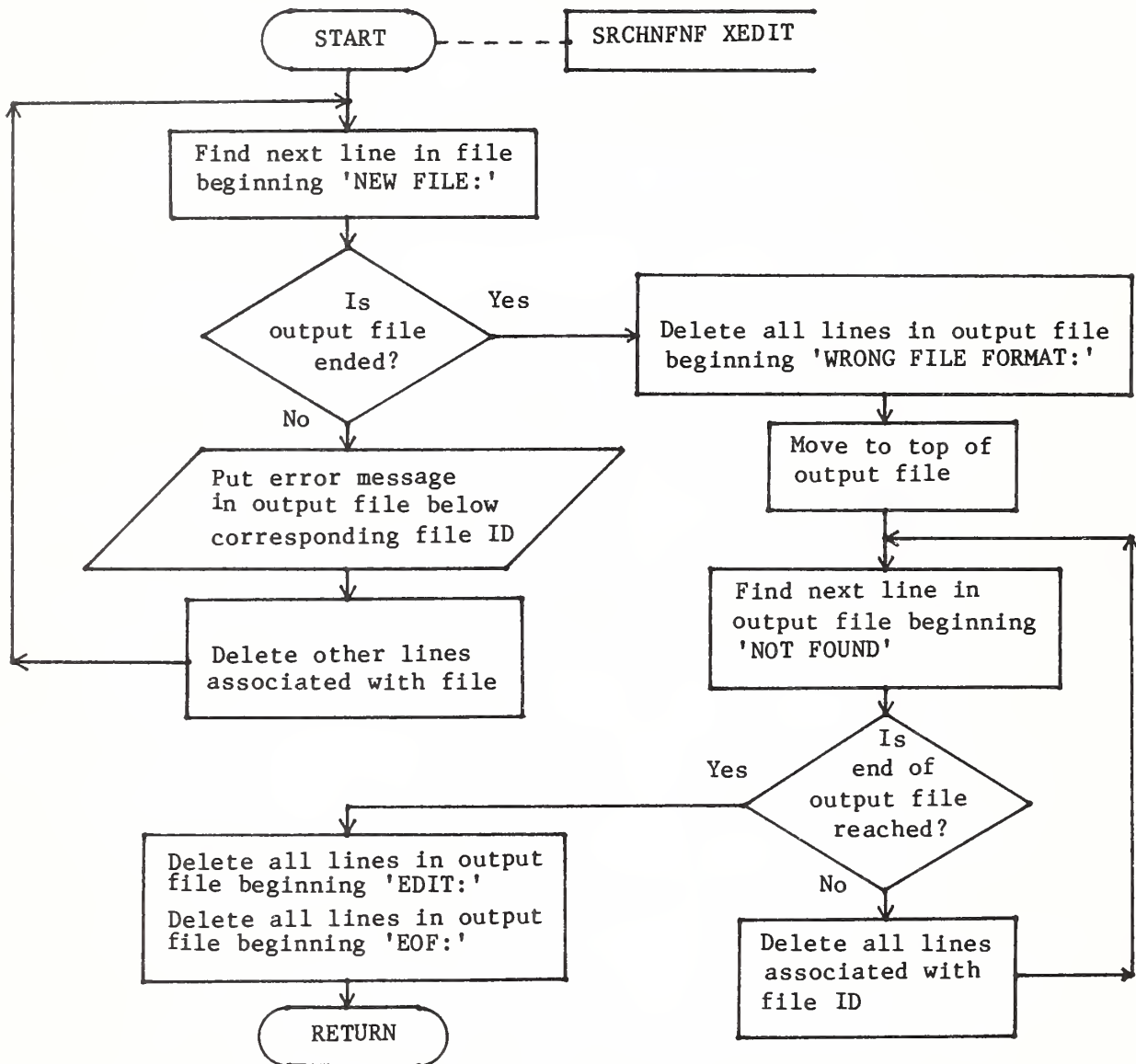


Figure 7. Flowchart of Program SRCHHELP EXEC

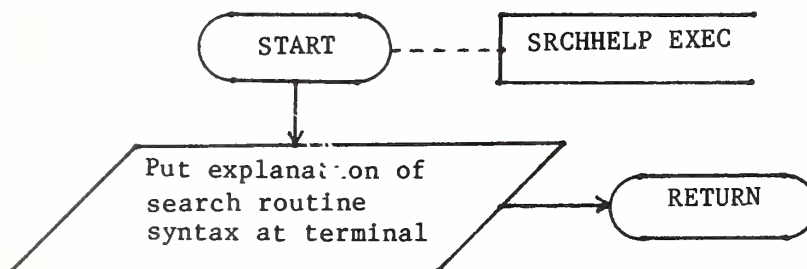


Figure 8. Flowchart of Program SRCHSFIL EXEC

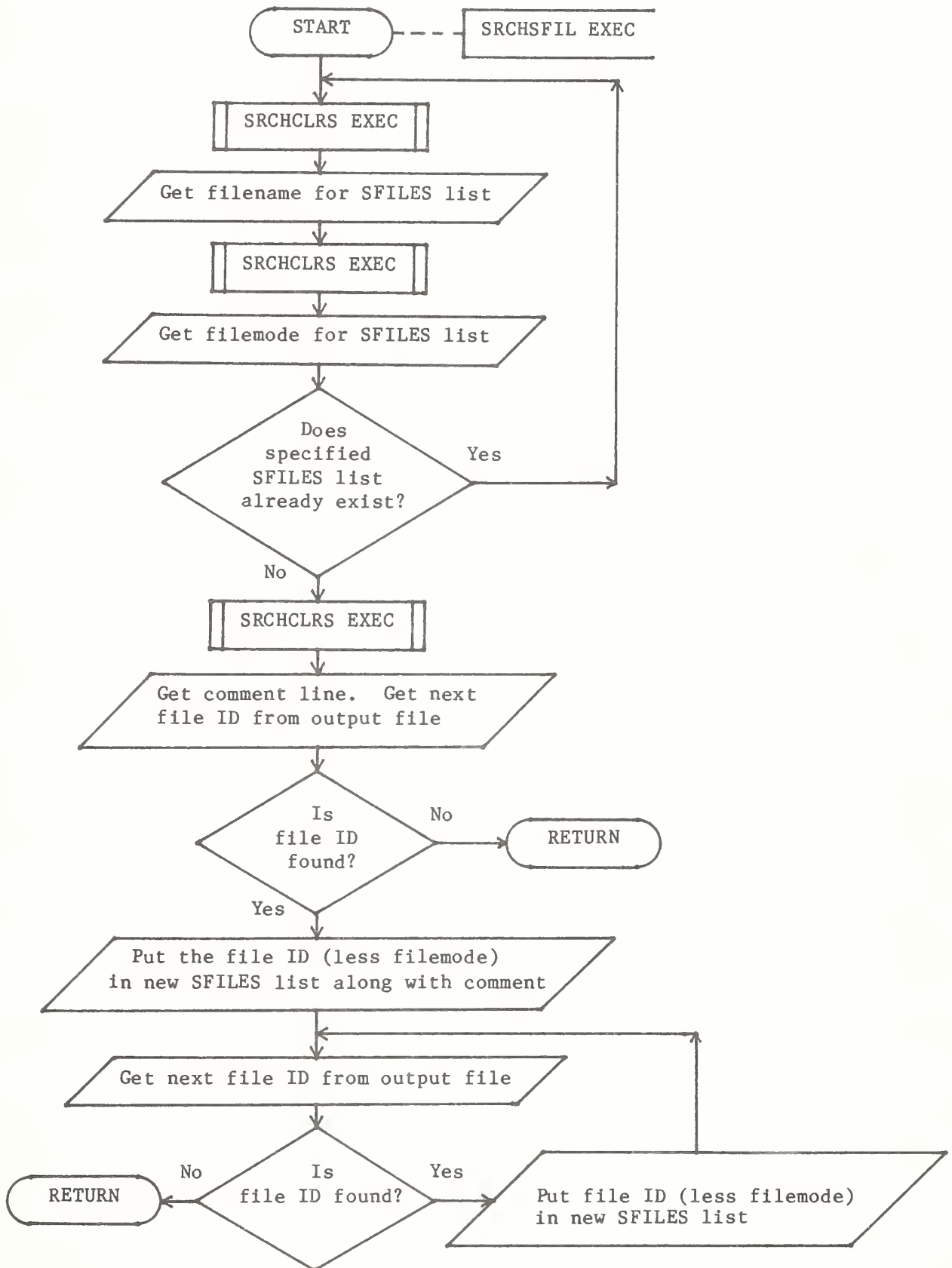




Figure 9. Flowchart of Program SRCHSTRG EXEC

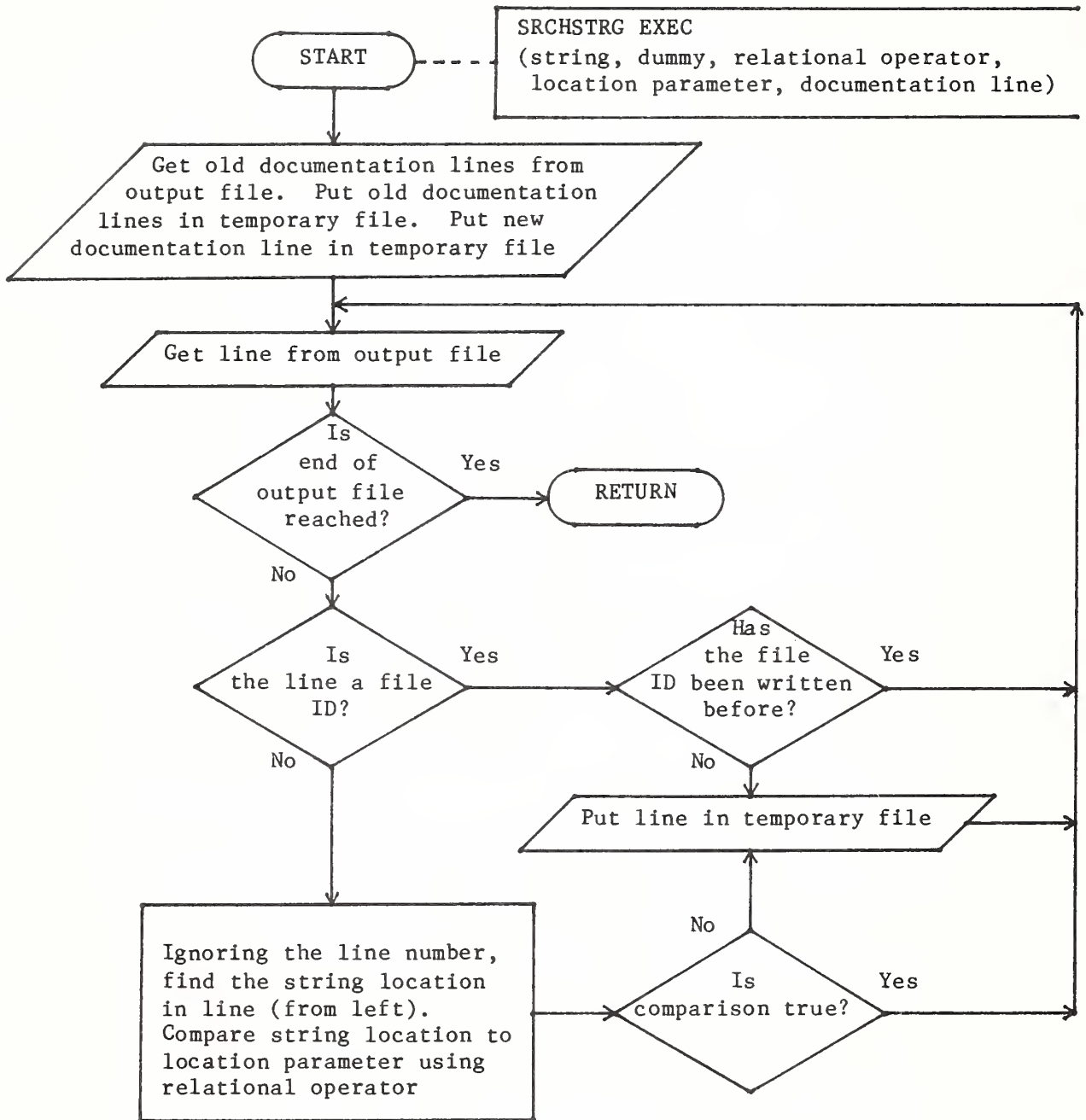


Figure 10. Flowchart of Program SRCHVAR EXEC

