



```

POSN=0
9910 FOR K=1 TO SL
9920 IF Y$(K TO K+ZL-1)=Z$ THEN LET POSN=K:LET
K=SL
9930 NEXT K:RETURN

```

Notice in this subroutine that the string variable Y\$ is treated as a subscripted array-type variable, even though it has not been DIMensioned. Since in Sinclair BASIC all string variables are array-type variables, a string variable that is not DIMensioned is implicitly a variable-length single-dimension array of single characters; if it is DIMensioned, its element length is fixed by the last number in the DIM statement. Whereas in other BASICS DIM X\$(8,7) creates a two-dimension array, in Sinclair BASIC it creates a single-dimension array of eight elements, each of them fixed in length at seven characters.

The strict attention paid to the length of DIMensioned string variables by Sinclair BASIC means that seemingly simple statements can have differing effects, depending upon whether a DIM statement has been executed or not. If a\$ is a simple string variable, then LET a\$="" makes the contents of a\$ equal to the null string (") and the length of a\$ equal to zero. If DIM a\$(7) has been executed previously, however, then LET a\$="" makes the contents of a\$ equal to seven spaces, and the length of a\$ equal to seven (which it will always be, following the DIM statement). Furthermore, in such a case, even though LET a\$="" has been executed, a test such as:

```
IF a$="" THEN PRINT "null-string"
```

will fail, and nothing will be printed — a\$ is equal to seven spaces, not the null-string.

If you need to test string array elements in this way, then it's probably best to set aside a string variable for the purpose, DIMension it to the length of the longest array variable used in the program, and test your array variables against it, like this:

```

100 DIM a$(12,34)
120 DIM b$(7,56)
140 DIM n$(56)
150 REM n$ will be used as the empty string
.....

```

```

580 IF b$(3)=n$( TO 56) THEN PRINT "empty"
590 IF a$(11)=n$( TO 34) THEN PRINT "empty"
.....

```

Here n\$ is used only as the empty string, and if it wasn't used in this way then the tests in lines 580-590 would have to use literals, thus:

```

580 IF b$(3)=" " THEN PRINT "empty"
585 REM 56 spaces between the quotes

```

This is inconvenient and prone to error. An alternative to using n\$ in this way is to DIMension all array variables with one more element than they need, and use that last element as an empty string for tests of that array, so that line 590 might be:

```
590 IF a$(11)=a$(12) THEN PRINT "empty"
```



IAN MCKINNELL

SuperBASIC

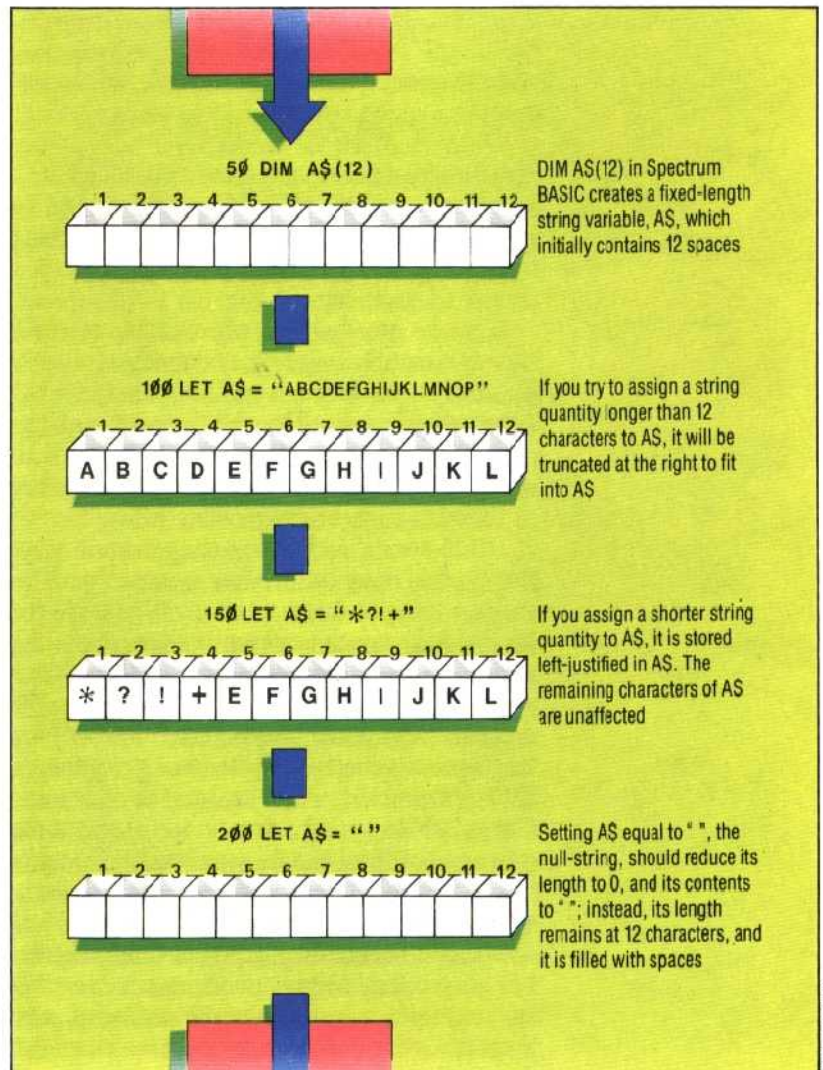
While Sinclair's SuperBASIC has a considerably enhanced range of commands over ZX BASIC, the most significant feature is its abandonment of the single-key reserved word entry system common to the ZX80, ZX81 and Spectrum. This was originally introduced as an economy measure for users (it was felt that pressing a single key rather than typing a whole word would prove attractive). The system dictated that a variety of different 'modes' would be necessary to allow the entry of single characters to be differentiated from the entry of key words. This system was attractive to Sinclair users who had never previously encountered a keyboard, but for those who had used a typewriter it proved to be a source of frustration

assuming that a\$(12) is never used and so contains only spaces.

Notice lastly that, in Sinclair BASIC, the first element in any array has the subscript one, whereas in some other BASICS the first element in an array has subscript zero. In the next instalment of the course, we will conclude this look at Spectrum BASIC.

Procrustean Strings

The mythical Greek character Procrustes was an innkeeper who kept only one size of bed, and stretched or truncated his guests to fit it



KEVIN JONES