

1984/77. Retrieval of records and data using FORTRAN program SEARCH

E.L. Martin

Abstract

Operating details and a program listing are given for FORTRAN program SEARCH, a program primarily intended for the retrieval of multi-line records from unstructured bibliographic data bases.

INTRODUCTION

SEARCH is a FORTRAN program primarily intended for the retrieval of information from unstructured bibliographic data bases consisting of multi-line records. Search is by keyword (found or not found). The program can also be used for searching files consisting of single line records. The maximum file record length is 256 characters per line.

Only one keyword can be sought at one pass, but the search results may be saved and used for a new search.

ACCESS TO THE PROGRAM

A user who is signed on to the Publications Access account may search a file in either that account, or the group account. The group account contains the main bibliographic data bases. Other publications files (on disc and tape) include the text for Explanatory Reports (e.g. Maria, Strahan); these may also be searched - e.g. for fossil names or localities. Tape files must be copied to disc before resuming SEARCH.

Users who wish to use the program with files in their own accounts should type SET GROUP 70 and then COPY SEARCH.CSS/G,SEARCH.CSS.

USING THE PROGRAM

The program is called by typing SEARCH fd[,dev:] where fd is the file descriptor of the file you wish to search, and dev: [optional] is the device to be used for printing the results of your search, if dev: is not specified the default is SPR: If SPR: is used set SKIP to ON.

The log of a typical session illustrating prompts and user input is included as Appendix 1.

If the search fails the response will be: No match found. You will be then asked if you wish to enter another keyword.

If you wish to search for records which do NOT contain a particular keyword, then follow the string delimiter by N (e.g. Devonian]N). The number of records in which the string has been FOUND will be specified, but all other lines will be output. If the keyword does not occur in any record the program response will be: No match found, as there is no point in outputting the entire file!

Although the search results are also output to the screen, the number of lines output is restricted to 50, as it is more convenient to view a hard copy version, or use CAT or EDIT on the output file (SEARCH.TMP) when a large number of items are retrieved.

REMEMBER

- (1) Follow your input string by the appropriate terminator (e.g. landslip]) do not leave a space unless it is to form part of the string. There are times when a leading space might be useful (e.g. if you wanted to search for TIN, as tin is contained in many longer words.
- (2) Use stems to retrieve similar keywords (e.g. landsl] for landslip(s) and landslide(s)).
- (3) If you plan a multiple search, search for the keyword less likely to be found first.
- (4) Scanning is line by line. If searching a bibliographic file for terms consisting of more than one word (e.g. seismic refraction) where one word may occur at the end of a line and the next word on the succeeding line, it is better to search for one word at a time, saving the first search file and then using it as the input for the next search.
- (5) Very long files may take an appreciable time to search. Allow about 30 seconds for each thousand average sized records. Searching for an exact match will reduce search time.
- (6) Search results are accumulated in the file SEARCH.TMP as successive searches are made, as long as you remain in the program. Should you exit the program and then re-run it your previous results will be lost. If you choose the option of keeping the search file before you exit the program you will be warned to rename it.
- (7) Delete all search files no longer required.

[16 November 1984]

EXAMPLES OF PRINTED OUTPUT

File: PRST.BIB/G Keyword: Smithton

CAREY,S.W.; SCOTT,B. 1952. Revised interpretation of the geology of the
Smithton district. Pap.Proc.R.Soc.Tasm. 86:63-70.
CAREY,S.W.; SCOTT,B. 1954. Native copper at Smithton - a correction.
Pap.Proc.R.Soc.Tasm. 88:271-272.

File: UR84.TXT/G Keyword: SEISMIC

1984/44	Seismic survey at the Craigow dam, Cambridge.	W.L.Matthews	22. 6.84
1984/60	A preliminary seismic refraction survey at a proposed dam site at Grindelwald, West Tamar.	W.R.Moore	14. 9.84
1984/75	Seismic survey of a proposed dam site at Craigow University Farm, Cambridge.	W.R.Moore	9.11.84

APPENDIX 1

The following log of a typical session illustrates user input, preceded by * or > prompts, and program and system responses.