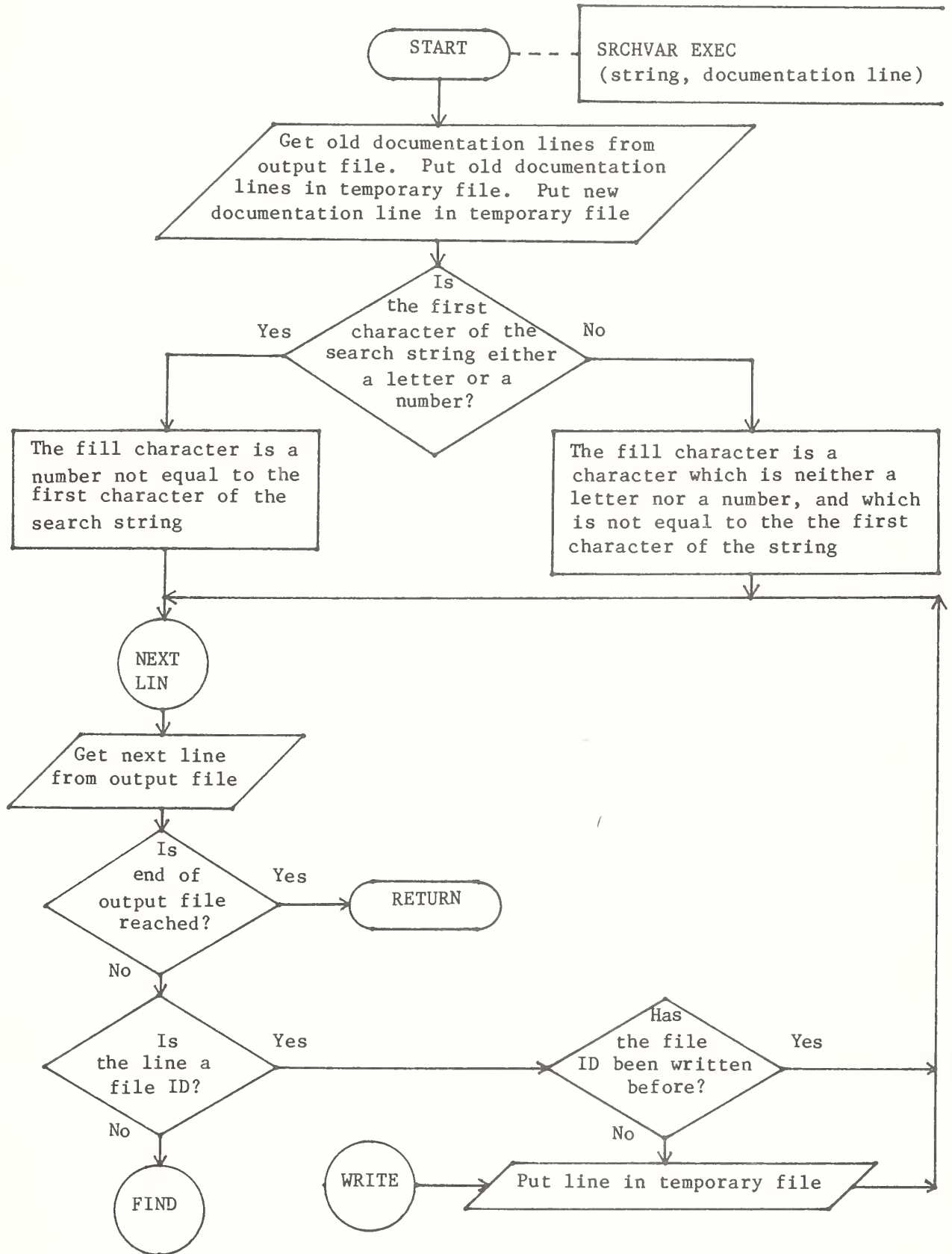
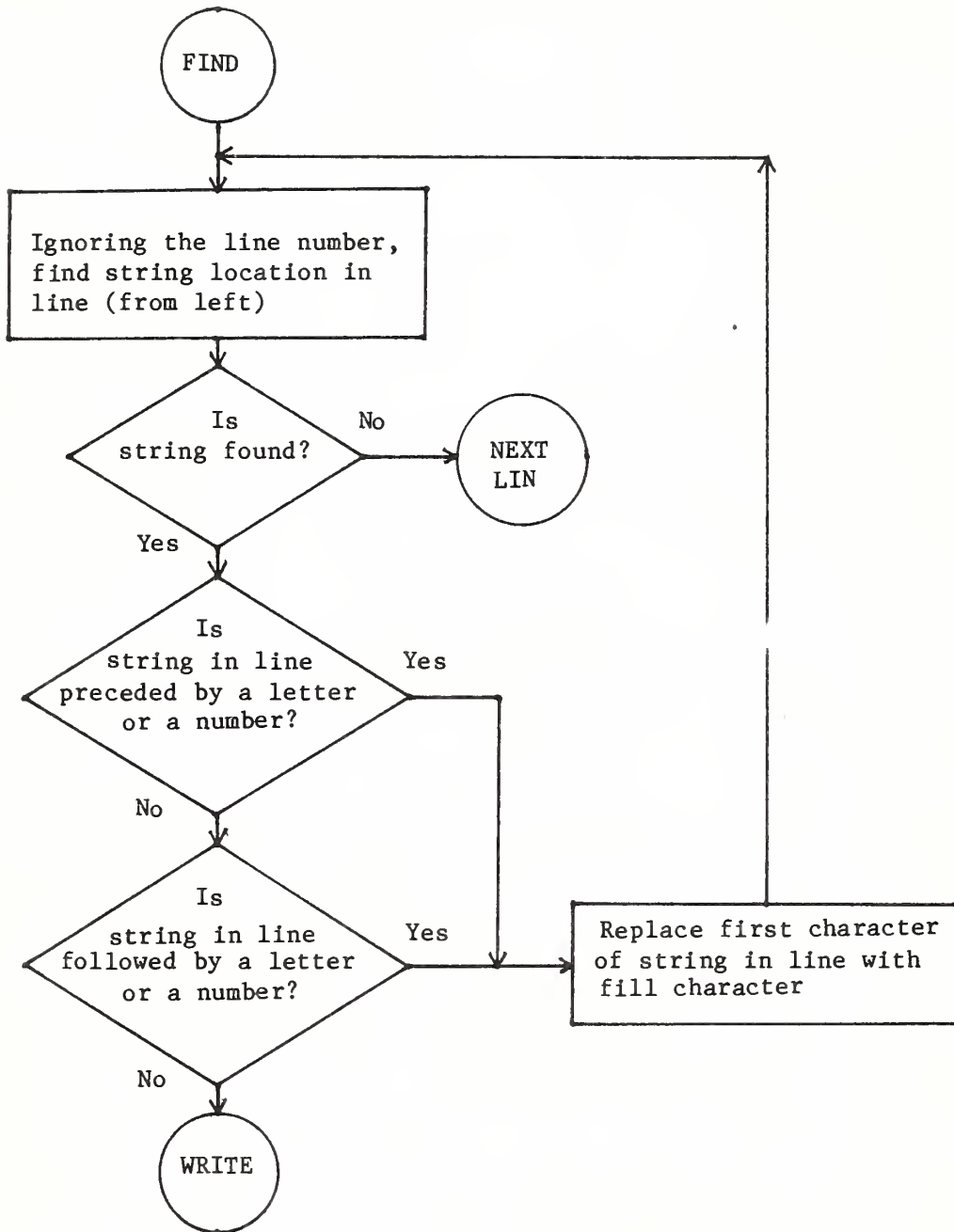


Figure 10. Flowchart of Program SRCHVAR EXEC -- Continued



Program Listings

The program files which make up the search command are listed in this section, beginning with the main program file, S EXEC. Each file name is underlined.

