

Generally though:

WRITE(P1,P2,.....Pn); is equivalent to:

BEGIN WRITE(P1); WRITE(P2);; WRITE(Pn) END;

The write parameters P1,P2,.....Pn can have one of the following forms:

<e> or <e:m> or <e:im:n> or <e:im:H>

where e, m and n are expressions and H is a literal constant.

We have 5 cases to examine:

1) e is of type integer; and either <e> or <e:m> is used.

The value of the integer expression e is converted to a character string with a trailing space. The length of the string can be increased (with leading spaces) by the use of m which specifies the total number of characters to be output. If m is not sufficient for e to be written or m is not present then e is written out in full, with a trailing space, and m is ignored. Note that, if m is specified to be the length of e without the trailing space then no trailing space will be output.

2) e is of type integer and the form <e:m:H> is used.

In this case e is output in hexadecimal. If m=1 or m=2 then the value (e MOD 16^m) is output in a width of exactly m characters. If m=3 or m=4 then the full value of e is output in hexadecimal in a width of 4 characters. If m>4 then leading spaces are inserted before the full hexadecimal value of e as necessary. Leading zeroes will be inserted where applicable. Examples:

```
WRITE(1025;m:H);
```

```
m=1  outputs: 1
m=2  outputs: 01
m=3  outputs: 0401
m=4  outputs: 0401
m=5  outputs: _0401
```

3) e is of type real. The forms <e>, <e:m> or <e:im:n> may be used.

The value of e is converted to a character string representing a real number. The format of the representation is determined by n.