

BBC BASIC Operators

The following operators may be used as part of expressions.

Arithmetic operators

The arithmetic operators take two numbers as operands.

+	add	-	subtract
*	multiply	/	divide
^	raise to the power		

Relational operators

The relational operators can compare numbers, and return a logical value of FALSE=0 and TRUE=-1.

<	less than	=	equal to
<=	less than or equal to	>	greater than
<>	not equal to	>=	greater than or equal to

Logical operators

The logical operators operate on numeric or logical values. Boolean FALSE is taken as zero, TRUE as -1.

Indirection operators

The following operators allow the programmer to access memory directly (called PEEK in other versions of BASIC) or to alter the contents of specified memory locations (called POKE in other BASICs).

These operators should only be used by experienced programmers, since they give the programmer the capability of altering the operation of other activities in the Cambridge Z88 operating system; in general, only addresses within the region allocated by a BASIC DIM statement should be altered.

?	byte indirection	!	word indirection
\$	string indirection		

The usual way of using these operators is to allocate a vector using the DIM statement, and then manipulate elements of this vector with the indirection operators. For example,

```
DIM vec 15
```

allocates a vector of 16 bytes, and assigns its address to the variable `vec`. These bytes can be addressed as

0	1	2	3	4	5	6	7	8	9	10	11	12
vec?0	vec?1	vec?2	vec?3	vec?4	vec?5	vec?6	vec?7	vec?8	vec?9	vec?10	vec?11	vec?12

Note that `?vec` is an alternative way of writing `vec?0`. A typical manipulation might be

```
vec?1 = vec?1 + 1
```

Alternatively the vector could be addressed as 4 four-byte words:

0	1	2	3	4	5	6	7	8	9	10	11	12
vec?0	vec?4	vec?8	vec?12									

Finally, the string indirection operator allows a string to be stored directly into an area of memory. For example:

```
$vec = "THE COMPUTER"
```

stores characters in the vector as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12
T	H	E	&20	C	O	M	P	U	T	E	R	&0D
\$vec												

The string can be accessed with the `$` operator, so the following statement will print the string:

```
PRINT $vec
```

Individual characters can be accessed with the `?` operator.

Order of precedence

The order of precedence of the operators is as follows:

Group 1	-	unary minus
	()	brackets
	NOT	logical NOT
Group 2	^	raise to the power
Group 3	*	multiply
	/	divide
Group 4	+	add
	-	subtract
Group 5	<	less than
	=	equal to
	<=	less than or equal to
	>	greater than
	<>	not equal to
	>=	greater than or equal to
Group 6	AND	Logical AND
Group 7	OR	Logical OR
	EOR	Logical Exclusive-OR

so, for example, $A-B*C^D$ is equivalent to $A-(B*(C^D))$.