S EXEC

```
&TRACE ERR
* ----- FILE IDENTIFIER:  S EXEC
* ----- THIS IS THE MAIN PROGRAM OF A SEARCH ROUTINE.
* ----- IT SEARCHES THROUGH A SPECIFIED SET OF CMS FILES,
* ----- LOCATING LINES WHICH CONTAIN A DESIRED CHARACTER
* ----- STRING.  A COPY OF EACH LOCATED LINE IS PLACED IN AN
* ----- OUTPUT FILE, PRECEDED BY THE IDENTIFIER OF THE
* ----- FILE IN WHICH IT WAS FOUND AND THE NUMBER OF THE
* ----- LINE RELATIVE TO THE TOP OF THE FILE IN WHICH IT
* ----- WAS FOUND.  THE STRING MAY BE SEARCHED FOR AS A
* ----- "VARIABLE" (NEITHER IMMEDIATELY PRECEDED NOR
* ----- FOLLOWED BY A LETTER OR A NUMBER).  THE OUTPUT FILE
* ----- MAY BE SEARCHED FOR ANOTHER STRING OR "VARIABLE."
* ----- LINES CONTAINING OR BEGINNING WITH A SPECIFIED
* ----- STRING MAY BE DELETED FROM THE OUTPUT FILE.
* ----- THE OUTPUT FILE MAY BE PRINTED, WITH AN INFORMATIVE
* ----- IDENTIFIER.  THE FILE SET, SEARCH COMMANDS, AND
* ----- SEARCH STRINGS ARE DOCUMENTED AT THE TOP OF THE
* ----- OUTPUT FILE AND DISPLAYED AT THE TERMINAL.
* -----
* ----- FURTHER DOCUMENTATION IS AVAILABLE IN THE FORM OF
* ----- AN ON-LINE HELP DOCUMENT AS WELL AS A MORE DETAILED
* ----- REPORT, WHICH MAY BE OBTAINED FROM THE AUTHOR.
* ----- AUTHOR:  RALPH SEELEY
* ----- PHONE:  (202) 447-8815
* ----- ADDRESS: 500 12TH ST. SW  WASHINGTON D.C. 20250
* ----- VERSION: 3-20-84
* ----- LAST PROGRAM MODIFICATION: 3-20-84 RMS
* -----
* -----
* ----- CLRSCRN
* ----- SAVE SEARCH STRING;  &ARGSTRING OVERWRITTEN LATER
&FIRSTSTRING = &ARGSTRING
* ----- SAVE FIRST WORD OF SEARCH STRING FOR PRINT RENAMING
&FIRSTWORD  =&1
* ----- IF SEARCH STRING NOT ENTERED WITH COMMAND, GET IT NOW
&IF &N GT 0 &GOTO -ATEST
&BEGPRINT -SRCHSTRGPROMPT
PLEASE ENTER THE SEARCH STRING.  ALL CHARACTERS ENTERED,
INCLUDING BLANKS, ARE PART OF THE STRING.  PRESSING ENTER
WITHOUT TYPING ANY CHARACTERS CAUSES DISPLAY OF HELP DOCUMENT.
-SRCHSTRGPROMPT
* ----- CLEAR ANY EXTRANEIOUS INPUT FROM STACK
EXEC SRCHCLRS
&READ STRING &FIRSTSTRING
* ----- IF NO SEARCH STRING IS ENTERED, DISPLAY HELP DOCUMENT
&IF /&FIRSTSTRING = / &GOTO -HELP
* ----- SAVE FIRST WORD OF SEARCH STRING FOR PRINT RENAMING
&STACK &FIRSTSTRING
&READ ARGS
```

```

&FIRSTWORD = &1
-AATEST CLRSCRN
* ----- @ IS USED INTERNALLY BY INITIAL SEARCH ROUTINES
&ATLOC = &LOCATION OF @ &FIRSTSTRING
&IF &ATLOC = 0 &GOTO -SETUP
&PRINT THE @ SIGN MAY NOT BE USED IN THE INITIAL SEARCH STRING
&EXIT
-SETUP &B4 = &CONCAT OF &BLANK &BLANK &BLANK &BLANK
&B5 = &CONCAT OF &BLANK &B4
&BEGPRINT -FILESETMENU
PLEASE ENTER FILESET IDENTIFIER. EXAMPLE: READ* FORTRAN J
DEFAULT FILENAME: * DEFAULT FILETYPE: FORTRAN
DEFAULT FILEMODE: MODE OF S PROGRAM, USUALLY A

TO RESUME WORK ON LAST SEARCH OUTPUT, ENTER SEARCH RESULTS.

TO USE A LIST OF FILE IDS, ENTER THE IDENTIFIER FOR THE
DATASET CONTAINING THE LIST. ITS FILETYPE MUST BE SFILES.
A FILEMODE MAY BE USED TO SHOW WHERE THE SFILES LIST
RESIDES. A SECOND FILEMODE MAY BE ENTERED TO SHOW WHERE
THE FILES TO BE SEARCHED ARE KEPT. DEFAULTS FOR BOTH
FILEMODES ARE THE MODE OF THE S PROGRAM.
EXAMPLE: EXCHANGE SFILES A J
-FILESETMENU
* ----- GET IDENTIFICATION OF FILES THROUGH WHICH TO SEARCH
&READ ARGS
&IF &N GE 2 &IF &1 = SEARCH &IF &2 = RESULTS &GOTO -LOOK
* ----- IF FILE IDENTIFIER NOT ENTERED, SUPPLY DEFAULTS
&FNSET = *
&IF &N GE 1 &FNSET = &1
&FTSET = FORTRAN
&IF &N GE 2 &FTSET = &2
&FMSET = &SUBSTR OF &FILEMODE 1 1
&IF &N GE 3 &FMSET = &3
-GETSTECH
CLRSCRN
&BEGPRINT -SRCHTECHMENU
PLEASE CHOOSE ONE OF THE STRING SEARCH TECHNIQUES BELOW. NULL
INPUT GIVES DEFAULT OF S. (EXPLANATIONS IN PARENTHESES):
S (SEARCH FOR A STRING. LINE NUMBERS SHOWN FOR ALL FILETYPES)
V (SEARCH FOR A VARIABLE; THAT IS, A STRING NOT DELIMITED BY
LETTERS OR NUMBERS. LINE NUMBERS SHOWN FOR ALL FILETYPES)
O (SEARCH FOR A STRING. FOR STRINGS WITH FEW LINES FOUND PER
FILE, IT IS FASTER THAN THE PRECEDING TECHNIQUES. LINE
NUMBERS NOT SHOWN FOR SEVERAL FILETYPES, INCLUDING DATA.)
X (EXIT FROM PROGRAM.)

-SRCHTECHMENU
* ----- DISPLAY SEARCH STRING AND FILESET ID TO CONFIRM ENTRY
&QUOTEDSTRING = &CONCAT OF STRING: &BLANK ' &FIRSTSTRING '
&PRINT &B5 &B5 &QUOTEDSTRING &B4 FILE SET: &FNSET &FTSET &FMSET
* ----- GET SEARCH TECHNIQUE; IF NULL, SUPPLY DEFAULT (=S)
&READ VAR &STECH
&IF /&STECH = /X &EXIT
&IF /&STECH = / &STECH = S
&IF &STECH NE S &IF &STECH NE V &IF &STECH NE O &GOTO -GETSTECH

```

```

&IF &FTSET = SFILES &GOTO -SUBSET
* ----- GENERATE LIST OF EXISTING FILES WHICH MATCH FILESET ID
LISTF &FNSET &FTSET &FMSET (EXEC
&IF &RC GT 0 &EXIT
SET CMSTYPE HT
* ----- REMOVE VARIABLES PRECEDING FILE NAMES IN CMS EXEC A
&BEGSTACK 3
V OFF
C /&1 &2// *
FILE
XEDIT CMS EXEC A (NOPROF NOSC
&GOTO -SEARCHTECH
-SUBSET
* ----- 3RD WORD OF FILESET ID FOR SFILES LIST IS MODE OF LIST
&FMSFILES = &FMSET
&FMSET = &SUBSTR OF &FILEMODE 1 1
* ----- 4TH WORD OF FILESET ID IS MODE OF FILES TO BE SEARCHED
&IF &N GE 4 &FMSET = &4
SET CMSTYPE HT
EXECIO 1 DISKR &FNSET SFILES &FMSFILES
&READ VAR &FN &FT &FM
* ----- IF FILEMODE OR FILETYPE MISSING, SUPPLY DEFAULTS
&IF /&FT = / &FT = FORTRAN
&IF /&FM = /* &FM = &FMSET
&IF /&FM = / &FM = &FMSET
ERASE CMS EXEC A
* ----- CREATE CMS EXEC A WITH RECFM F AND LRECL 80
EXECIO 1 DISKW CMS EXEC A 1 F 80 (ST &FN &FT &FM
EXECIO 1 DISKR &FNSET SFILES &FMSFILES
&LOOP -ENDALIGN WHILE &RC = 0
&READ VAR &FN &FT &FM
&IF /&FT = / &FT = FORTRAN
&IF /&FM = / &FM = &FMSET
EXECIO 1 DISKW CMS EXEC A (ST &FN &FT &FM
-ENDALIGN EXECIO 1 DISKR &FNSET SFILES &FMSFILES
* ----- USE FILEMODE OF SFILES LIST IN OUTPUT DOCUMENTATION
&FMSET = &FMSFILES
-SEARCHTECH
SET CMSTYPE HT
* ----- ALPHABETIZE LIST OF FILES FROM CMS EXEC A
&STACK 1 80
SORT CMS EXEC A SEARCH SORTFILE A
EXEC SRCHCLRS
* ----- PASS FILESET ID TO INITIAL SEARCH ROUTINE
&STACK &FNSET &FTSET &FMSET
SET CMSTYPE HT
* ----- SEARCH FOR STRING USING SELECTED TECHNIQUE
&IF &STECH = S EXEC SRCHLOCS &FIRSTSTRING
&IF &STECH = V EXEC SRCHLOCV &FIRSTSTRING
&IF &STECH = O EXEC SRCHCHNG &FIRSTSTRING
SET CMSTYPE RT
-LOOK
XEDIT SEARCH RESULTS A (PROF SRCHPROF
-MAINMENU
&BEGPRINT -PRTMAINMENU
ONE OF THESE COMMANDS (EXPLANATIONS IN PARENTHESES) MAY BE USED:

```

X (EXIT FROM THE SEARCH PROGRAM)  
 P (PRINT THE FILE NOW)  
 L (LOOK AT THE FILE AGAIN)  
 F (CREATE SFILES LIST USING FILE IDS IN SEARCH RESULTS)  
 (THE COMMANDS BELOW MAY BE FOLLOWED BY A BLANK AND A STRING)  
 D (DELETE LINES CONTAINING A SPECIFIED STRING)  
 C (DELETE LINES BEGINNING IN COLUMN 1 WITH A SPECIFIED STRING)  
 S (KEEP LINES CONTAINING A SPECIFIED STRING)  
 V (KEEP LINES CONTAINING A SPECIFIED STRING NOT PRECEDED OR  
 FOLLOWED BY LETTERS OR NUMBERS; THAT IS, A VARIABLE)

-PRTMMAINMENU

\* ----- DISPLAY DOCUMENTATION STORED AT TOP OF OUTPUT FILE

```

EXECIO 1 DISKR SEARCH RESULTS A 1
&LOOP -DOCUEND WHILE &RC = 0
  &READ STRING &DOCUOLD
  &DASHLOC = &LOCATION OF ----- &DOCUOLD
  &IF &DASHLOC = 6 &GOTO -GETCOMMAND
  SET CMSTYPE RT
  &PRINT &B5 &DOCUOLD
  SET CMSTYPE HT

```

-DOCUEND EXECIO 1 DISKR SEARCH RESULTS A

-GETCOMMAND

```

FINIS SEARCH RESULTS A
SET CMSTYPE RT

```

\* ----- CLEAR ANY EXTRANEIOUS INPUT FROM STACK

```

EXEC SRCHCLRS
&READ STRING &RESPONSE
&IF /&RESPONSE = / &GOTO -BADINPUT
&IF &RESPONSE = X &EXIT
&IF &RESPONSE = P &GOTO -PRINT
&IF &RESPONSE = L &GOTO -LOOK
&IF &RESPONSE = F &GOTO -MAKESFILES

```

\* ----- AVOID ACCIDENTAL INPUT; FOR EXAMPLE, FROM PF KEY

```

&SECONDCHAR = &SUBSTR OF &RESPONSE 2 1
&IF /&SECONDCHAR NE /&BLANK &GOTO -BADINPUT
&INPUTLEN = &LENGTH OF &RESPONSE
&IF &INPUTLEN GT 2 &GOTO -STRINGOK

```

-NOSTRING

```

&PRINT PLEASE ENTER THE STRING. IT MAY CONTAIN BLANKS.

```

\* ----- CLEAR ANY EXTRANEIOUS INPUT FROM STACK

```

EXEC SRCHCLRS
&READ STRING &STRING
&IF /&STRING = / &GOTO -NOSTRING
&GOTO -STRINGFOUND

```

-STRINGOK

```

&STRING = &SUBSTR OF &RESPONSE 3

```

-STRINGFOUND

```

&QUOTEDSTRING = &CONCAT OF ' &STRING '
&COMMAND = &SUBSTR OF &RESPONSE 1 1
&IF &COMMAND = D &GOTO -DELETE
&IF &COMMAND = C &GOTO -DELETELEADING
&IF &COMMAND = S &GOTO -STRING
&IF &COMMAND = V &GOTO -VARIABLE

```

-BADINPUT

```

CLRSCRN

```

```

&PRINT PLEASE USE ONE OF THE COMMANDS IN THE LIST
&GOTO -MAINMENU
-PRINT
* ----- PRINT USING SEARCH STRING FOR FILETYPE, IF POSSIBLE
  SET CMSTYPE HT
* ----- DETERMINE LRECL OF OUTPUT FILE; IF TOO LONG, TRUNCATE
  LISTF SEARCH RESULTS A (EXEC FORMAT
  EXECIO 1 DISKR CMS EXEC A
  &READ ARGS
  &IF &7 GT 132 &GOTO -TRUNC
  REN SEARCH RESULTS A SEARCH: &FIRSTWORD A
  &IF &RC GT 0 &GOTO -INVAL
  PRINT SEARCH: &FIRSTWORD A
  REN SEARCH: &FIRSTWORD A SEARCH RESULTS A
  &GOTO -DOCUPRT
-INVAL PRINT SEARCH RESULTS A
  &GOTO -DOCUPRT
-TRUNC COPY SEARCH RESULTS A SEARCH- RESULTS A (LRECL 132 RECFM F REP
  REN SEARCH- RESULTS A SEARCH-: &FIRSTWORD A
  &IF &RC GT 0 &GOTO -INVTR
  PRINT SEARCH-: &FIRSTWORD A
  ERASE SEARCH-: &FIRSTWORD A
  &GOTO -DOCUPRT
-INVTR PRINT SEARCH- RESULTS A
  ERASE SEARCH- RESULTS A
* ----- PUT STRING IN OUTPUT FILE, BEFORE DASHED LINE.
-DOCUPRT &STACK V OFF
  &STACK FIND -----
  &STACK UP
  &STACK INPUT &B4 PRINT
  &STACK FILE
  XEDIT SEARCH RESULTS A (NOPROF NOSC
  CLRSCRN
  SET CMSTYPE RT
  &GOTO -MAINMENU
-MAKESFILES
* ----- USE FILE IDS IN SEARCH RESULTS TO CREATE SFILES LIST
  EXEC SRCHSFIL
  &GOTO -MAINMENU
-DELETE
* ----- 'LOCATION' A DUMMY; DELETE LINES CONTAINING STRING
  &STACK LOCATION GT 0
  &STACK &B4 DELETE STRING: &B5 &BLANK &B4 &QUOTEDSTRING
  EXEC SRCHSTRG &STRING
  &GOTO -LOOK
-DELETELEADING
* ----- 'LOCATION' A DUMMY; DELETE LINES BEGINNING WITH STRING
  &STACK LOCATION = 1
  &STACK &B4 DELETE LEADING STRING: &B4 &QUOTEDSTRING
  EXEC SRCHSTRG &STRING
  &GOTO -LOOK
-STRING
* ----- 'LOCATION' A DUMMY; DELETE LINES NOT CONTAINING STRING
  &STACK LOCATION = 0
  &STACK &B4 SEARCH STRING: &B5 &BLANK &B4 &QUOTEDSTRING
  EXEC SRCHSTRG &STRING

