DECUS
PROGRAM LIBRARY

DECUS NO.	L-53
TITLE	FIND 1
AUTHOR	Richard A. Harshman
COMPANY	University of California Los Angeles, California
DATE	June 1968
SOURCE LANGUAGE	LAP6

Although this program has been tested by the contributor, no warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related program material, and no responsibility is assumed by these parties in connection therewith.

FIND 1

Find 1 allows the user to define categories or classes of data sets to be searched for in a large file. It then locates and retrieves relevant data from the files stored on magnetic tape. Files can be written in natural language (e.g. English) and entries need not be specially coded for subject headings or cross references. It is possible to search for data fitting into categories not anticipated when the file was created.

The files are created and stored as LAP6 manuscripts, and can be created, edited or updated by a typist or secretary who has no special knowledge of computers after only short training on the use of LAP6.

Basic Functions

FIND 1 can scan files and

1. automatically recognize, extract and type all entries related to certain areas of interest; (these interest areas are defined at the time of search and need not have been anticipated when the file was made); it will recognize entries containing particular bits of data in particular relationships

2. type out, if desired, only specific sections of those entries which it locates -- (find the entries of interest and then type out only the data requested from each of these entries)

3. inform the user of the location in the file of all entries of interest (by giving their line numbers in the LAP6 manuscript which comprises the file)

Typical applications

1. automatic bibliographic search

2. medical histories (e.g. find the heart rate and patient number for all patients that showed a certain conjunction of characteristics)

3. automated mailing lists (type out labels for only those people on the list living in a certain area, or who have received a certain prior notice and responded, etc.)

4. subject indexes, author indexes, concordances, etc. can be created by repeated searching with FIND 1

5. abstracts of research articles or reports can be filed, and searches can be made for all reports that mention certain key terms but not others, etc.

6. personnel files (what salary is earned by all individuals fulfilling certain descriptions, and who are these persons)

7. debugging aid for LAP6 programs (e.g. locate all program manuscript lines which refer to sense switches, type these out with line number; or locate all JMP 3T+2 lines, etc.)

How to Use FIND 1

A. Manuscript format: FIND 1 is a companion program to LAP6 and assumes that the user is familiar with the operation of the various LAP6 meta-commands for the creation, editing, and filing of manuscripts.

Manuscripts for FIND 1 are prepared using the standard LAP6 procedures. When typing in lines, care must be taken not to exceed 70 characters since FIND 1 does not automatically return the carriage when a teletype line is full. When viewed on the oscilloscope, 70 characters equal about 3 1/2 rows of characters per numbered line.

Type in entries in the format in which you would like to see them typed out. It is sometimes useful to divide complicated entries into sections using special characters (e.g. the origin symbol followed by a number). This facilitates later selection of parts to type out, or parts to search for specific data. (See example of a FIND bibliography manuscript, appendix 1).

A suitably constructed LAP6 manuscript can be considered a file, which must be divided into entries. The basic function of FIND 1 is to search through such a file to locate those entries which contain certain combinations of words or data in certain relationships as specified by the person using the program to search. In order for the FIND program to discriminate between one entry and the next, a special character must be reserved by the person typing in the file, and placed between entries. This special character is called the entry delimiter. An entry delimiter must occur (a) before the first entry in the file; (b) between each succeeding pair of entries; and (c) after the final entry in the file. Any characters typed before the first entry delimiter or after the last entry delimiter will not be searched by the FIND program, but may still be useful as comments or for identification of the file.

With FIND 1R (current version) the maximum size of an entry is about 1530 characters (3 blocks worth). The maximum size of the manuscript is limited only by the size of the LAP6 working area. FIND 1 could search

an entire tape. For convenience long manuscripts can be broken up into several shorter pieces which are searched independently.

