**Formula for Screen Location Calculation**

To calculate the offset in video memory for a given position (row, column):

**Offset = (row \* number of columns + column) \* 2**

**Offset = (y \* 80 + x) \* 2**

* **row**: The row number (0-based index, ranging from 0 to 24 in an 80x25 screen).
* **column**: The column number (0-based index, ranging from 0 to 79 in an 80-column screen).
* **number of columns**: This is typically 80 in standard text mode.
* **2**: Each character cell takes up 2 bytes (1 byte for the character, 1 byte for the color attribute).

**Example**

To find the offset for a character at row 10, column 20 or (x, y) = (20, 10) on an 80x25 screen:

1. **Substitute the values**:

**Offset = (10 \* 80 + 20) \* 2**

So, the offset in video memory for row 10, column 20 would be **1640** (or 0x668 in hexadecimal).

**Putting It in Context**

In assembly, you could load this offset into DI or SI for operations with the video segment 0xB800. For example:

mov ax, 0xB800 ; Load video segment

mov es, ax ; Set extra segment to video memory

mov di, 1640 ; Offset calculated for row 10, column 20

mov al, 'A' ; Character to display

mov ah, 0x0F ; Color attribute (white on black, for example)

mov [es:di], ax ; Place character with color at desired location