program area after it has been altered, and we will need a machine code program to do this for us.

The Spectrum manual gives a great deal of information about the way a BASIC program is stored and what the various areas of memory are used for. However, because of the large number of different sections in the working area, and the way these areas can move around, it is difficult to write utility programs without using machine code subroutines from the ROM. If you want to do any serious utility programming on the Spectrum, a valuable reference work is *The Complete Spectrum ROM Disassembly*, by Dr Ian Logan and Dr Frank O'Hara. This explains how all the ROM routines work.

Two of the most important subroutines in the ROM for use in utility programs are the routines that open up or reclaim space in the working area, and we will be looking at these when we come to the variable search and replace program.

Experimenting With BASIC

You can try altering the contents of a program during execution but you should save the program first, as a system crash is a common result. Use the Monitor program (see page 118), which allows you to inspect and alter the contents of memory. This can inspect and alter itself under your command. Insert some extra REM lines at the start of the program and try these suggestions on them first:

- Find the start of BASIC text area (see page 58), and inspect the Monitor program in memory until you can identify program lines.
- Change the values of the bytes after a REM token, then quit the program and list the altered line.
- Try putting a value greater than 127 into a REM line — again, quit and list: you may be surprised.
- Alter the line number bytes of a line — this produces unpredictable results, especially if the new number is out of sequence with its neighbours.
- You can alter the line length bytes — try decreasing the indicated length first — but you should insert a new end-of-line marker at the indicated byte.
- On the Commodore 64 you can change the link address bytes: try replacing one line's link address by the link address of the succeeding line, and then list the program.
- If you're feeling more ambitious, consult your manual and explore the variables' storage area. This usually begins in the memory map where the BASIC text area ends. There are up to six different variable types, each with its own storage format: numeric variables, numeric arrays, integer variables, integer arrays, string variables and string arrays. String and integer variable formats are the simplest, being essentially straightforward representations of the data and the variable name; numeric array data is the most complicated.
- You can try changing token values in program lines: this will change the command word. If you do this via the Monitor program to a line that it is currently executing, you will be introducing a potentially massive paradox into the interpreter

ERRATA

On page 118 in the BBC and Commodore BASIC flavours:

- In line 1150 there are two successive assignments to CS(3) — change the second assignment to
  CS(4)="Q"
- In line 6600 change Z=1 to Z=2
- Line 200 of the BBC flavours should be
  200 CLS

as in the Spectrum version

How BASIC Programs Are Stored

Most micros follow essentially the same storage format in the BASIC program area. Each program line begins with the line data — BASIC line number in two-byte form, and some information about the length of the line. Program text is stored more or less unaltered, though BASIC keywords are replaced by one-byte code numbers called 'tokens'

500 REM • • EXAMPLE • •

BBC VERSION

13	HI 1	LO 244	17	32	REM	*	*	E	X	A	M	P	L	E	*	*
START LINE MARKER	LINE NUMBER		LINE LENGTH		TOKEN	PROGRAM TEXT										

COMMODORE 64 VERSION

LO 18	HI 8	LO 244	HI 1	REM	*	*	E	X	A	M	P	L	E	*	*	0
LINK ADDRESS		LINE NUMBER		TOKEN	PROGRAM TEXT										END LINE MARKER	

SPECTRUM VERSION

HI 1	LO 244	LO 13	HI 0	REM	*	*	E	X	A	M	P	L	E	*	*	13
LINE NUMBER		LINE LENGTH		TOKEN	PROGRAM TEXT										END LINE MARKER	

In this format 13 (ASCII for [RETURN]) is a start-of-line marker; it is more commonly placed at the end of the line. Line length has only a single byte, which confines program lines to 255 characters. The space directly after the line number has been stored

The Link Address bytes contain the two-byte address of the first byte of the next program line. The BASIC program text area starts at address 2049, and this line is 17 bytes long, so the start address of the next line is 2066

Spectrum methods are always interesting: the line length occupies two bytes, so a single program line could be 65,535 characters long! The line length here is 13 — the bytes in the line including the end-of-line byte, but not counting the line data bytes

LIZ DIXON