*SEARCH PRST.BIB/G
14/11/84 10:29:16

Specify file type:

- B for bibliographic
- D for data, or single line records

>B

Enter keyword or string to be matched, terminate by | to enable UPPER/lower case search or by } for exact match only.

If you wish to find records that do NOT contain a match with your search string then follow the string delimiter by N

]

>Smithton}

Number of records found 2: 628 lines in file
The search results will be output to the screen and to the disc file SEARCH.TMP

Do you also want a hard-copy printout ? (Y/N)

>Y

CAREY,S.W.; SCOTT,B. 1952. Revised interpretation of the geology of the Smithton district. Pap.Proc.R.Soc.Tasm. 86:63-70.
CAREY,S.W.; SCOTT,B. 1954. Native copper at Smithton - a correction. Pap.Proc.R.Soc.Tasm. 88:271-272.

Do you wish to search for another keyword? (Y/N)

>N

Do you wish to keep search file ? (Y/N)

>N

TED -END OF TASK CODE= 0 CPUTIME=9.958/0.741
14/11/84 10:30:03

APPENDIX 2

```

* SEARCH.CSS RUNS SEARCH.TSK
$IFNUL @1; $WR INPUT FILE MUST BE SPECIFIED; $EXIT; $ENDC
$IFNX @1; $WR @1 DOES NOT EXIST; $EXIT; $ENDC
XAL S.TMP,IN,80; XDE SEARCH.TMP; L SEARCH/G;5
D F,@1,S.TMP; AS 7,S.TMP; FR 7; FR 7
AS 1,@1,SRO; AS 3,CON:
$IFNUL @2; AS 4, SPR:
$ELSE; AS 4,@2
$ENDC
AS 5,CON:; D T; ST; XDE S.TMP
D T; $EXIT

```

APPENDIX 3

```

*TITL SEARCH.FTN  VERSION 2.0  13-NOV-84  E.L.MARTIN
C ***** SEARCHES A BIBLIOGRAPHIC OR DATA FILE FOR A SEARCH STRING
C OPERATES USING SEARCH.CSS - COMMAND IS SEARCH fd[,dev:]
C WHERE fd IS THE FILE TO BE SEARCHED (LU 1), AND dev: IS OUTPUT DEVICE
C OUTPUT IS TO LU 5 (CON:>) AND TO A DISC FILE (SEARCH.TMP) ON LU 8
C A PRINTOUT CAN ALSO BE ENABLED (LU 4 - DEFAULT = SPR:>)
C PROGRAM OPERATED BY SEARCH.CSS
C
C ALINE      Input line read from LU 1
C FLINE      Contains directory details of input file
C HYLINE     A line of hyphens
C FNAME      Name of input file
C FILESTAT   Close status of SEARCH.TMP output file
C RECL,NCL   Record length (number of characters/line) of input file
C RECS,NRECS Number of lines in input file
C NDEX       Index array for lines found during search
C EOF1,EOF2  Search string delimiters 1, 2
C CHARS      Array containing search string
C CHARSA     Alternative array (U/C, l/c switched)
C FMT1,FMT2  Contain output formats
C ALPHA,BETA Upper/lower case data arrays
      CHARACTER*256 ALINE
      CHARACTER*80 FLINE
      CHARACTER*72 HYLINE
      CHARACTER*20 FNAME
      CHARACTER*9  FMT2
      CHARACTER*7  TYPE
      CHARACTER*6  FILESTAT,FMT1,RECS
      CHARACTER*3  RECL
      CHARACTER*1  ALPHA(26),BETA(26),CHARS(73),
      CHARSA(72),HYP,BLANK,EOF1,EOF2,ANS
      INTEGER*2    NDEX(2000),K
C Initialise data arrays
      DATA (ALPHA(I),I=1,26)/'A','B','C','D','E','F','G','H','I','J',
      .'K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/,
      (BETA(I),I=1,26)/'a','b','c','d','e','f','g','h','i','j','k',
      .'l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'/
      DATA HYP/'-'/,BLANK/' '/,EOF1/'1'/,EOF2/'2'/
C Get details of input file
      READ(7,907)FLINE
907 FORMAT(A80)
C Find record length and convert to integer
      RECL=FLINE(34:36)
      NCL=CTOI(RECL,K)
      FMT1='(A'//RECL//)''
      FMT2='(1X,A'//RECL//)''
      RECS=FLINE(39:44)
      NRECS=CTOI(RECS,K)
C Construct line of hyphens for use in printout
      DO 2 I=1,72
        HYLINE(I:I)=HYP
      2 CONTINUE
C Ask whether bibliographic (multiline record) or data file
      5 WRITE(5,903)
903 FORMAT('/ Specify file type:'// ' B for bibliographic'/
      .' D for data, or single line records'//)
      READ(3,801)ANS
      KF=-1
      IF(ANS .EQ. 'B' .OR. ANS .EQ. 'b')KF=0
      IF(ANS .EQ. 'D' .OR. ANS .EQ. 'd')KF=1
C Loop back if not B or D
      IF(KF .LT. 0)GOTO 5
C Set flags: KD Match type; KE End of file; KF File type;
C KI End of index array; KN No match option
      10 KD=0
      KE=0
      KI=0
      KN=0
      KP=0
C Open input file
      OPEN(UNIT=8,FILE='SEARCH.TMP',RECL=NCL,FORM='FORMATTED',ACCESS=
      .'DIRECT')