```



```

&GOTO -LOOK
-VARIABLE
  &STACK &B4 SEARCH VARIABLE: &B5 &B4 &QUOTEDSTRING
  EXEC SRCHVAR &STRING
  &GOTO -LOOK
-HELP
  EXEC SRCHHELP

```

SRCHCHNG EXEC

```

&TRACE ERR
* ----- SRCHCHNG EXEC IS PART OF SEARCH ROUTINE.
* ----- SEE S EXEC, SRCHHELP EXEC, AND WRITTEN REPORT.
* ----- AUTHOR: RALPH SEELEY (202)447-8815.
* ----- VERSION: 3-12-84. LAST MODIFICATION: 3-12-84 RMS.
* ----- PROGRAM EDITS FILES FROM ALPHABETIZED LIST; IN EACH
* ----- IT CHANGES FIRST OCCURRENCE OF STRING TO ITSELF IN
* ----- EVERY LINE WHERE STRING FOUND, DISPLAYING
* ----- "CHANGED" LINES AT TERMINAL AND THUS IN CONSOLE.
* ----- CONSOLE IS RETRIEVED AND EDITED IN OUTPUT FILE.
&B4 = &CONCAT OF &BLANK &BLANK &BLANK &BLANK
&B5 = &CONCAT OF &B4 &BLANK
&QUOTEDSTRING = &CONCAT OF ' &ARGSTRING '
* ----- &S USED IN CHANGE COMMAND; CHANGES STRING TO ITSELF
&S = &CONCAT OF @ &ARGSTRING @ &ARGSTRING @
&READ VAR &FNSET &FTSET &FMSET
CP SET MSG OFF
CP SET MSG OFF
* ----- CLEAR EXTRANEIOUS INPUT FROM STACK
EXEC SRCHCLRS
* ----- END PREVIOUS CONSOLE RECORDING, BEGIN NEW ONE
CP SP CON CLOSE
* ----- ENSURE THAT NEW CONSOLE WILL BE SENT TO USER'S READER
CP SP CON * TERM START
EXECIO 1 DISKR SEARCH SORTFILES A
&LOOP -CHANGE WHILE &RC = 0
  CLRSCRN
  SET CMSTYPE RT
  &READ VAR &FN &FT &FM
  &PRINT _____ &BLANK &FN &FT &FM
* ----- CHANGE STRING TO ITSELF AND DISPLAY FOR CONSOLE
  &STACK LINE RIGHT
  &STACK V ON
  &STACK C &S *
  &STACK QUIT
  EDIT &FN &FT &FM (NODISP)
  SET CMSTYPE HT
-CHANGE EXECIO 1 DISKR SEARCH SORTFILES A
* ----- PUT CURRENT READER FILES IN HOLD STATUS
CP CH R ALL HOLD
* ----- END CONSOLE RECORDING AND SEND TO USER'S READER
CP SP CON CLOSE
CP SET MSG ON
SET CMSTYPE HT
ERASE SEARCH RESULTS A

```

```

* ----- GET NEXT READER FILE NOT IN HOLD, WHICH IS CONSOLE
  READ SEARCH RESULTS A
* ----- RETURN REMAINING READER FILES TO NOHOLD STATUS
  CP CH R ALL NOHOLD
  CP SET IMSG ON
    &STACK SET V OFF
* ----- WARN WHEN FILE NOT FOUND ("NEW FILE");
* ----- REMOVE OTHER UNWANTED LINES FROM OUTPUT FILE.
  &STACK SRCHNFNF
  &STACK INPUT &B4 SEARCH FILE SET: &B5 &B4 &FNSET &FTSET &FMSET
  &STACK INPUT &B4 SEARCH STRING: &B5 &BLANK &B4 &QUOTEDSTRING
  &STACK INPUT &B4 -----
  &STACK FILE
  XEDIT SEARCH RESULTS A (NOSC NOPROF

```

### SRCHCLRS EXEC

```

  &TRACE ERR
* ----- SRCHCLRS EXEC IS PART OF SEARCH ROUTINE.
* ----- SEE S EXEC, SRCHHELP EXEC, AND WRITTEN REPORT.
* ----- AUTHOR: RALPH SEELEY (202)447-8815.
* ----- VERSION: 3-12-84. LAST MODIFICATION: 3-12-84 RMS.
* ----- PROGRAM CLEARS STACK OF ANY EXTRANEIOUS INPUT.
  SET CMSTYPE HT
* ----- FIND NUMBER OF LINES IN STACK;
* ----- THEN READ AND IGNORE EACH LINE IN STACK.
  SENTRIES
  &LOOP 1 &RC
    &READ VAR
  SET CMSTYPE RT

```

### SRCHHELP EXEC

```

  &TRACE ERR
* ----- SRCHHELP EXEC IS PART OF SEARCH ROUTINE.
* ----- SEE S EXEC AND WRITTEN REPORT.
* ----- AUTHOR: RALPH SEELEY (202)447-8815.
* ----- VERSION: 3-12-84. LAST MODIFICATION: 3-22-84 RMS.
* ----- PROGRAM DISPLAYS HELP DOCUMENT AT TERMINAL.
  &BEGPRINT *
  SYNTAX: S (SEARCH STRING)
          (FILES THROUGH WHICH TO SEARCH)
          (SEARCH TECHNIQUE, FROM MENU)
          (OPTIONAL FURTHER PROCESSING COMMAND, FROM MENU, AND STRING)
          ETC.

  EXAMPLE: S WRITE(6,
           * FORTRAN B
           S
           C C
           P
           X

```

- NOTE 1: AFTER EACH LINE ABOVE IS ENTERED, THE PROGRAM PROMPTS FOR THE NEXT INPUT. A FEW MINUTES OF PROCESSING COULD FOLLOW ENTERING OF THE SEARCH TECHNIQUE IF THERE ARE TENS OF THOUSANDS OF LINES THROUGH WHICH TO SEARCH.
- NOTE 2: THE PROGRAM REQUIRES READ/WRITE (R/W) ACCESS TO AN "A" DISK. THIS DISK NEED NOT BE THE 191 DISK. A TEMPORARY DISK MAY BE CREATED AND THEN ACCESSED AS THE "A" DISK.
- NOTE 3: THE PROGRAM IS NOT INTENDED TO WORK WITH STRINGS IN WHICH THE @ (AT) SYMBOL IS PRESENT.
- NOTE 4: THE PROGRAM TRUNCATES PRINTOUT TO A RECORD LENGTH OF 132 CHARACTERS. THE INITIAL SEARCH TECHNIQUE O WILL NOT WORK CORRECTLY ON FILES WITH RECORD LENGTHS OVER 132. THE INITIAL SEARCH TECHNIQUES S AND V WILL NOT WORK CORRECTLY ON FILES WITH RECORD LENGTHS OVER 249 CHARACTERS.
- NOTE 5: IF THE INITIAL SEARCH TECHNIQUE O IS USED AND UNNUMBERED LINES ARE FOUND, FURTHER SEARCHING IN THE OUTPUT FILE WILL BE INACCURATE.
- NOTE 6: IF THE INITIAL SEARCH TECHNIQUE O IS USED, THE USER'S READER FILES WILL BE IN NOHOLD STATUS AFTER THE SEARCH. THIS IS NOT NORMALLY A PROBLEM.

SEARCH STRING: THE SEARCH STRING MAY INCLUDE LEADING, TRAILING, AND EMBEDDED BLANKS. ANY BLANK AFTER THE SINGLE BLANK SEPARATING THE S COMMAND FROM THE SEARCH STRING IS PART OF THE STRING. TO REMOVE AN UNWANTED BLANK, USE THE DEL KEY OR THE ERASE EOF KEY.

FILES TO SEARCH: THERE ARE FIVE OPTIONS HERE. THE IDENTIFIER (THAT IS, THE FILENAME, FILETYPE, AND FILEMODE) OF A PARTICULAR FILE MAY BE ENTERED. EXAMPLE: SUPPLY FORTRAN F

AN ASTERISK (\*) OR PERCENT SIGN (%) MAY BE USED AS PART OF THE FILE IDENTIFIER. THE \* DENOTES FROM ZERO TO EIGHT VALID CHARACTERS. THE PERCENT SIGN DENOTES EXACTLY ONE VALID CHARACTER. VALID CHARACTERS WHICH ARE PERMITTED IN A SPECIFIC FILE IDENTIFIER ARE LETTERS, NUMBERS, AND \$ - + : \_  
 EXAMPLE: \*DAT%00 FORTRAN \*

THE DEFAULTS MAY BE USED IN WHOLE OR PART. NULL INPUT (THAT IS, NOT TYPING ANY FILE IDENTIFIER BEFORE PRESSING ENTER) GIVES ALL OF THE DEFAULTS.

DEFAULT FILENAME: \*  
 DEFAULT FILETYPE: FORTRAN  
 DEFAULT FILEMODE: MODE OF THE S PROGRAM, USUALLY A  
 TWO EXAMPLES: EXCHNG FORTRAN  
 DEMAND

TO RESUME WORKING WITH THE LAST SEARCH OUTPUT, START AS USUAL WITH THE COMMAND S FOLLOWED BY THE ORIGINAL SEARCH STRING. FOR THE FILE ID, USE SEARCH RESULTS

TO USE YOUR OWN LIST OF FILES THROUGH WHICH TO SEARCH, FIRST CREATE THE LIST IN A DATASET WITH FILETYPE SFILES. IN IT, THE FILE IDS SHOULD BE ENTERED ONE PER LINE WITH NO BLANK LINES. IT IS BETTER NOT TO HAVE FILEMODES IN THIS LIST. IF THE FILETYPE IS NOT PRESENT, THE DEFAULT IS FORTRAN. THE PROGRAM LEFT-JUSTIFIES AND ALPHABETIZES THE LIST DURING USE. WHEN THE PROGRAM ASKS FOR THE FILE IDENTIFIER OVER WHICH TO SEARCH, ENTER THE IDENTIFIER OF THE DATASET CONTAINING THE LIST. THE MODE OF THE SFILES LIST MAY BE ENTERED. A SECOND MODE TO INDICATE WHERE THE FILES TO BE SEARCHED RESIDE MAY BE ENTERED. THE DEFAULT FOR THESE MODES IS THE MODE OF THE S PROGRAM. TWO EXAMPLES: WORK SFILES A J  
EXCHANGE SFILES

SEARCH TECHNIQUE: THERE ARE THREE SEARCH OPTIONS. THEY GIVE TRADEOFFS WITH RESPECT TO SPEED OF SEARCHING, VOLUME OF OUTPUT GENERATED, AND OPPORTUNITY FOR FURTHER SEARCHING. THE SPEED OF ALL THREE TECHNIQUES DEPENDS IN PART ON THE NUMBER OF FILES SEARCHED, THE TOTAL NUMBER OF LINES SEARCHED, AND THE TOTAL NUMBER OF LINES FOUND.

