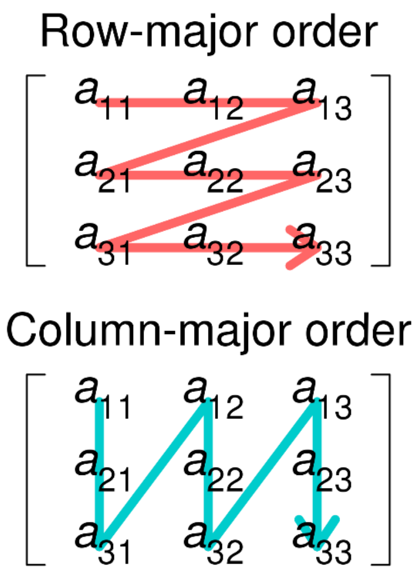
Row Major and Column Major order



Address Calculation in 1D Array:

Memory address calculation Address of an element of an array say A[I] is calculated using the following formula:

Address of A [I] = B + W * (I- LB)

Where,

B = Base address

W = Storage Size of one element stored in the array (in byte)

I = Subscript of element whose address is to be found

LB = Lower limit / Lower Bound of subscript, if not specified assume 0 (zero)

Example:

Given the base address of an array B[1300..1900] as 1020 and size of each element is 2 bytes in the memory. Find the address of B[1700].

Solution:

The given values are: B = 1020, LB = 1300, W = 2, I = 1700

Address of A [I] = B + W * (I -LB)

= 1020 + 2 * (1700-1300)

= 1020 + 2 * 400

= 1020 + 800

= 1820

Address calculation in 2-D array: suppose A is a 2D array with dimensions M1 X M2. So that here total number of elements is equal to M1 X M2. Address should be calculated using one of the two methods as the following.

a. **Row-by-Row**
Location (A [j, k]) = Base (A) + w [N (j-1) + (k -1)]
Here, **j** and **k** indicates index value of the element whose address you want to search. **Base (A)** is the starting address of the given array. **W** is the memory word required to store given array's element. Here, array is the of size **M(rows) X N(columns).**

b. **Column-by-Column**
Location (A [j, k]) = Base (A) + w [M (k-1) + (j -1)]
Here, **j** and **k** indicates index value of the element whose address you want to search. **Base (A)** is the starting address of the given array. **W** is the memory word required to store given array's element. Here, array is the of size **M(rows) X N(columns).**

Examples:

1. Consider 30 X 4 2D arrays and base address is 200 and 1 word per memory locations. Find out the address of A (15, 3).
Base (A), M =30, N=4, j=15, k=3, w=1

a. Row-by-row
Location (A [j, k]) = Base (A) + w [N (j-1) + (k -1)]
Location (A [15, 3]) = 200 + 1 [4 (15 - 1) + (3 - 1)]
= 200 + 1 [56 + 2]
= 200 + 58
=258

b. Column-by-column
Location (A [j, k]) = Base (A) + w [M (k-1) + (j -1)]
Location (A [15, 3]) = 200 + 1 [30 (3 – 1) + (15 – 1)]
= 200 + 1 [30 (2) + 14]
= 200 + 74
=274

Address calculation in multidimensional array: Suppose A is a 3D array of size M1 X M2 X M3. So total number if elements are M1 X M2 X M3. In 3D array lower bound (LB) is not always zero (0). It can be any number. Base address of any row of matrix is called effective address.

A [2:11, 3:15] indicates that A is a 2D array with lower bound (LB) of row is 2 and Upper Bound (UB) of row is 11. In the same way lower bound (LB) of column is 3 and upper bound (UB) of column is 15.

Length of each dimension is equal to L_i = UB – LB +1

Lower bound (LB) is not always started with zero (0). So, we have to calculate effective address of each value.

Effective Address (E_i) = K_i - LB

Real address of any element is calculated as follows, Row-by-Row: **Base (A) + W [(E_1L_2 +E_2) L_3 + E_3]**

Column-by-column: **Base (A) + W [(E_3L_2 + E_2) L_1 + E_1)**

Example: Suppose multidimensional array A and B are declared using **A [-2:2, 2:22] and B [1:8, -5:5, -10:5].**

1. Find out the length of each dimension and number of elements in array A and B.
2. Consider B [3, 3, 3] elements in array B find effective address E1, E2 and E3. Also find out the real address of these elements. (In this case the base address of array B is 400 and W is 4)

Answer: For array A

L_1 = UB –LB +1
=2 – (-2) +1
= 4 + 1 = 5

L_2 = UB –LB +1
=22 – 2 + 1
= 21

For array B

L_1 = UB –LB +1
= 8 – 1 + 1
= 8

L_2= UB –LB +1
= 5 – (-5) +1
= 11

L_3= UB –LB +1
=5 – (-10) +1
= 16

Total number of elements in array B is: L_1 X L_2 X L_3 = 8 X 11 X 16 = 1408.

Here in given question K_i = 3, 3, 3 therefore K_1=3, k_2 = 3 and K_3 = 3

LB_1= 1, LB_2 = -5 LB_3 = -10
$E_1 = K_1 – LB_1$ = 3 – 1 = 2
$E_2 = K_2 – LB_2$ = 3- (-5) = 8
$E_3 = K_3 – LB_3$ = 3 – (-10) = 13

The real address Row – by- Row:
= Base (A) + W [(E_1L_2 + E_2) L_3 + E_3]
= 400 + 4 [(2 (11) + 8) 16 + 13]
= 400 + 4 [(22+8) 16 + 13]
= 400 + 4 [(30) 16 + 13]
= 400 + 4 [480 + 13]
= 400 + 4 [493]
= 400 + 1972
= 2372

The real address Column – by – column:
= Base (A) + W [(E_3L_2 +E_2) L_1 + E_1]
= 400 + 4 [(13(11) + 8) 8 + 2]
= 400 +4 [(143 + 8) 8 + 2]
= 400 + 4[(151) 8 + 2)]
= 400 + 4[1208 + 2]
= 400 + 4840
= 5240

Let **C** be an n-dimensional array.

The length L***i*** of dimension ***i*** of **C** can be calculated by,

Li=upper bound - lower bound +1

For a given subscript **K*i*** , the effective index E***i*** of L***i*** can be calculated from,

Ei=Ki - lower bound

Then address C[K1,K2,...,Kn] of element of **C** is, *Base* (C) +w[(((.....($E_nL_{n-1}+E_{n-1})L_{n-2})+....+E_3)L_2+E_2)L_1+E_1$] , when C is stored in column major, and

Base (C)+w[(...(($E_1L_2+E_2)L_3+E_3)L_4+...+E_{n-1})L_n+E_n$] , when C is stored in row major order.

Base (C) denotes the address of 1st element of C and w denotes the number of words per memory location.

Compute the location of A[10]10] if the base address of A[-15:20, 10:35] is given as 3000 and occupying 4 bytes of memory. Assume row major storage.