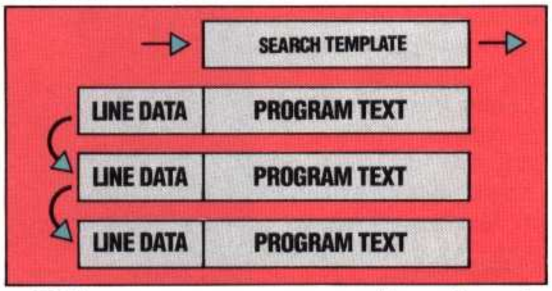
ROLL CALL

Programs that are built in to home computers and provide editing facilities or other programming aids are known as 'utilities'. We begin a series of articles that explore a range of utility programs for the more popular home computers and develop our own BASIC utilities.

The range of utilities featured on home micros varies considerably: some computers have only a simple editor, as in the Sinclair Spectrum, while other machines feature more extensive facilities. The BBC Micro, for example, includes the TRACE and RENUMBer commands: the former causes the line number of each BASIC statement to be displayed as the line is being executed, and the latter automatically renumbers the lines of a BASIC program. Both of these facilities are immensely useful in program development and debugging. But whatever is provided with your machine, it is invariably helpful to have additional utilities, and there is a wide range of commercially available programs to choose from.



Searching Hi And Lo
On the BBC Micro and the Spectrum a program line begins with three or four bytes dedicated to the line number and the length of the line. This is followed by the tokenised BASIC text. When the program encounters the line it records the line number and calculates the start address of the next line from the length of the current line. It then 'slides' the search template along the program text until it meets end-of-line, end-of-program, a REM token or a successful match

Utility programs are generally written in Assembly language, partly because of the speed of machine code and partly because it is not easy for a BASIC program to alter itself without crashing the computer. However, we will begin by looking at some simple utilities that can be writen in BASIC. In this way, we can concentrate our attention on what a utility program has to do, without having to consider other complicating details, such as the role of the computer's operating system and the BASIC interpreter.

Although it is difficult for a BASIC program to alter itself, there is no problem in creating a BASIC program that inspects another BASIC program. The utility program we give here, in versions for the Spectrum and the BBC Micro, searches through a BASIC program for the name of a variable or function, and prints the line numbers where the name is found.

Both programs begin by finding where the program text starts in the computer's memory. Then they go through the program line by line, skipping over those sections that cannot include a name and extracting all the names. The final step involves comparing each extracted name with the name the program was asked to find.

When the program starts its search of a new line of BASIC text, it first notes the line number, which in both cases is stored in two bytes, and the line's length (the number of bytes it occupies). In the BBC Micro, the line length is contained in one byte, and is the total number of bytes in the line from the line number to the end-of-line marker (ASCII code 13). In the Spectrum, the line length is stored in two bytes, and represents the number of bytes from the character following the length bytes to the end-of-line marker (thus not including the line number and line length bytes in its total).

In both versions of the program, we ignore all the REM statements and anything that is enclosed in quotes, as we will not normally have any program variables in these character strings. The BBC Micro allows you to include hexadecimal numbers in a program, prefixed by the character &. We need to make sure that our program does not mistake these hexadecimal numbers for variable names, and therefore we need to make the program skip over any strings preceded by an &. For example, we do not want our program to mistake the hex number A0 in &A0 as the variable name 'A0'.

In the Spectrum, numbers are stored in a program as the ASCII characters for the digits of the number, followed by the ASCII code byte 14, and then five bytes containing the binary equivalent of the number. Our program needs to be able to skip over the number code and the five byte binary equivalent.

Having tested for these conditions, the program proceeds to inspect the current line for any names. In both programs, a name is defined as beginning with a letter, followed by another letter or a digit. The BBC Micro version allows integer variables (distinguished by a % character after the name) and the underscore character, and both versions allow string variables, which are followed by a $ character.

The name of an array, a function or (in the BBC Micro) a procedure will be followed by an open bracket — (. Strictly speaking, this is not part of the name, but is used by the programs to distinguish these from simple variables.

There are further complications in the Spectrum program. In particular, Spectrum BASIC does not distinguish between upper- and lower-case for characters in a variable name. Thus, FRED, Fred or FRed are all treated as the same variable name. The Spectrum program, therefore, converts