THE FIRST TECHNIQUE IS DENOTED S. IT SEARCHES FOR A STRING. LINE NUMBERS FOR FOUND LINES ARE DISPLAYED FOR ANY FILETYPE. ITS SPEED IS RATHER PREDICTABLE BASED ON THE CRITERIA ABOVE.

THE NEXT TECHNIQUE IS DENOTED V. IT SEARCHES FOR A 'VARIABLE;' THAT IS, A STRING THAT IS NOT IMMEDIATELY PRECEDED OR FOLLOWED BY A LETTER OR A NUMBER. THIS DEFINITION INCLUDES ALL OCCURRENCES OF A FORTRAN VARIABLE THAT WOULD BE RECOGNIZED BY A COMPILER, AS WELL AS SOME SPURIOUS OCCURRENCES IN COMMENT CARDS AND IN SOME LITERALS (TWO EXAMPLES: 'XMAX' AND 6H/XMAX/ ), USUALLY IN FORMAT STATEMENTS. LINE NUMBERS ARE DISPLAYED FOR ALL FILETYPES. THE SPEED OF THE SEARCH DEPENDS IN PART ON THE NUMBER OF PLACES WHERE THE STRING IS FOUND, BUT IS DELIMITED BY LETTERS OR NUMBERS. THE VOLUME OF OUTPUT GENERATED BY THIS TECHNIQUE IS NEVER MORE, AND FREQUENTLY MUCH LESS, THAN THAT GENERATED BY SEARCHING FOR THE EQUIVALENT STRING, PARTICULARLY WHERE THE STRING IS SHORT. THEREFORE, THERE IS LESS CHANCE WITH THIS TECHNIQUE OF CREATING AN OUTPUT FILE TOO LARGE TO BE EDITED.

THE THIRD TECHNIQUE IS DENOTED O. IT SEARCHES FOR A STRING. THE FOUND LINES ARE DISPLAYED ON THE SCREEN. IF MORE THAN A SCREENLOAD IS FOUND PER FILE, THE SCREEN MUST BE CLEARED FOR THE PROGRAM TO PROCEED. THE CLEARING MAY BE ACCOMPLISHED BY PRESSING THE CLEAR BUTTON. OTHERWISE, THE SCREEN WILL BE AUTOMATICALLY CLEARED AFTER ONE MINUTE. THUS, THE SPEED OF THE TECHNIQUE IS HIGHLY DEPENDENT ON THE NUMBER OF TIMES THAT MORE THAN 20 OCCURRENCES OF THE SEARCH STRING WILL BE FOUND PER FILE, GIVEN THAT THE USER IS UNWILLING TO DIVERT ATTENTION FROM OTHER WORK WHILE THE SEARCH IS TAKING PLACE JUST TO PRESS THE CLEAR BUTTON. IF NO FULL SCREENLOADS ARE FOUND, THE O TECHNIQUE IS ALMOST TWICE AS FAST AS THE S TECHNIQUE. IN GENERAL, THE O TECHNIQUE ONLY GIVES LINE NUMBERS FOR THE FOLLOWING FILETYPES: ASM3705, ASSEMBLE, AMSERV, COBOL, COPY, DIRECT, FORTRAN, MACLIB, MACRO, PL1, PL1OPT, UPDATE, AND

UPDTXXXX.

THIS IS BECAUSE OF SERIALIZATION, TRUNCATION, RECORD LENGTH, AND RECORD FORMAT DEFAULTS FOR VARIOUS FILETYPES. A FILE WILL NOT HAVE LINE NUMBERS UNLESS THE FOLLOWING CONDITIONS HELD WHEN IT WAS LAST SAVED: SERIALIZATION WAS ON, RECORD LENGTH WAS AT LEAST 8 GREATER THAN TRUNCATION LENGTH, AND RECORD FORMAT WAS FIXED (F). FOR ALL OTHER FILETYPES, NOT ONLY IS THE INITIAL OUTPUT LESS USEFUL, BUT FURTHER PROCESSING OF THE OUTPUT FILE IS PREVENTED BY THE LACK OF LINE NUMBERS. IF A FILE THAT IS NORMALLY NUMBERED BECAUSE OF ITS FILETYPE IS LAST EDITED WITH SPF, INSERTED LINES MAY BE LOST IN ANY FURTHER PROCESSING AFTER THE INITIAL SEARCH WITH THE O TECHNIQUE.

THE USER MAY EXIT THE PROGRAM FROM THE SEARCH TECHNIQUE MENU BY ENTERING X. DURING SEARCHING, THE USER MAY STOP THE PROGRAM BY PRESSING THE PA1 KEY REPEATEDLY UNTIL THE MESSAGE VM READ APPEARS AT THE LOWER RIGHT CORNER OF THE TERMINAL SCREEN. NEXT, THE COMMAND HX SHOULD BE ENTERED. THE MESSAGE CMS WILL THEN APPEAR, AFTER WHICH ANY CMS COMMAND MAY BE ENTERED.

THE INITIAL SEARCH STRING AND THE SEARCH FILE SET ARE DISPLAYED BELOW THE SEARCH TECHNIQUE MENU. IF THEY ARE NOT CORRECT, ENTER X AND START AGAIN.

EDITOR: WHEN THE INITIAL SEARCH IS COMPLETED, THE PROGRAM ENTERS THE EDITOR, WHERE THE OUTPUT MAY BE VIEWED OR MODIFIED. WHEN YOU ARE DONE LOOKING AT THE FILE, EXIT FROM THE EDITOR BY PRESSING PF 3. IN THE EDITOR, XEDIT LINE NUMBERS ARE TURNED OFF, IN ORDER THAT ENTIRE FORTRAN LINES MAY BE VIEWED AT ONCE. YOU MAY DISPLAY THE EDITOR LINE NUMBERS BY PRESSING PF KEY 5 AND REMOVE THE NUMBERS BY PRESSING PF KEY 6. PF KEY DEFINITIONS FOLLOW:

- PF 1: RETURN TO SINGLE SCREEN AFTER USING PF 2.
- PF 2: SPLIT SCREEN. TO VIEW ANOTHER FILE, TYPE 'XEDIT' AND THE OTHER FILE IDENTIFIER, ALL ON THE COMMAND INPUT LINE.
- PF 3: FILE. EXIT FROM THE EDITOR, SAVE THE CURRENT VERSION OF THE OUTPUT FILE, AND ENTER THE NEXT MENU.
- PF 4: REPEAT THE LAST XEDIT COMMAND.
- PF 5: DISPLAY XEDIT LINE NUMBERS FOR THE OUTPUT FILE.
- PF 6: END DISPLAY OF XEDIT LINE NUMBERS FOR THE OUTPUT FILE.
- PF 7: SCROLL THE SCREEN UP, TOWARD THE BEGINNING OF THE FILE.
- PF 8: SCROLL THE SCREEN DOWN, TOWARD THE END OF THE FILE.
- PF 9: SCROLL THE SCREEN TO THE BEGINNING OF THE FILE.
- PF 10: SCROLL THE SCREEN TO THE LEFT (AFTER USING PF 11).
- PF 11: SCROLL THE SCREEN TO THE RIGHT.
- PF 12: SCROLL THE SCREEN TO THE END OF THE FILE.

FURTHER PROCESSING MENU: THERE ARE SEVERAL OPTIONS AVAILABLE IN THIS MENU ENABLING THE USER TO TRIM THE OUTPUT FILE TO JUST THOSE LINES OF INTEREST. ANY SEARCHING IS DOCUMENTED IN THE OUTPUT FILE AND ALSO DISPLAYED ON THE SCREEN BELOW THE MENU.

THE FIRST COMMAND IS X. IF X IS ENTERED, THE PROGRAM STOPS IMMEDIATELY.

THE SECOND COMMAND IS P. IF P IS ENTERED, THE OUTPUT FILE AS IT WAS LAST SEEN IS PRINTED. THE FILETYPE OF THE PRINTED OUTPUT REFLECTS THE INITIAL SEARCH STRING, AS AN AID FOR LATER LEAFING THROUGH PRINTED OUTPUT.

NEXT IS THE COMMAND L. L SIMPLY RETURNS THE PROGRAM TO THE EDITOR WHERE THE OUTPUT MAY BE VIEWED OR MODIFIED.

THE COMMAND F CREATES AN SFILES LIST BY COPYING THE FILE IDENTIFIERS (LESS FILEMODES) FROM THE CURRENT OUTPUT SEARCH RESULTS INTO AN SFILES LIST. THE PROGRAM PROMPTS THE USER FOR INPUT OF A FILENAME AND FILEMODE FOR THE SFILES LIST. THE USER MAY ENTER A COMMENT LINE THAT THE PROGRAM WILL PLACE IN THE FIRST LINE OF THE LIST. THE COMMENT LINE COULD EXPLAIN THE ORIGIN OR INTENDED USE OF THE LIST, AND THE DATE OF CREATION.

THE REMAINING COMMANDS SEARCH FOR A STRING AND KEEP OR DELETE LINES CONTAINING THAT STRING. THE STRING MAY BE ENTERED AFTER THE COMMAND AND A SINGLE BLANK WHICH SEPARATES THE COMMAND FROM THE STRING. ANY OTHER BLANKS ENTERED ARE PART OF THE STRING. BLANKS ENTERED BY MISTAKE MAY BE REMOVED WITH THE DEL KEY OR THE ERASE EOF KEY. IF THE STRING IS NOT ENTERED ON THE SAME LINE AS THE COMMAND, THE PROGRAM WILL PROMPT FOR THE STRING. THE PROGRAM DISTINGUISHES BETWEEN FILE IDENTIFIERS AND LINES FOUND WITHIN FILES AS A RESULT OF SEARCHING. SEARCHING IS PERFORMED ONLY ON THE FOUND LINES, NOT ON THE LINE NUMBERS TO THE LEFT OF EACH FOUND LINE AND NOT ON THE FILE IDS. IF ALL OF THE FOUND LINES ASSOCIATED WITH A GIVEN FILE IDENTIFIER ARE REMOVED, THE PROGRAM WILL REMOVE THAT FILE IDENTIFIER.

THE D COMMAND DELETES ALL LINES WHICH CONTAIN THE SPECIFIED STRING. IF THE STRING IS JUST A BLANK OR IS ONE OF THE STRINGS THAT WERE PREVIOUSLY SEARCHED FOR, THE RESULT MAY BE AN OUTPUT FILE EMPTY EXCEPT FOR DOCUMENTATION.

THE C COMMAND DELETES ALL LINES WHICH BEGIN IN COLUMN 1 WITH THE SPECIFIED STRING. THIS COMMAND MAY BE USED TO DELETE COMMENT STATEMENTS. EXAMPLE FOR FORTRAN: C C

THE S COMMAND RETAINS ONLY LINES WHICH CONTAIN THE SPECIFIED STRING.

THE V COMMAND RETAINS ONLY LINES WHICH CONTAIN THE SPECIFIED STRING AS A VARIABLE. THIS MEANS AN OCCURRENCE OF THE STRING IN WHICH IT IS NEITHER IMMEDIATELY PRECEDED NOR FOLLOWED BY A LETTER OR A NUMBER.

