ALL CHANGE

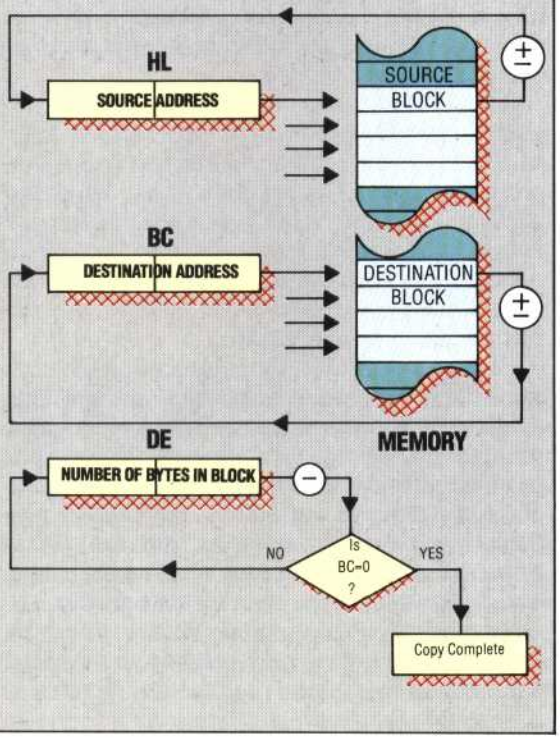
To implement the variable replace program on the Spectrum, we must use a different method from the BBC and Commodore versions. Instead of having the utility program in a different area of memory from the program it is working on, the variable replace program is merged onto the end.

As the variable replace program scans through the program it makes a copy of the altered version in an area above RAMTOP. The altered version is then copied back to the main program area by a machine code program that adjusts the amount of space available if the length of the program has been changed, so that the new version fits into the BASIC text area.

The first part of the BASIC program is similar to the variable search program (see page 665). There are some extra variables, including Altprog, which points to the start of the area reserved for the copy of the program, and Altpointer, which keeps track of where the next byte in the altered program should go. The main changes involve copying the program, instead of just reading it. The copying is done by the subroutine at line 9800, which copies

Automatic Writing

The Z80 LDIR and LDDR op-codes are block transfer instructions using automatic increment or decrement: the HL register is initialised to point to the start of the source block, DE must point to the start of the destination, and BT must contain the number of bytes in the block. LDIR and LDDR then copy the source byte to the destination byte, automatically increment or decrement HL and DE, and decrement BC until it reaches zero, when the copy is deemed complete. Notice that LDIR is a 'dumb' copy (see page 726) — the programmer's intelligence is assumed



KEVIN JONES

Variable Replace Program

```
9000 INPUT "Name to search for? "; LINE
t$
9005 INPUT "Replace by?"; LINE r$
9010 FOR i=1 TO LEN (t$)
9020 IF t$(i)>="a" AND t$(i)<="z" THEN
LET t$(i)=CHR$ (CODE (t$(i))-32)
9030 NEXT i
9040 LET TokenforREM=234
9050 LET Quote=34
9060 LET Newline=13
9070 LET Underscore=95
9080 LET Number=14
9090 LET PROG=23635
9100 LET Textpointer=PEEK (PROG)+256*PEE
K (PROG+1)
9102 LET Altprog=46000
9105 LET Altpointer=Altprog
9110 LET Lineno=256*PEEK (Textpointer)+P
EEK (Textpointer+1)
9111 PRINT lineno
9120 IF Lineno>=9000 THEN  GO TO 9600
9130 LET q=2: GO SUB 9800
9135 LET Lengthaddr=Altpointer
9140 LET Nextline=Textpointer+2+PEEK (Te
xtpointer)+256*PEEK (Textpointer+1)
9150 LET q=2: GO SUB 9800
9160 LET Byte=PEEK (Textpointer): LET q=
1: GO SUB 9800
9170 IF Byte=Newline THEN  GO TO 9110
9180 IF Byte<>TokenforREM THEN  GO TO 92
20
9190 REM Copy REM  unaltered
9200 LET q=Nextline-Textpointer: GO SUB
9800
9210 GO TO 9110
9220 IF Byte<>Quote THEN  GO TO 9280
9230 REM Copy anything between quotes, b
ut stop at end of line in case of unmatc
hed quote
9235 LET q=1
9240 IF PEEK (Textpointer+q-1)=Newline T
HEN  GO SUB 9800: GO TO 9110
9250 IF PEEK (Textpointer+q-1)=Quote THE
N  GO SUB 9800: GO TO 9160
9260 LET q=q+1
9270 GO TO 9240
9280 REM Copy 5-byte binary number
9290 IF Byte=Number THEN  LET q=5: GO SU
B 9800: GO TO 9160
9310 REM First character of name must be
 upper or lower case letter
9320 IF Byte>=CODE ("A") AND Byte<=CODE
("Z") THEN  LET c$=CHR$ (Byte): GO TO 93
70
9330 REM Use upper case instead of lower
 case
9340 IF Byte>=CODE ("a") AND Byte<=CODE
("z") THEN  LET c$=CHR$ (Byte-32): GO TO
 9370
9360 GO TO 9160
9370 LET n$=""
9380 LET n$=n$+c$
9400 REM Letter, digit or underscore aft
er first character of name
9410 IF PEEK (Textpointer)>=CODE ("A") A
ND PEEK (Textpointer)<=CODE ("Z") THEN
LET c$=CHR$ (PEEK (Textpointer)): LET Te
xtpointer=Textpointer+1: GO TO 9380
9420 REM Use upper case instead of lower
 case
9430 IF PEEK (Textpointer)>=CODE ("a") A
ND PEEK (Textpointer)<=CODE ("z") THEN
LET c$=CHR$ (PEEK (Textpointer)-32): LET
 Textpointer=Textpointer+1: GO TO 9380
9440 IF PEEK (Textpointer)>=CODE ("0") A
ND PEEK (Textpointer)<=CODE ("9") THEN
LET c$=CHR$ (PEEK (Textpointer)): LET Te
xtpointer=Textpointer+1: GO TO 9380
9450 IF PEEK (Textpointer)=Underscore TH
EN  LET c$=CHR$ (PEEK (Textpointer)): LE
T Textpointer=Textpointer+1: GO TO 9380
9460 REM End with $ for string variable
9470 IF PEEK (Textpointer)=CODE ("$") TH
EN  LET n$=n$+"$": LET Textpointer=Textp
ointer+1: GO TO 9500
9480 REM ( if array or function
9490 IF PEEK (Textpointer)=CODE ("(") TH
EN  LET n$=n$+CHR$ (PEEK (Textpointer)):
 LET Textpointer=Textpointer+1
```