**2D Array**

There are two types of implementation of array in memory

1. Row Major Implementation
2. Column major implementation

The whole 2d array in memory is inserted into memory in single dimension

**Row Major Implementation**

In row major implementation the first row is inserted into memory then after that 2nd row is inserted into memory and so on.

Eg.

Matrix:

|  |  |  |
| --- | --- | --- |
| 16 | 106 | 76 |
| 10 | 32 | 98 |

Memory allocation of the above matrix in row major implementation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 16 | 106 | 76 | 10 | 32 | 98 |

100 104 108 112 116 120 - address of each data into memory

How to compute the address of memory in row major

a[i][j] = B + ((i-idx)\*n + (j-idx))\*sizeof(data\_type) B – being base address

i – number of row, j – number of column,

n- total number of columns,

idx – starting index

**Column Major Implementation**

In the column major implementation, first we insert the 1st column, then 2nd column after that 3rd column and so on.

Eg. Matrix

|  |  |  |
| --- | --- | --- |
| 16 | 106 | 76 |
| 10 | 32 | 98 |

Memory allocation of the above matrix in column major implementation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 16 | 10 | 106 | 32 | 76 | 98 |

100 104 108 112 116 120

How to compute the address of memory in column major

a[i][j] = B + ( (j-idx)\*m + (i-idx) )\*sizeof(data\_type) m – being total number of rows