FOR MORE INFORMATION, CONTACT RALPH SEELEY (202)447-8815  
500 12TH ST. SW WASHINGTON, D.C. 20250

SRCHLOCS EXEC

```
&TRACE ERR
* ----- SRCHLOCS EXEC IS PART OF SEARCH ROUTINE.
* ----- SEE S EXEC, SRCHHELP EXEC, AND WRITTEN REPORT.
* ----- AUTHOR: RALPH SEELEY (202)447-8815.
* ----- VERSION: 3-12-84. LAST MODIFICATION: 3-12-84 RMS.
* ----- PROGRAM RECORDS IN OUTPUT FILE ANY LINES IN FILES FROM
* ----- ALPHABETIZED LIST WHICH CONTAIN SEARCH STRING.
ERASE SEARCH RESULTS A
&B4 = &CONCAT OF &BLANK &BLANK &BLANK &BLANK
&B5 = &CONCAT OF &BLANK &B4
&LOCSTRING = &CONCAT OF @ &ARGSTRING @
&DOCSTRING = &CONCAT OF ' &ARGSTRING '
* ----- FOR FOLLOWING FILETYPES, SEARCH AND KEEP ONLY SPECIFIED
* ----- RANGE OF COLUMNS; IGNORE SERIAL IDENTIFICATION COLUMNS.
* ----- THE ASSIGNMENTS PREPARE FOR EXECIO DISKR ... &XSER&FT.
* ----- THE FILETYPE DESIGNATIONS BELOW ARE PRECEDED BY 'XSER'
* ----- BECAUSE &XSER&FT CREATES A UNIQUE VARIABLE NAME, AS
* ----- OPPOSED TO &&FT, WHICH COULD OVERWRITE ANOTHER VALUE.
&XSERCOPY = &STRING OF ZONE 1 71 MARGINS 1 71
&XSERASM3705 = &XSERCOPY
&XSERASSEMBLE = &XSERCOPY
&XSERMACLIB = &XSERCOPY
&XSERMACRO = &XSERCOPY
&XSERUPDATE = &XSERCOPY
&XSERFORTRAN = &STRING OF ZONE 1 72 MARGINS 1 72
&XSERAMSERV = &XSERFORTRAN
&XSERCOBOL = &XSERFORTRAN
&XSERDIRECT = &XSERFORTRAN
&XSERPLI = &XSERFORTRAN
&XSERPLIOPT = &XSERFORTRAN
&READ VAR &FNSET &FTSET &FMSET
&STACK &B4 SEARCH FILE SET: &B4 &B5 &FNSET &FTSET &FMSET
&STACK &B4 SEARCH STRING: &B4 &BLANK &B5 &DOCSTRING
&STACK &B4 -----
* ----- CREATE OUTPUT FILE WITH RECFM V (LRECL UP TO 255)
EXECIO 3 DISKW SEARCH RESULTS A 1 V
* ----- GET A FILE ID FROM ALPHABETIZED LIST
EXECIO 1 DISKR SEARCH SORTFILES A
&LOOP -GETFN WHILE &RC = 0
&READ VAR &FN &FT &FM
* ----- IF FILETYPE IS UPDTXXXX, IGNORE SERIAL ID COLUMNS
&UPDTTEST = &SUBSTR OF &FT 1 4
&IF &UPDTTEST = UPDT &XSER&FT = &XSERCOPY
CLRSCRN
SET CMSTYPE RT
&PRINT _____&BLANK &FN &FT &FM
SET CMSTYPE HT
* ----- FIND NEXT LINE IN FILE WHICH CONTAINS &LOCSTRING.
* ----- IGNORE SERIAL IDENTIFICATION COLUMNS, USING &XSER&FT
EXECIO * DISKR &FN &FT &FM (LOCATE &LOCSTRING &XSER&FT
&READCODE = &RC
* ----- IF &LOCSTRING NOT FOUND IN FILE, GET NEXT FILE ID
&IF &READCODE = 3 &GOTO -GETFN
* ----- IF FILE NOT FOUND, PUT WARNING IN OUTPUT FILE
```

```

&IF &READCODE NE 24 &GOTO -READOK
EXECIO 1 DISKW SEARCH RESULTS A (ST ____&BLANK &FN &FT &FM
&X = =====
EXECIO 1 DISKW SEARCH RESULTS A (ST &X FILE NOT FOUND &X
&GOTO -GETFN
-READOK EXECIO 1 DISKW SEARCH RESULTS A (ST ____&BLANK &FN &FT &FM
&LOOP -ENDLOC WHILE &RC = 0
* ----- GET NUMBER OF LINE, RELATIVE TO BEGINNING OF FILE
&READ VAR * &LINENUM
* ----- RIGHT-JUSTIFY LINE NUMBER IN 5-COLUMN FIELD
&NUMRIGHT = &RIGHT OF &LINENUM 5
* ----- READ FOUND LINE
&READ STRING &LOCLINE
EXECIO 1 DISKW SEARCH RESULTS A (ST &NUMRIGHT &LOCLINE
-ENDLOC EXECIO * DISKR &FN &FT &FM (LOCATE &LOCSTRING &XSER&FT
-GETFN EXECIO 1 DISKR SEARCH SORTFILES A

```

### SRCHLOCV EXEC

```

&TRACE ERR
* ----- SRCHLOCV EXEC IS PART OF SEARCH ROUTINE.
* ----- SEE S EXEC, SRCHHELP EXEC, AND WRITTEN REPORT.
* ----- AUTHOR: RALPH SEELEY (202)447-8815.
* ----- VERSION: 3-12-84. LAST MODIFICATION: 3-12-84 RMS.
* ----- PROGRAM RECORDS IN OUTPUT FILE ANY LINES IN FILES FROM
* ----- ALPHABETIZED LIST WHICH CONTAIN SEARCH STRING, NEITHER
* ----- PRECEDED NOR FOLLOWED BY LETTERS OR NUMBERS.
ERASE SEARCH RESULTS A
&B4 = &CONCAT OF &BLANK &BLANK &BLANK &BLANK
&B5 = &CONCAT OF &BLANK &B4
&DOCSTRING = &CONCAT OF ' &ARGSTRING '
&LOCSTRING = &CONCAT OF @ &ARGSTRING @
&LENVAR = &LENGTH OF &ARGSTRING
&X = =====
* ----- FOR FOLLOWING FILETYPES, SEARCH AND KEEP ONLY SPECIFIED
* ----- RANGE OF COLUMNS; IGNORE SERIAL IDENTIFICATION COLUMNS.
* ----- THE ASSIGNMENTS PREPARE FOR EXECIO DISKR ... &XSER&FT.
* ----- THE FILETYPE DESIGNATIONS BELOW ARE PRECEDED BY 'XSER'
* ----- BECAUSE &XSER&FT CREATES A UNIQUE VARIABLE NAME, AS
* ----- OPPOSED TO &&FT, WHICH COULD OVERWRITE ANOTHER VALUE.
&XSERCOPY = &STRING OF ZONE 1 71 MARGINS 1 71
&XSERASM3705 = &XSERCOPY
&XSERASSEMBLE = &XSERCOPY
&XSERMACLIB = &XSERCOPY
&XSERMACRO = &XSERCOPY
&XSERUPDATE = &XSERCOPY
&XSERFORTRAN = &STRING OF ZONE 1 72 MARGINS 1 72
&XSERAMSERV = &XSERFORTRAN
&XSERCOBOL = &XSERFORTRAN
&XSERDIRECT = &XSERFORTRAN
&XSERPLI = &XSERFORTRAN
&XSERPLIOPT = &XSERFORTRAN
&FIRSTCHAR = &SUBSTR OF &ARGSTRING 1 1
* ----- A CHARACTER GREATER THAN OR EQUAL TO 'A' IS A LETTER
* ----- OR NUMBER. A CHARACTER LESS THAN 'A' IS NEITHER A

```



```

* ----- LETTER NOR A NUMBER.
&IF &FIRSTCHAR GE A &IF &FIRSTCHAR NE 9 &FILLCHAR = 9
&IF &FIRSTCHAR GE A &IF &FIRSTCHAR = 9 &FILLCHAR = 8
&IF &FIRSTCHAR LT A &IF &FIRSTCHAR NE @ &FILLCHAR = @
&IF &FIRSTCHAR LT A &IF &FIRSTCHAR = @ &FILLCHAR = ¢
&STRGTRUNC = &SUBSTR OF &ARGSTRING 2 *
&FILLSTRG = &CONCAT OF &FILLCHAR &STRGTRUNC
&READ VAR &FNSET &FTSET &FMSET
&STACK &B4 SEARCH FILE SET: &B4 &B5 &FNSET &FTSET &FMSET
&STACK &B4 SEARCH VARIABLE: &B4 &B5 &DOCSTRING
&STACK &B4 -----
* ----- CREATE OUTPUT FILE WITH RECFM V (LRECL UP TO 255)
EXECIO 3 DISKW SEARCH RESULTS A 1 V
* ----- GET FILE ID FROM ALPHABETIZED LIST
EXECIO 1 DISKR SEARCH SORTFILES A
&LOOP -GETFN WHILE &RC = 0
    &READ VAR &FN &FT &FM
* ----- IF FILETYPE IS UPDTXXXX, IGNORE SERIAL ID COLUMNS
    &UPDTTEST = &SUBSTR OF &FT 1 4
    &IF &UPDTTEST = UPDT &XSER&FT = &XSERCOPY
    CLRSCRN
    SET CMSTYPE RT
    &PRINT _____&BLANK &FN &FT &FM
    SET CMSTYPE HT
* ----- FILE ID HAS NOT YET BEEN WRITTEN IN OUTPUT FILE
    &FIDWRIT = 0
* ----- FIND NEXT LINE IN FILE WHICH CONTAINS &LOCSTRING.
* ----- IGNORE SERIAL IDENTIFICATION COLUMNS, USING &XSER&FT
EXECIO * DISKR &FN &FT &FM (LOCATE &LOCSTRING &XSER&FT
&READCODE = &RC
* ----- IF STRING IS NOT FOUND IN FILE, GET NEXT FILE ID
&IF &READCODE = 3 &GOTO -GETFN
* ----- IF FILE IS NOT FOUND, PUT WARNING IN OUTPUT FILE
&IF &READCODE NE 24 &GOTO -READ
EXECIO 1 DISKW SEARCH RESULTS A (ST _____&BLANK &FN &FT &FM
EXECIO 1 DISKW SEARCH RESULTS A (ST &X FILE NOT FOUND &X
&GOTO -GETFN
-READ &LOOP -ENDLOC WHILE &RC = 0
* ----- GET NUMBER OF LINE, RELATIVE TO BEGINNING OF FILE
    &READ VAR * &LINENUM
* ----- READ FOUND LINE
    &READ STRING &LOCLINE
    &MODLINE = &LOCLINE
    &LOOP -FIND *
        &LOCVAR = &LOCATION OF &ARGSTRING &MODLINE
* ----- A LOCATION OF ZERO MEANS STRING NOT FOUND
        &IF &LOCVAR = 0 &GOTO -ENDLOC
        &LOCPLUSLEN = &LOCVAR + &LENVAR
* ----- THE BEGINNING OF THE LINE MAY BE A DELIMITER
        &IF &LOCVAR = 1 &GOTO -SUFFIX
        &LOCLESS1 = &LOCVAR - 1
        &PREFIX = &SUBSTR OF &MODLINE &LOCLESS1 1
        &IF &PREFIX GE A &GOTO -FILL
- SUFFIX &LENMOD = &LENGTH OF &MODLINE
* ----- THE END OF THE LINE MAY BE A DELIMITER
        &LOCPLUSLENLESS1 = &LOCPLUSLEN - 1

```

```

&IF &LOCPLUSLENLESS1 GE &LENMOD &GOTO -WRITE
&SUFFIX = &SUBSTR OF &MODLINE &LOCPLUSLEN 1
&IF &SUFFIX LT A &GOTO -WRITE
-FILL      &AFTERSTRG = &SUBSTR OF &MODLINE &LOCPLUSLEN
*          ----- CREATE NEW LINE STARTING WITH FILLED STRING
-FIND      &MODLINE = &CONCAT OF &FILLSTRG &AFTERSTRG
-WRITE     &NUMRIGHT = &RIGHT OF &LINENUM 5
*          ----- ONLY PLACE FILE ID IN OUTPUT FILE ONCE
&IF &FIDWRIT = 1 &GOTO -ONEFID
EXECIO 1 DISKW SEARCH RESULTS A (ST _____&BLANK &FN &FT &FM
&FIDWRIT = 1
-ONEFID    EXECIO 1 DISKW SEARCH RESULTS A (ST &NUMRIGHT &LOCLINE
-ENDLOC    EXECIO * DISKR &FN &FT &FM (LOCATE &LOCSTRING &XSER&FT
-GETFN     EXECIO 1 DISKR SEARCH SORTFILES A

```

SRCHNFNF XEDIT

```

&TRACE ERR
*----- SRCHNFNF XEDIT IS PART OF SEARCH ROUTINE.
*----- SEE S EXEC, SRCHHELP EXEC, AND WRITTEN REPORT.
*----- AUTHOR: RALPH SEELEY (202)447-8815.
*----- VERSION: 3-12-84. LAST MODIFICATION: 3-12-84 RMS.
*----- PROGRAM FINDS LINES BEGINNING "NEW FILE:" IN OUTPUT FILE,
*----- PLACES WARNING BELOW FILE ID, AND DELETES OTHER LINES
*----- ASSOCIATED WITH FILE. ALL LINES WHICH BEGIN "WRONG FILE
*----- FORMAT" ARE DELETED. ALL LINES OF OUTPUT ASSOCIATED WITH
*----- FILES WHICH DON'T CONTAIN THE SEARCH STRING ARE DELETED.
*----- THE PROGRAM THEN DELETES ALL LINES BEGINNING
*----- "EDIT:" OR "EOF:"
&X = =====
FIND NEW FILE:
&LOOP 7 UNTIL &RC NE 0
  UP
  INPUT &X FILE NOT FOUND &X
  DOWN
  DELETE
  DELETE
  DELETE
  FIND NEW FILE:
TOP
*----- DELETE ANY LINES BEGINNING "WRONG FILE FORMAT"
FIND WRONG FILE FORMAT
&LOOP 2 UNTIL &RC NE 0
  DELETE
  FIND WRONG FILE FORMAT
TOP
*----- DELETE ALL LINES FOR FILE IN WHICH STRING NOT FOUND
FIND NOT FOUND
&LOOP 6 UNTIL &RC NE 0
  UP 2
  DELETE
  DELETE
  DELETE
  DELETE
  FIND NOT FOUND

```

```

TOP
*----- DELETE ANY LINES BEGINNING "EDIT:"
FIND EDIT:
&LOOP 2 UNTIL &RC NE 0
  DELETE
  FIND EDIT:
TOP
*----- DELETE ANY LINES BEGINNING "EOF:"
FIND EOF:
&LOOP 2 UNTIL &RC NE 0
  DELETE
  FIND EOF:
TOP

```

SRCHPROF XEDIT

```

&TRACE ERR
*----- SRCHPROF XEDIT IS PART OF SEARCH ROUTINE.
*----- SEE S EXEC, SRCHHELP EXEC, AND WRITTEN REPORT.
*----- AUTHOR: RALPH SEELEY (202)447-8815.
*----- VERSION: 3-12-84. LAST MODIFICATION: 3-12-84 RMS.
*----- XEDIT PROFILE INITIALIZES SEARCH PROGRAM EDITOR ENVIRONMENT.
*----- THERE IS SOME RESEMBLANCE TO SPF SETTINGS EXCEPT PF5 & PF6.
COMMAND SET CMDLINE TOP
COMMAND SET CURLINE ON 5
COMMAND SET IMPCMSCP ON
COMMAND SET LRECL *
COMMAND SET MACRO ON
COMMAND SET MSGM ON L
COMMAND SET NULL ON
COMMAND SET NUM ON
COMMAND SET PREFIX OFF
*----- CREATE SHORT REMINDER OF PF KEY SETTINGS FOR XEDIT DISPLAY
&R = &STRING OF PF 1:1SCR 2:2SCR 3:FILE 4:REP 5:NUM 6:UNNUM
&L = &STRING OF 7:UP 8:DN 9:TOP 10:LE 11:RI 12:BOT
COMMAND SET RESERVED +3 HIGH &R &L
COMMAND SET SCALE OFF
COMMAND SET STAY ON
COMMAND SET VER ON 1 79
COMMAND SET WRAP OFF
COMMAND SET PF01 SET SCR 1
COMMAND SET PF02 SET SCR 2
COMMAND SET PF03 FILE
COMMAND SET PF04 =
*----- ASSIGN PF5 AND PF6 THREE COMMANDS EACH; FILL EDITOR SCREEN
SET LINEND OFF
COMMAND SET PF05 SET PREF ON # SET V 1 73 # SET PF11 RI 60
COMMAND SET PF06 SET PREF OFF # SET V 1 79 # SET PF11 RI 54
SET LINEND ON #
COMMAND SET PF07 BACKWARD
COMMAND SET PF08 FORWARD
COMMAND SET PF09 TOP
COMMAND SET PF10 LE 0
COMMAND SET PF11 RI 54
COMMAND SET PF12 B

```

SRCHSFIL EXEC

```
&TRACE ERR
* ----- SRCHSFIL EXEC IS PART OF SEARCH ROUTINE.
* ----- SEE S EXEC, SRCHHELP EXEC, AND WRITTEN REPORT.
* ----- AUTHOR: RALPH SEELEY (202)447-8815.
* ----- VERSION: 3-20-84. LAST MODIFICATION: 3-20-84 RMS.
* ----- PROGRAM CREATES SFILES LIST FROM CURRENT OUTPUT FILE.
* ----- THE PROGRAM GETS A FILENAME AND A FILEMODE FOR
* ----- THE SFILES LIST BEING CREATED. IT PROMPTS FOR A
* ----- COMMENT LINE TO DESCRIBE THE LIST.
* ----- IT THEN PUTS ALL FILE IDS (LESS FILEMODES)
* ----- IN THE SFILES FILE, ALONG WITH THE COMMENT LINE.
CLRSCRN
&FIDMARK = &CONCAT OF @ &BLANK &BLANK @
-GETFN &PRINT PLEASE ENTER THE FILENAME FOR THE SFILES LIST
* ----- CLEAR ANY EXTRANEIOUS TERMINAL INPUT FROM STACK
EXEC SRCHCLRS
&READ VAR &SFNAME
&IF /&SFNAME = / &GOTO -GETFN
&PRINT PLEASE ENTER THE FILEMODE FOR THE SFILES LIST
&PRINT THE DEFAULT FILEMODE IS A
* ----- CLEAR ANY EXTRANEIOUS TERMINAL INPUT FROM STACK
EXEC SRCHCLRS
&READ VAR &SFMODE
&IF /&SFMODE = / &SFMODE = A
SET CMSTYPE HT
* ----- CHECK TO BE SURE THAT THE SFILES IDENTIFIER IS NEW
STATE &SFNAME SFILES &SFMODE
&IF &RC GT 0 &GOTO -COMNT
SET CMSTYPE RT
&PRINT &SFNAME SFILES &SFMODE &BLANK ALREADY EXISTS
&GOTO -GETFN
-COMNT SET CMSTYPE RT
&PRINT YOU MAY ENTER A COMMENT LINE TO BE PLACED IN THE FILE
* ----- CLEAR ANY EXTRANEIOUS TERMINAL INPUT FROM STACK
EXEC SRCHCLRS
&READ STRING &COMMENT
SET CMSTYPE HT
* ----- GET THE NEXT FILE IDENTIFIER
EXECIO * DISKR SEARCH RESULTS A (FIND &FIDMARK
&IF &RC = 0 &GOTO -FIRST
SET CMSTYPE RT
&PRINT NO FILE IDS WERE FOUND
&EXIT
* ----- IGNORE LINE NUMBERS PLACED IN STACK WITH FILE ID
-FIRST &READ VAR
&READ ARGS
* ----- PUT FILENAME, FILETYPE, AND COMMENT IN SFILES LIST.
* ----- THE FILEMODE IS *, A DUMMY LOCATION HOLDER.
EXECIO 1 DISKW &SFNAME SFILES &SFMODE 1 V (ST &2 &3 * &COMMENT
EXECIO * DISKR SEARCH RESULTS A (FIND &FIDMARK
&LOOP -BOTTOM WHILE &RC = 0
* ----- IGNORE LINE NUMBERS PLACED IN STACK WITH FILE ID
&READ VAR
&READ ARGS
```

```

*           ----- PLACE THE FILENAME AND FILETYPE IN THE SFILES LIST
          EXECIO 1 DISKW &SFNAME SFILES &SFMODE (ST &2 &3
-BOTTOM   EXECIO * DISKR SEARCH RESULTS A (FIND &FIDMARK
*           ----- CLOSE THE SFILES FILE
          FINIS &SFNAME SFILES &SFMODE
          SET CMSTYPE RT
          CLRSCRN

```

SRCHSTRG EXEC

```

          &TRACE ERR
*           ----- SRCHSTRG EXEC IS PART OF SEARCH ROUTINE.
*           ----- SEE S EXEC, SRCHHELP EXEC, AND WRITTEN REPORT.
*           ----- AUTHOR: RALPH SEELEY (202)447-8815.
*           ----- VERSION: 3-12-84. LAST MODIFICATION: 3-12-84 RMS.
*           ----- PROGRAM SEARCHES OUTPUT FILE. DOCUMENTATION SECTION
*           ----- IS PUT IN TEMPORARY FILE. IGNORING LINE NUMBERS,
*           ----- LOCATION OF SEARCH STRING IN EACH FOUND LINE IS
*           ----- TESTED AGAINST LOCATION PARAMETER USING RELATIONAL
*           ----- OPERATOR. IF TEST RESULT FALSE, LINE IS PUT IN
*           ----- TEMPORARY FILE, AS ARE NON-CONSECUTIVE FILE IDS.
*           ----- COMPLETED TEMPORARY FILE IS NAMED OUTPUT FILE.
          SET CMSTYPE HT
*           ----- GET DUMMY, RELATIONAL OPERATOR, & LOCATION PARAMETER
          &READ VAR * &RELOPER &LOCPARM
*           ----- GET DOCUMENTATION OF LAST COMMAND AND SEARCH STRING
          &READ STRING &DOCUNEW
          EXECIO 1 DISKR SEARCH RESULTS A
          ERASE SRCHTEMP RESULTS A
*           ----- CREATE TEMPORARY FILE WITH RECFM V (LRECL UP TO 255)
          EXECIO 1 DISKW SRCHTEMP RESULTS A 1 V
          EXECIO 1 DISKR SEARCH RESULTS A
          &READ STRING &DOCUOLD
          &DASHLOC = 0
*           ----- PUT DOCUMENTATION SECTION IN OUTPUT FILE.
          &LOOP -DOCUEND UNTIL &DASHLOC = 6
              EXECIO 1 DISKW SRCHTEMP RESULTS A (STRING &DOCUOLD
              EXECIO 1 DISKR SEARCH RESULTS A
              &IF &RC NE 0 &EXIT
              &READ STRING &DOCUOLD
-DOCUEND &DASHLOC = &LOCATION OF ----- &DOCUOLD
          EXECIO 1 DISKW SRCHTEMP RESULTS A (STRING &DOCUNEW
          EXECIO 1 DISKW SRCHTEMP RESULTS A (STRING &DOCUOLD
          EXECIO 1 DISKR SEARCH RESULTS A
          &LOOP -BOTTOM UNTIL &RC NE 0
              &READ STRING &RAWDATA
*           ----- IF &FIDTEST = THEN THE LINE IS A FILE ID.
*           ----- OTHERWISE, IT IS A FOUND LINE, READY TO BE TESTED.
          &FIDTEST = &SUBSTR OF &RAWDATA 1 1
          &IF &FIDTEST NE &GOTO -TESTLOC
          &FILEID = &RAWDATA
          &FIDWRIT = 0
          &GOTO -BOTTOM
*           ----- IGNORE LINE NUMBER WHEN LOCATING STRING
-TESTLOC &MODDATA = &SUBSTR OF &RAWDATA 7 *