B. Calling FIND 1 up for use: The FIND program is written as a special meta-command of LAP6. (It uses the "Free" meta-command.) To call up FIND 1 on a suitably modified version of LAP6, simply type case, EOL (to LF get the meta arrow) then FI,EOL. This will provide a graceful exit from LAP6 (except that if you reenter manually you must reenter with start 17, rather than start 20). After exiting LAP6, it will present you with the FIND program option display (see figure 1).

→ FI_{EOL}

```
SPECIFY ?

1=SEARCH WORDS

2=SEARCH RELATIONS

3=PARTS TO TYPE OUT

4=ENTRY DELIMITER

5=TEXT LOCATION

6=GO SEARCH TEXT

7=RETURN TO LAP6
```

Figure 1: Option Display for FIND 1 Program

C. Option display: Options 1 through 5 allow the user to specify different sorts of information needed by the FIND program. Options 6 and 7 command the program to do something. To specify an option, type the number of that option (this number should replace the "?") and then the EOL ("Return" key for the LINC-8). For example, to specify bits of data or words to be recognized within entries, type 1_{EOL}.

D. Options:

1. SEARCH WORDS

When this option is selected, the following display appears:

```
KEY WORDS ARE

A
??????????????????????
B
??????????????????????
C
??????????????????????
D
??????????????????????
E
??????????????????????
F
??????????????????????
```

Below each letter (A, B, etc.) is a row of 22 question marks. This is a field reserved for holding the strings of characters which are to be recognized by the FIND program. These are key words or sequences of data that help pick out an entry. At this point you are just specifying the relevant character strings, not the relations among these which will determine if an entry is to be typed out. Relations will be specified by option 2.

1. Type in the strings of characters you wish to have FIND recognize. These strings may include spaces, and any special characters except "EOL", "#", or the colon (":"). (The question-and-answer subroutine confuses the colon -- code 76 -- with an end-of-data-field, EOL is used to progress to the next data field, and # is used to signify end of a string, see below).

2. When you have completed a string (any length from 1 to 21 characters) mark the end of the character string with the cross hatch symbol used to designate TAGS in program manuscripts ("#").

3. Then hit the EOL bar, and the remaining question marks after the # on that line will disappear. This proceeds automatically to the next row of question marks, ready to accept a new string of characters.

4. Each character string has a name. Its name is the letter right above it. You will have to remember these names for option 2, so it may help to make a note of them and what they stand for.

5. If there are more rows than you need, simply hit EOL repeatedly and fill the unneeded rows with blanks, until you reach the bottom of the page. Control will return to the option display unless you have sense switch raised.

Display pages for input of additional words

Normally, when you have filled in all 6 rows of question marks with characters or blanks, control will return to the option display. If you need to specify more strings of characters ("key words"), lift Sense Switch 0 before reaching the end of a page. When you finish with that page, control will then go to an additional page of named fields of question marks (G-L, or M-R). When on a new page, lower Sense Switch 0 if you don't wish to progress to a third additional page.

Variable character

Sometimes, you wish to specify the characters at only certain locations in a character string, and let characters at other locations be anything whatsoever. To allow this, a special character is used to represent a variable character, or place holder. The character "p" (small p) is used as a place holder by FIND 1. An example of an application would be selecting all number or letter codes that have certain portions as specified but other portions don't matter. Another use would be in inputting words with varied spelling.

Example: To search the Bibliography of LINC Related Publications for all articles related to analysis of unit activity, the following character strings might be selected:

```
KEY WORDS ARE

A
UNIT#
B
SPIKEpINTERVAL#
C
SINGLE SPIKE#
D
INTERVAL#
E
(ABSTRACT)#
F
FREQUENCY#
```

and on the next page

```
G
CODING#
H
EEG#
I
1968#
J
1967#
K

L
```

(Note for example that SPIKEpINTERVAL did not specify what character was to be between SPIKE and INTERVAL, nor did it specify a leading or trailing space, so "interspike intervals" or "interspike-interval" would also fulfill it.)