```

```

C Get search string and check if match or no match option
120 WRITE(5,920)
920 FORMAT(' Enter keyword or string to be matched, terminate by ]'/'
.' to enable UPPER/lower case search or by } for exact match only.'
.'//'' If you wish to find records that do NOT contain a match with''/
.' your search string then follow the string delimiter by N''/
.'73X,'1'')
READ(3,921,ERR=120)CHARS
921 FORMAT(73A1)
DO 121 I=1,72
CHARSA(I)=CHARS(I)
121 CONTINUE
C Find length of input string
DO 122 I=1,72
IF(CHARS(I) .EQ. EOF1 .OR. CHARS(I) .EQ. EOF2)LENGTH=I-1
IF(CHARS(I) .EQ. EOF2)KD=1
122 CONTINUE
904 FORMAT(1X,A72/)
C Set NOT FOUND flag
IF(CHARS(LENGTH+2) .EQ. 'N' .OR. CHARS(LENGTH+2) .EQ. 'n')KN=1
IF(CHARS(LENGTH+3) .EQ. 'N' .OR. CHARS(LENGTH+3) .EQ. 'n')KN=1
C Loop back if null input
IF(LENGTH .EQ. 0)GOTO 120
C UPPER/lower CASE SWITCH
DO 123 I=1,LENGTH
DO 124 J=1,26
IF(CHARS(I) .EQ. ALPHA(J))CHARSA(I)=BETA(J)
IF(CHARS(I) .EQ. BETA(J))CHARSA(I)=ALPHA(J)
124 CONTINUE
123 CONTINUE
LINUM=0
C I = Number of current line; K = Number of records found
I=1
K=0
NDEX(1)=0
C Skip if data file
IF(KF .EQ. 1)GOTO 230
C Read line from input file
130 READ(1,FMT1,END=60,ERR=9)ALINE(1:NCL)
C LINUM = Line number of first line of record
LINUM=LINUM+1
135 IF(ALINE(1:1) .NE. BLANK)IS=LINUM
C M=Character position in search string
M=1
C Check each character in input line
DO 132 IC=1,NCL
IF(KD .EQ. 1)GOTO 133
IF(ALINE(IC:IC) .NE. CHARS(M) .AND. ALINE(IC:IC) .NE. CHARSA(M))
.GOTO 131
GOTO 134
133 IF(ALINE(IC:IC) .NE. CHARS(M))GOTO 131
C IF M=LENGTH then match found so check for continuation lines
134 IF(M .EQ. LENGTH)GOTO 136
M=M+1
GOTO 132
131 M=1
132 CONTINUE
C Go back to read next line
GOTO 130
C Check only for single line records
C Read line from input file
230 READ(1,FMT1,END=60,ERR=9)ALINE(1:NCL)
LINUM=LINUM+1
C M is character position in search string
M=1
DO 232 IC=1,NCL
IF(KD .EQ. 1)GOTO 233
IF(ALINE(IC:IC) .NE. CHARS(M) .AND. ALINE(IC:IC) .NE. CHARSA(M))
.GOTO 231
GOTO 234
233 IF(ALINE(IC:IC) .NE. CHARS(M))GOTO 231
C If M = LENGTH then match found
234 IF(M .EQ. LENGTH)GOTO 236

```

```

M=M+1
GOTO 232
231 M=1
232 CONTINUE
GOTO 230
236 K=K+1
C Add line number to index array
NDEX(I)=LINUM
I=I+1
GOTO 230
C Get continuation lines if multiline record
136 READ(1,FMT1,END=138,ERR=9)ALINE
LINUM=LINUM+1
IF(ALINE(1:1) .EQ. BLANK)GOTO 136
GOTO 139
C Check for end of file
138 KE=1
LINUM=LINUM+1
139 IL=LINUM
C IS=First line of reference, IL=last line
L=IL-IS
C K=Number of records found
K=K+1
C Add line numbers to index array (NDEX)
DO 140 J=1,L
NDEX(I)=IS
IS=IS+1
I=I+1
140 CONTINUE
C Skip if end of file
IF(KE .EQ. 1)GOTO 60
GOTO 135
C Rewind input file ready to read back
60 REWIND 1
I=I-1
C If I <= 0 then no records found
IF(I .LE. 0)GOTO 160
C Display search results
WRITE(5,901)K,NRECS
901 FORMAT(' Number of records found ',I4,',',I6,' lines in file')
WRITE(5,800)
800 FORMAT(' The search results will be output to the screen and to th
.e disc file SEARCH.TMP.'/' Do you also want a hard-copy printout
.? (Y/N)')
READ(5,801)ANS
801 FORMAT(A1)
IF(ANS .EQ. 'N' .OR. ANS .EQ. 'n')KP=1
IF(KP .EQ. 1)GOTO 65
INQUIRE(UNIT=1,NAME=FNAME)
FNAME=FNAME(6:20)
C Set found/not found type
TYPE=' '
IF(KN .EQ. 1)TYPE=' .NOT. '
IF(LENGTH .GT. 32)LENGTH=32
WRITE(4,923)FNAME,TYPE,(CHARS(J),J=1,LENGTH)
923 FORMAT(' File: ',A14,2X,'Keyword: ',A7,3A1)
WRITE(4,904)HYLINE
65 IF(KN .EQ. 1)GOTO 75
DO 70 K=1,I
READ(1,REC=NDEX(K))ALINE(1:NCL)
C Output only first 50 lines to screen
IF(K .LE. 50)WRITE(5,FMT2)ALINE(1:NCL)
IF(K .EQ. 51)WRITE(5,924)
924 FORMAT('/ Only first 50 lines displayed !')
WRITE(8,FMT1)ALINE(1:NCL)
IF(KP .EQ. 1)GOTO 70
WRITE(4,FMT2)ALINE(1:NCL)
70 CONTINUE
IF(KP .EQ. 0)WRITE(4,927)
927 FORMAT(' ',/, ' ',/, ' '/')
GOTO 9

```