```

```

&LOCVAR = &LOCATION OF &ARGSTRING &MODDATA
* ----- IF THE LOCATION TEST RESULT IS TRUE, DISCARD LINE
&IF &LOCVAR &RELOPER &LOCPARM &GOTO -BOTTOM
* ----- ENSURE THAT FILE ID ONLY WRITTEN ONCE.
&IF &FIDWRIT = 1 &GOTO -ONEFID
EXECIO 1 DISKW SRCHTEMP RESULTS A ( STRING &FILEID
&FIDWRIT = 1
-ONEFID EXECIO 1 DISKW SRCHTEMP RESULTS A ( STRING &RAWDATA
-BOTTOM EXECIO 1 DISKR SEARCH RESULTS A
FINIS SRCHTEMP RESULTS A
ERASE SEARCH RESULTS A
REN SRCHTEMP RESULTS A SEARCH RESULTS A

```

### SRCHVAR EXEC

```

&TRACE ERR
* ----- SRCHVAR EXEC IS PART OF SEARCH ROUTINE.
* ----- SEE S EXEC, SRCHHELP EXEC, AND WRITTEN REPORT.
* ----- AUTHOR: RALPH SEELEY (202)447-8815.
* ----- VERSION: 3-12-84. LAST MODIFICATION: 3-12-84 RMS.
* ----- PROGRAM SEARCHES OUTPUT FILE. DOCUMENTATION SECTION
* ----- IS PUT IN TEMPORARY FILE. IGNORING LINE NUMBERS,
* ----- EACH FOUND LINE IS TESTED FOR
* ----- PRESENCE OF SEARCH STRING, NEITHER
* ----- PRECEDED NOR FOLLOWED BY LETTERS OR NUMBERS.
* ----- LINES MEETING TEST ARE PUT IN TEMPORARY FILE,
* ----- AS ARE NON-CONSECUTIVE FILE IDS. COMPLETED
* ----- TEMPORARY FILE IS NAMED OUTPUT FILE.
SET CMSTYPE HT
&FIRSTCHAR = &SUBSTR OF &ARGSTRING 1 1
* ----- A CHARACTER GREATER THAN OR EQUAL TO 'A' IS A LETTER
* ----- OR NUMBER. A CHARACTER LESS THAN 'A' IS NEITHER A
* ----- LETTER NOR A NUMBER.
&IF &FIRSTCHAR GE A &IF &FIRSTCHAR NE 9 &FILLCHAR = 9
&IF &FIRSTCHAR GE A &IF &FIRSTCHAR = 9 &FILLCHAR = 8
&IF &FIRSTCHAR LT A &IF &FIRSTCHAR NE @ &FILLCHAR = @
&IF &FIRSTCHAR LT A &IF &FIRSTCHAR = @ &FILLCHAR = ¢
&STRGTRUNC = &SUBSTR OF &ARGSTRING 2 *
&FILLSTRG = &CONCAT OF &FILLCHAR &STRGTRUNC
* ----- GET DOCUMENTATION OF LAST COMMAND AND SEARCH STRING
&READ STRING &DOCUNEW
EXECIO 1 DISKR SEARCH RESULTS A
ERASE SRCHTEMP RESULTS A
* ----- CREATE TEMPORARY FILE WITH RECFM V (LRECL UP TO 255)
EXECIO 1 DISKW SRCHTEMP RESULTS A 1 V
EXECIO 1 DISKR SEARCH RESULTS A
&READ STRING &DOCUOLD
&DASHLOC = 0
* ----- PUT DOCUMENTATION SECTION IN OUTPUT FILE.
&LOOP -DOCUEND UNTIL &DASHLOC = 6
EXECIO 1 DISKW SRCHTEMP RESULTS A (STRING &DOCUOLD
EXECIO 1 DISKR SEARCH RESULTS A
&IF &RC NE 0 &EXIT
&READ STRING &DOCUOLD
-DOCUEND &DASHLOC = &LOCATION OF ----- &DOCUOLD

```

```

EXECIO 1 DISKW SRCHTEMP RESULTS A (STRING &DOCUNEW
EXECIO 1 DISKW SRCHTEMP RESULTS A (STRING &DOCUOLD
&LENVAR = &LENGTH OF &ARGSTRING
EXECIO 1 DISKR SEARCH RESULTS A
&LOOP -BOTTOM UNTIL &RC NE 0
  &READ STRING &RAWDATA
*      ----- IF &FIDTEST =  THEN THE LINE IS A FILE ID.
*      ----- OTHERWISE, IT IS A FOUND LINE, READY TO BE TESTED.
      &FIDTEST = &SUBSTR OF &RAWDATA 1 1
      &IF &FIDTEST NE  &GOTO -TESTLOC
      &FILEID = &RAWDATA
      &FIDWRIT = 0
      &GOTO -BOTTOM
*      ----- IGNORE LINE NUMBER WHEN LOCATING STRING
-TESTLOC &MODDATA = &SUBSTR OF &RAWDATA 7 *
        &LOOP -FIND *
          &LOCVAR = &LOCATION OF &ARGSTRING &MODDATA
*          ----- A LOCATION OF ZERO MEANS STRING NOT FOUND
          &IF &LOCVAR = 0 &GOTO -BOTTOM
          &LOCPLUSLEN = &LOCVAR + &LENVAR
*          ----- THE BEGINNING OF THE LINE MAY BE A DELIMITER
          &IF &LOCVAR = 1 &GOTO -SUFFIX
          &LOCLESS1 = &LOCVAR - 1
          &PREFIX = &SUBSTR OF &MODDATA &LOCLESS1 1
          &IF &PREFIX GE A &GOTO -FILL
-SUFFIX  &LENMOD = &LENGTH OF &MODDATA
*          ----- THE END OF THE LINE MAY BE A DELIMITER
          &LOCPLUSLENLESS1 = &LOCPLUSLEN - 1
          &IF &LOCPLUSLENLESS1 GE &LENMOD &GOTO -WRITE
          &SUFFIX = &SUBSTR OF &MODDATA &LOCPLUSLEN 1
          &IF &SUFFIX LT A &GOTO -WRITE
-FILL   &AFTERSTRG = &SUBSTR OF &MODDATA &LOCPLUSLEN
*          ----- CREATE NEW LINE STARTING WITH FILLED STRING
-FIND   &MODDATA = &CONCAT OF &FILLSTRG &AFTERSTRG
*          ----- ENSURE THAT FILE ID ONLY WRITTEN ONCE.
-WRITE  &IF &FIDWRIT = 1 &GOTO -ONEFID
        EXECIO 1 DISKW SRCHTEMP RESULTS A (STRING &FILEID
        &FIDWRIT = 1
-ONEFID EXECIO 1 DISKW SRCHTEMP RESULTS A (STRING &RAWDATA
-BOTTOM EXECIO 1 DISKR SEARCH RESULTS A
        FINIS SRCHTEMP RESULTS A
        ERASE SEARCH RESULTS A
        REN SRCHTEMP RESULTS A SEARCH RESULTS A

```

## REFERENCES

1. Computer Science Division, Department of Electrical Engineering and Computer Science. UNIX Programmer's Manual: Virtual VAX-11 Version, University of California, Berkeley, California, June 1981.
2. International Business Machines Corporation. IBM Virtual Machine Facility/370: Terminal User's Guide, File Number S370-40, Order Number GC20-1810-9, Endicott, New York, March 1980.
3. \_\_\_\_\_. IBM Virtual Machine/System Product: CMS Command and Macro Reference, Program Number 5664-167, Order Number SC19-6209-1, Endicott, New York, March 1982.
4. \_\_\_\_\_. IBM Virtual Machine/System Product: CMS User's Guide, Program Number 5664-167, Order Number SC19-6210-1, Endicott, New York, April 1982.
5. \_\_\_\_\_. IBM Virtual Machine/System Product: CP Command Reference for General Users, Program Number 5664-167, Order Number SC19-6211-1, Endicott, New York, April 1982.
6. \_\_\_\_\_. IBM Virtual Machine/System Product: EXEC2 Reference, Program Number 5664-167, Order Number SC24-5219-1, Endicott, New York, April 1982.
7. \_\_\_\_\_. IBM Virtual Machine/System Product: System Product Editor Command and Macro Reference, Program Number 5664-167, Order Number SC24-5221-1, Endicott, New York, April 1982.
8. \_\_\_\_\_. System Productivity Facility for VM/SP: Program Reference, Program Number 5668-009, Order Number SC34-2047-0, Endicott, New York, March 1981.
9. Walker, Terry M. Introduction to Computer Science, Allyn and Bacon, Boston, 1972.



Character String: a consecutive group of characters on one line of a file.

Default: a value which is used by the program when the user does not enter a value at the terminal.

Fileset Identifier: the designation of files to be searched. The user may specify one of five sorts of fileset identifiers: a unique identifier for one file; a group identifier for several files with some similarity in their identifiers; a default group identifier; the identifier of the previous search output file; or an SFILES list of file identifiers. For additional information, see the section of the article on identification of files to search.

Identifier: the unique designation for a particular file. The identifier consists of a filename, a filetype, and a filemode.

Line Number: the number of a line with respect to the top of the file in which it is found. The first line in a file is line number one, and so on.

List of File Identifiers: several file identifiers kept in a special file. The special file has filetype SFILES. Exactly one file identifier is placed on each line of the SFILES file. (see also SFILES List)

Menu: a list of the possible commands or types of input that the user may enter at a given point in the program.

Null Input: the result of responding to a request from the search command for terminal input by pressing the ENTER button without typing anything else.

Search String: the character string which the search command tries to find in the file or files specified. The search string may be treated simply as a string, or its context may be checked, as for a word or variable.

SEARCH RESULTS: the output file which results from searching. When the SEARCH RESULTS file is printed, its identifier may be temporarily modified. Most importantly, the first word of the initial search string may be used temporarily for the filetype.

SFILES List: a special file with filetype SFILES. It contains a list of file identifiers. If the SFILES list is specified in the fileset identifier, the SFILES file containing the list is not searched. Instead, the files mentioned in the list are searched.

Variable: (see Word)

Word: as defined here, a character string which is neither immediately preceded nor followed by a letter or number. The designations "word" and "variable" are used interchangeably in this paper.



1022323015

Asterisk (\*) iv, 2-3  
 At (@) Symbol 9  
 Blanks in Search String iv, 2, 6  
 Break During Search 5, 11  
 Command Limitations 9-10  
 Command Name iv, 2, 10  
 Defaults:  
     Fileset Identifier iv, 3-4  
     Search Technique 4  
     SFILES List Fileset Identifier 3-4  
 Filetypes:  
     Definition 2  
     Line Numbers 4-5, 10  
 Further Processing Menu 6-7  
 IDs (see Identifiers)  
 Identifiers:  
     File 2  
     Fileset 2-4  
 List of File Identifiers (SFILES List) iv, 3-4  
 Menus:  
     Fileset Identifier 2-4  
     Further Processing 6-7  
     Initial Search Technique 4-5  
 Narrative Program Explanation 12-13  
 Percent Sign (%) iv, 2-3  
 Program Function (PF) Keys 5-6, 11-12, 42  
 Programs Which Make Up Search Command:  
     S EXEC 14-17, 27-32  
     SRCHCHNG EXEC 21, 32-33  
     SRCHCLRS EXEC 21, 33  
     SRCHHELP EXEC 22, 33-37  
     SRCHLOCS EXEC 18, 38-39  
     SRCHLOCV EXEC 19-20, 39-41  
     SRCHNFNH XEDIT 22, 41-42  
     SRCHPROF XEDIT 42  
     SRCHSFIL EXEC 23, 43-44  
     SRCHSTRG EXEC 24, 44-45  
     SRCHVAR EXEC 25-26, 45-46  
 Search Session, Sample 7-8  
 Search Results, Sample 9  
 Search String iv, 2, 6  
 Search Techniques:  
     Further Processing iv, 6-7, 10-11  
     Initial Search iv, 4-5, 10-11  
 SFILES List:  
     Creation 3-4  
     Use iv, 3-4  
 Speed of Search 4-5, 12  
 Syntax:  
     Detailed Explanation 1-9  
     Help Document 33-37  
     Summary iv  
 Variable, Definition 4-5, 10-11, 48  
 Word, Definition 4-5, 10-11, 48  
 XEDIT 5-6, 11-12

\* NATIONAL AGRICULTURAL LIBRARY



1022323015

