2.3.1.2 WRITELN

WRITELN outputs gives a newline. This is equivalent to WRITE(CHR(13)). Note that a linefeed is included.

WRITELN(P1,P2,.....P3); is equivalent to:

BEGIN WRITE(P1,P2,.....P3); WRITELN END;

2.3.1.3 PAGE

The procedure PAGE is equivalent to WRITE(CHR(12)); and causes the video screen to be cleared or the printer to advance to the top of a new page.

2.3.1.4 READ

The procedure READ is used to access data from the keyboard. It does this through a buffer held within the runtimes - this buffer is initially empty (except for an end-of-line marker). We can consider that any accesses to this buffer take place through a text window over the buffer through which we can see one character at a time. If this text window is positioned over an end-of-line marker then before the read operation is terminated a new line of text will be read into the buffer from the keyboard. While reading in this line all the various control codes detailed in Section 0.0 will be recognised. Now:

READ(V1,V2,.....Vn); is equivalent to:

BEGIN READ(V1); READ(V2); .....; READ(Vn) END;

where V1, V2 etc. may be of type character, string, integer or real.

The statement READ(V); has different effects depending on the type of V. There are 4 cases to consider:

1) V is of type character.

In this case READ(V) simply reads a character from the input buffer and assigns it to V. If the text window on the buffer is positioned on a line marker (a CHR(13) character) then the function EOLN will return the value TRUE and a new line of text is read in from the keyboard. When a read operation is subsequently performed then the text window will be positioned at the start of the new line.

Important note: Note that EOLN is TRUE at the start of the program. This