The next step would be to specify the relations among these character strings which would allow an entry to qualify. This is done with option 2.

2. SEARCH RELATIONS

When this option is selected, the following display appears:

```
SEARCH FOR

??????????????????????

??????????????????????

??????????????????????

??????????????????????

??????????????????????

??????????????????????
```

Six fields of question marks are provided for specifying relations among the key words (character strings) named in option 1. In this way, the FIND 1 program will not only test for the presence of certain words, but also check to see that the relationship among the words is as desired, before typing out an entry.

There are 4 basic types of relations that can be specified. These are given in the following table:

Relation	Symbolized by	Examples	Interpretation
AND	(grouping letters together; separate lines are also ANDed)	AB	A and B
		C DE	C and D and E
OR	comma (,)	A,BC,D	A, or (B and C) or D
NOT	asterisk (*)	BC*A,*BD	(B and C and not A) or ((not B) and D)
FOLLOWED BY	dash (-)	A-B,C D-B-E	([(A followed by B) or C] and [D followed by B followed by E])

Letters specifying character strings may be combined in any order using these operators and thus very complicated logical relationships can be specified if required. Each line allows a disjunction ("or" relation) of a number of conjunctions ("and" relations) to be specified. Further, all the individual lines are ANDed together, because an entry must pass the test specifications specified on every line before it is acceptable to be typed out.

There is one exception to the statement that the operators may be combined in any order. The "not" operator may occur only in the last term of a "followed by" sequence. This is because the computer must go to the end of an entry to see if something is not present, but when it is at the end, it cannot properly check for further "followed by" relationships. A-B-*D is OK, as is C-B-*B. But *A-B is no good.

Example: Let us continue our example of the search for information relating to single unit activity in the LINC bibliography. If we assume that the key words are defined as in the prior example (page 5), we might specify the following relationships:

```
SEARCH FOR

A,B,C,D,FG

*E,*H

I,J
```

This would specify a search for those entries satisfying the following three criteria: (1) they must contain either the word UNIT or SPIKEpINTERVAL or SINGLE SPIKE or INTERVAL or both FREQUENCY and CODING; (2) they must contain neither the word (ABSTRACT) nor EEG; (3) they must contain 1967 or 1968.

This would select articles related to unit activity but it would reject abstracts (assuming that all abstracts were so labeled in the bibliography). It would accept only articles published in the last two years. The not-EEG condition helps reject articles that might use some of the key words in the title but are really about EEGs (e.g. EEG FREQUENCY CODING).

This set of conditions is not the best we could specify. For example, it would reject entries that mention both Units and EEGs. It is also redundant, since D (INTERVAL) would automatically include all cases of B (SPIKEpINTERVAL). Better planning would have removed these faults. Nonetheless, it is a good example of how one might begin to go about defining areas of interest in terms of logical relations among words occurring in file entries.

In the example given, only three of the six lines of question marks

were used. Continuing to hit EOL blanks out any that are not used. The sixth EOL returns control to the option display.

3. PARTS TO TYPE OUT

A. If this option is not selected at all, any file entry that passes the tests specified in 2 will be typed out in full. This is a default specification (default because of failure to specify anything different).

B. If this option is selected, the following display will appear:

```
TYPE OUT FROM

???????????
TO
???????????
AND FROM
???????????
TO
???????????
AND FROM
???????????
TO
???????????
```

Character strings should be entered into this display with the same conventions as in the option 1 "Key Words" display. The end of each string should be marked by a "#".

The computer will locate the specified character strings in each acceptable entry and then type out that text which occurs between each pair of strings. ("Between" means it will type only the last character of that string marking the beginning of each section to type out, and it will not type any of the characters in the string marking the end of each section to type out.)

From one to three sections can be specified. These can be overlapping, independent, or even include one another.

Example: To type out the entire entry either do not choose the option in the first place, or if it had been previously specified as some other partial type-out, re-specify as the following: