CALLING COMMODORE

The variable search program that we developed for the BBC Micro and the Spectrum on pages 664 to 665 can easily be converted to work on the Commodore 64. The Commodore 64 program is, in fact, a little simpler because there are not so many special cases to allow for.

Many of the variable names in the Commodore version of the program have to be abbreviated to avoid including a BASIC keyword. For example, NEWLINE cannot be used as a variable name because it starts with NEW, and TEXTPOINTER cannot be used either because it includes INT.

The changes near the beginning of the program are necessary because of the differences in the way a line of BASIC is stored in the computer's memory. In the BBC Micro and the Spectrum, a line of BASIC in the internal format begins with a two-byte line number, with the high order byte coming first, and one or two bytes for the length of the line. In the Commodore 64 a line of BASIC begins with a two-byte pointer to the start of the next line and a two-byte line number, with the low order byte coming first in both.

We still have to skip REM lines and strings inside quotes, but we do not have to look for any other special cases that might cause confusion, like the hexadecimal numbers on the BBC Micro or the hidden binary form of numbers on the Spectrum.

The section of the program that actually picks out the variable names looks for a letter of the alphabet first, then letters or digits, and at the end it looks for a $ or % sign indicating a string or integer variable and a (, indicating a function or array. The Commodore 64 does not allow the underscore character that can be included in variable names on the BBC Micro and the Spectrum.

Although the Commodore 64 can display both upper and lower case letters, the difference is only in the form of the character as it appears on the screen, and not in the internal code for the character. Thus, the program only needs to look for upper case letters in a variable name.

The Commodore 64 version of the program is used in the same way as the BBC Micro and Spectrum versions. Type in the search program and SAVE it, then LOAD the program to be searched and append the search program to it. You can then search the program by "RUN 30000", and typing in the variable name when the program asks for it, ending the name with "(" if you want to find an array name.

There is a simple method of joining two SAVEd programs on the Commodore 64, provided the line numbers in the first program are all less than the line numbers in the second program. The method uses two of the pointers in page zero: TXTTAB, at addresses 43 and 44, which hold the address where the BASIC program starts, and VARTAB, at addresses 45 and 46. A BASIC program ends with a byte containing zero that marks the end of the last line of the program, then two more zero bytes that mark the end of the program. The address in VARTAB is normally the byte following the last of these zeros. To join two programs together, first LOAD the program with the lowest line numbers, then type:

```
PRINT PEEK(45), PEEK(46)
```

If the first number is between two and 255 subtract two from it and POKE the result into address 43. If it is zero or one, POKE 254 or 255 into address 43 and POKE one less than the result of PEEK(46) into address 44. You can then LOAD the second program, and finally type:

```
POKE 43,1: POKE 44,8
```

This will put the normal value back into the 'start of BASIC' pointer and the programs will now be joined together.

Commodore 64 Search Utility

```
30000 INPUT "NAME TO SEARCH FOR"; T$
30010 RE=143
30020 QUOTE=34
30030 NL=0
30040 TEXTPTR=2049
30050 NXTLINE=PEEK(TEXTPTR)+256*PEEK(TEXTPTR+1)
30070 TEXTPTR=TEXTPTR+2
30080 LINENO=PEEK(TEXTPTR)+256*PEEK(TEXTPTR+1)
30085 IF LINENO>=30000 THEN END
30090 TEXTPTR=TEXTPTR+2
30140 IF PEEK(TEXTPTR)=NL THEN TEXTPTR=TEXTPTR+1:GOTO 30050
30150 IF PEEK(TEXTPTR)<>RE THEN GOTO 30180
30160 REM SKIP OVER REM LINE
30170 TEXTPTR=NXTLINE:GOTO 30050
30180 IF PEEK(TEXTPTR)<>QUOTE THEN GOTO 30300
30190 REM SKIP ANYTHING IN QUOTES,STOP AT END OF LINE IN CASE OF UNMATCHED
      QUOTE
30200 TEXTPTR=TEXTPTR+1
30210 IF PEEK(TEXTPTR)=NL THEN TEXTPTR=TEXTPTR+1:GOTO 30500
30220 IF PEEK(TEXTPTR)<>QUOTE THEN GOTO 30200
30230 TEXTPTR=TEXTPTR+1
30235 GOTO30140
30290 REM FIRST CHARACTER OF NAME MUST BE LETTER
30300 IF PEEK(TEXTPTR)>=ASC("A") AND PEEK(TEXTPTR)<=ASC("Z") THEN GOTO 30330
30310 TEXTPTR=TEXTPTR+1
30320 GOTO30140
30330 NAME$=""
30340 NAME$=NAME$+CHR$(PEEK(TEXTPTR))
30350 TEXTPTR=TEXTPTR+1
30360 REM LETTER OR DIGIT AFTER FIRST CHARACTER
30370 IF PEEK(TEXTPTR)>=ASC("A") AND PEEK(TEXTPTR)<=ASC("Z") THEN GOTO 30340
30390 IF PEEK(TEXTPTR)>=ASC("0") AND PEEK(TEXTPTR)<=ASC("9") THEN GOTO 30340
30410 REM END WITH $ FOR STRING VARIABLE, % FOR INTEGER VARIABLE
30420 IF PEEK(TEXTPTR)=ASC("$") THEN NAME$=NAME$+"$":TEXTPTR=TEXTPTR+1
      :GOTO30450
30430 IF PEEK(TEXTPTR)=ASC("%") THEN NAME$=NAME$+"%":TEXTPTR=TEXTPTR+1
30440 REM ( IF ARRAY OR FUNCTION
30450 IF PEEK(TEXTPTR)=ASC("(") THEN NAME$=NAME$+"(":TEXTPTR=TEXTPTR+1
30460 IF NAME$=T$ THEN PRINT NAME$;" IN LINE";LINENO
30470 GOTO 30140
30480 END
```