Array

Address computation

1D array – address calculation

- Let A be a one dimensional array.
- Formula to compute the address of the I^{th} element of an array (A[I]) is:

Address of A[I] = B + W * (I – LB)

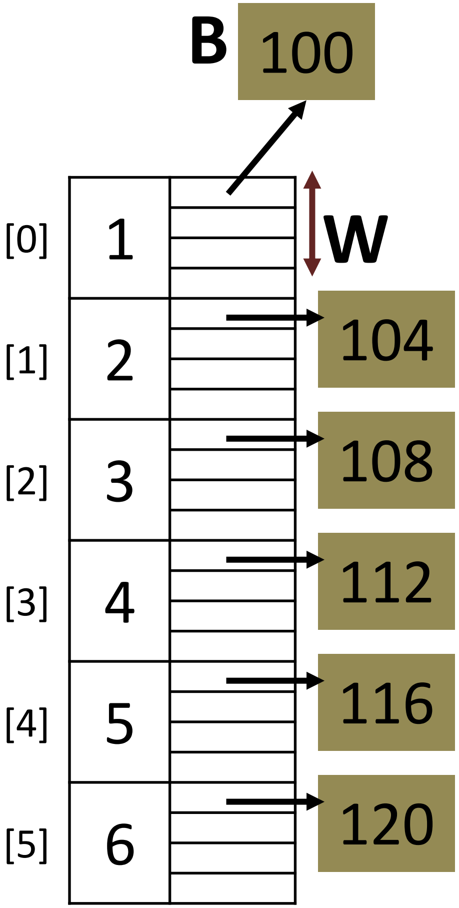
where,

B = Base address/address of first element, i.e. A[LB].

W = Number of bytes used to store a single array element.

I = Subscript of element whose address is to be found.

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



Example

- Similarly, for a character array where a single character uses 1 byte of storage.
- If the base address is 1200 then,

Address of A[I] = B + W * (I – LB)

Address of A[0] = 1200 + 1 * (0 – 0) = 1200

Address of A[1] = 1200 + 1 * (1 – 0) = 1201

…

Address of A[10] = 1200 + 1 * (10 – 0) = 1210

Example

- If **LB** = 5, **Loc(A[LB])** = 1200, and **W** = 4.
- Find **Loc(A[8])**.

Address of A[I] = B + W * (I – LB)

Loc(A[8]) = Loc(A[5]) + 4 * (8 – 5)

= 1200 + 4 * 3

= 1200 + 12

= 1212

Example

- Base address of an array **B[1300.....1900]** is **1020** and size of each element is 2 bytes in the memory. Find the address of **B[1700]**.

Address of A[I] = B + W * (I – LB)

- Given: **B** = 1020, **LB** = 1300, **W** = 2, **I** = 1700

Address of B[1700] = 1020 + 2 * (1700 – 1300)
= 1020 + 2 * 400
= 1020 + 800
= 1820

	[0]	[1]	[2]	[3]
[0]	1	2	3	4
[1]	5	6	7	8
[2]	9	10	11	12

Row-major

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

Column-major

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

2D array – address calculation

- If **A** be a two dimensional array with **M** rows and **N** columns. We can compute the address of an element at **I**th row and **J**th column of an array (**A[I][J]**).

B = Base address/address of first element, i.e. A[LBR][LBC]
I = Row subscript of element whose address is to be found
J = Column subscript of element whose address is to be found
W = Number of bytes used to store a single array element
LBR = Lower limit of row/start row index of matrix, if not given assume 0
LBC = Lower limit of column/start column index of matrix, if not given assume 0
N = Number of column of the given matrix
M = Number of row of the given matrix

Contd…

- Row Major

Address of A[I][J] = B + W * (N * (I – LBR) + (J – LBC))

- Column Major

Address of A [I][J] = B + W * ((I – LBR) + M * (J – LBC))

- Note: **A[LBR…UBR, LBC…UBC]**

M = (UBR – LBR) + 1

N = (UBC – LBC) + 1

Example

- Suppose elements of array **A[4][5]** occupies **4** bytes, and the address of the first element is **49**. Find the address of the element **A(4,3)** when the storage is row major.

Address of A[I][J] = B + W * (N * (I – LBR) + (J – LBC))

- Given: **B** = 49, **W** = 4, **M** = 4, **N** = 5, **I** = 4, **J** = 3, **LBR** = 0, **LBC** = 0.

Address of A[4][3] = 49 + 4 * (5 * (4 – 0) + (3 - 0))

= 49 + 4 * (23)

= 49 + 92

= 141

Example

- An array **X [-15…10, 15…40]** requires **one** byte of storage. If beginning location is **1500** determine the location of **X [15][20]** in column major.

Address of A[I][J] = B + W * [(I – Lr) + M * (J – Lc)]

- Number or rows (**M**) = **(UBR – LBR) + 1** = [10 – (- 15)] +1 = 26
- Given: **B** = 1500, **W** = 1, **I** = 15, **J** = 20, **LBR** = -15, **LCR** = 15, **M** = 26

Address of X[15][20] = 1500 + 1 * [(15 – (-15)) + 26 * (20 – 15)]

= 1500 + 1 * [30 + 26 * 5]

= 1500 + 1 * [160]

= 1660