Representation of 1D-Array in memory

Let A be a 1D array. Elements of A are stored in successive memory locations. The address of the first element of the array is known as **Base Address** and is denoted by **base(A).**

The address of the k^{th} element of the array A is denoted by **Loc(A[k])**.

Hence, **Loc(A[k]) = base(A) + w * (k – lb)**

where **w** is size of each element of the array in byte and **lb** is the lower bound of the array.

Example:

The linear array A shown below can be represented as either A[1:6] or A(1..6) to mean that array A has 6 homogeneous elements with lower bound 1 and upper bound 6.

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
100	102	104	106	108	110

If we assume base(A) = 100 and each element A contains integer values and if we require 2 bytes to store integer data then w = 2.

Hence Loc(A[3]) = base(A) + 2 * (3 – 1) = 100 + 4 = 104.

Representation of 2D Array in memory

An 2D-array with m rows and n columns is denoted as **either** A[1**:**m, 1**:**n] **or** A[1**..**m, 1**..**n]. In the memory, a 2D-array of order **m x n** is stored as 1D-array having (**m * n**) elements.

Now the elements can be stored in two ways –

1. **Column Major Order** – Elements are stored column by column.

2. **Row Major Order** – Elements are stored row by row.

Example:

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}_{3 \times 4}$$

Column Major Order

A[1, 1]	A[2, 1]	A[3, 1]	A[1, 2]	A[2, 2]	A[3, 2]	A[1, 3]	A[2, 3]	A[3, 3]	A[1, 4]	A[2,4]	A[3,4]
1	5	9	2	6	10	3	7	11	4	8	12

100

Row Major Order

A[1, 1]	A[1, 2]	A[1, 3]	A[1, 4]	A[2, 1]	A[2, 2]	A[2, 3]	A[2, 4]	A[3, 1]	A[3, 2]	A[3,3]	A[3,4]
1	2	3	4	5	6	7	8	9	10	11	12

100

Location of an element in the i^{th} row and j^{th} column is represented as Loc(A[i, j]).

Column Major Order:

Loc(A[i, j]) = base(A) + w * [m * (j – lbc) + (i – lbr)].

Row Major Order:

Loc(A[i, j]) = base(A) + w * [n * (i – lbr) + (j – lbc)].

Where lbr – lower bound of row
lbc – lower bound of column.
m – number of rows.
n – number of columns.
w – size of each element in bytes.

Check:

Find address of element A[2, 3] of the above 2D-array A in both methods of storage assuming base address is 100 and size of each element is 2 bytes.

Row Major Order

Loc(A[2, 3]) = 100 + 2 * [4 * (2 – 1) + (3 – 1)] = 100 + 12 = 112.

Column Major Order

Loc(A[2, 3]) = 100 + 2 * [3 * (3 – 1) + (2 – 1)] = 100 + 14 = 114.

TRY : Find address of A[3,4] in both cases. Ans = 122.

Problem 1:

Let the size of the elements stored in an 8 x 3 matrix be 4 bytes each. If the base address of the matrix is 3500 then find the address of A[4, 2] for both row major and column major cases.

Solution:

Location of an element in the i^{th} row and j^{th} column of matrix A is represented as Loc(A[i, j]).

Column Major Order –

Location of an element in the i^{th} row and j^{th} column of matrix A is

Loc(A[i, j]) = base(A) + w * [m * (j – lbc) + (i – lbr)].

where lbr – lower bound of row
lbc – lower bound of column.
m – number of rows.
w – size of each element in bytes.

So, in column major order,

address of A[4, 2] = 3500 + 4 * [8 * (2 – 1) + (4 – 1)] = 3500 + 4 * 11 = 3500 + 44 = 3544.

Row Major Order –

Loc(A[i, j]) = base(A) + w * [n * (i – lbr) + (j – lbc)].

where lbr – lower bound of row
lbc – lower bound of column.
n – number of columns.
w – size of each element in bytes.

So, in row major order,

address of A[4, 2] = 3500 + 4 * [3 * (4 – 1) + (2 – 1)] = 3500 + 4 * 10 = 3500 + 40 = 3540.

Problem 2:
Consider the array int a [1.. 10] [1..10] and the base address 2000, then calculate the address of the array a[2] [3] in the row and column major ordering.
Solution :

Let us assume, 2 bytes are required to store each integer element in the array.

Column Major Order –

We know, Location of an element in the i^{th} row and j^{th} column of matrix A is

Loc(A[i, j]) = base(A) + w * [m * (j – lbc) + (i – lbr)].

where lbr – lower bound of row
lbc – lower bound of column.
m – number of rows.
w – size of each element in bytes.

So, in **Column Major Order**,

Address of the element a[2][3] = 2000 + 2 * (10 * (3 – 1) + (2 – 1)) = 2000 + 42 = 2042

Row Major Order –

We know Loc(A[i, j]) = base(A) + w * [n * (i – lbr) + (j – lbc)].

where lbr – lower bound of row
lbc – lower bound of column.
n – number of columns.
w – size of each element in bytes.

So, in **row Major Order**,

Address of the element a[2][3] = 2000 + 2 * (10 * (2 – 1) + (3 – 1)) = 2000 + 24 = 2024.

Problem 3:
Suppose one 2-D array is initialized as int a[5][7]; Base address is 4000.Find the location of element a[2][4] in row major form and column major form.