**Row-major and column-major order**

**1-D array**

Address of A[I] = B + W \* (I – LB)

I = address to be found,

B = Base address,

W = array(in byte)

LB = Lower Bound of subscript(If not specified assume zero).

Example:

A[1300…………1900] as 1020 and the size of each element is 2 bytes in the memory, find the address of A[1700]?

B = 1020

LB = 1300

W = 2 Byte

I = 1700

Address of A[I] = B + W \* (I – LB)

Address of A[1700] = 1020 + 2 \* (1700 – 1300)

= 1020 + 2 \* (400)

= 1020 + 800

Address of A[1700] = 1820

**2-D array**

**Row-major order**

Address of A[I][J] = B + W \* ((I – LR) \* N + (J – LC))

I = Row address to be found,

J = Column address to be found,

B = Base address

W = array(in byte)

LR = Lower Limit of row (If not given assume it as zero),

LC = Lower Limit of column (If not given assume it as zero),

N = Number of column given in the matrix.

Example:

Given an array, arr[1………10][2………15] with base value 100 and the size of each element is 1 Byte in memory. Find the address of arr[8][6] with the help of row-major order?

B = 100

W = 1 Bytes

I = 8

J = 6

LR = 1

Number of column given in the matrix N = Upper Bound – Lower Bound + 1

= 15 – 1 + 2

= 16

Address of A[I][J] = B + W \* ((I – LR) \* N + (J – LC))

Address of A[8][6] = 100 + 1 \* ((8 – 1) \* 16 + (6 – 2))

Address of A[I][J] = 216

**column-major order**

B = 100

W = 1 Bytes

I = 8

J = 6

LR = 1

Number of column given in the matrix M = Upper Bound – Lower Bound + 1

= 10 – 1 + 1

= 10

Address of A[I][J] = B + W \* ((J – LC) \* M + (I – LR))

Address of A[8][6] = 100 + 1 \* ((6 – 2) \* 10 + (8 – 1))

Address of A[I][J] = 147

**3D-ARRAY**

**Row Major Order**

Address of [i][j][k] = B + W \* {[(I – LR) \* N] + [(J – LC)]\* R + [K – LB]}

I = Row address to be found,

J = Column address to be found,

K = Block address to be found,

B = Base address

W = Storage size of array(in byte),

LR = Lower Limit of row (If not given assume it as zero

LC = Lower Limit of column (If not given assume it as zero),

LB = Lower Limit of blocks

N = Number of column

R = Number of blocks

Example:

Given an array arr[1:9, -4:1, 5:10] with base value 400 and size of each element is 2 Bytes in memory find the address of element arr[5][-1][8] with the help of row-major order?

I= 5

J = -1

K = 8

B = 400

W = 2

LR = 1

LC = -4

LB = 5

Number of column given in the matrix N = Upper Bound – Lower Bound + 1

= 1 – (-4) + 1

= 6

Number of blocks given in the matrix R = Upper Bound – Lower Bound + 1

= 10 – 5 + 1

= 6

Address of[i][j][k] = B + W \* {[(I – LR) \* N] + [(J – LC)] \* R + [K – LB]}

Address of[i][j][k] = 400 + 2 \* {[(5 – 1) \* 6] + [(-1 + 4)]} \* 6 + [8 – 5]

= 400 + 2 \* ((4 \* 6 + 3) \* 6 + 3)

= 400 + 2 \* (165)

= 730

**Column Major Order:**

Address of[i][j][k] = B + W \* {[(I – LR)] + [(J – LC) \* M]\* R + [K – LB]}

I = Row address to be found,

J = Column address to be found

K = Block address to be found

B = Base address

W = Storage array(in byte),

LR = Lower Limit of Row (If not given assume it as zero),

LC = Lower Limit of column (If not given assume it as zero),

LB = Lower Limit of blocks

M = Number of rows

R = Number of blocks

Example:

Given an array arr[1:8, -5:5, -10:5] with base value 400 and size of each element is 4 Bytes in memory find the address of element arr[3][3][3] with the help of column-major order?

Given:

I = 3

J = 3

K = 3

B = 400

W = 4

LR = 1

LC = -5

LB = -10

Number of rows given in the matrix M = Upper Bound – Lower Bound + 1

= 8 – 1 + 1

= 8

Number of blocks given in the matrix R = Upper Bound – Lower Bound + 1

= 5 + 10 + 1

= 16

Address of[i][j][k] = B + W \* {[(I – LR)] + [(J – LC) \* M]\* R + [K – LB]}

Address of[3][3][3] = 400 + 4 \* {[(3 – 1)] + [3 + 5] \* 8]} \* 16 + [3 + 10]

= 400 + 4 \* ((2 + 64) \* 16 + 13)

= 400 + 4 \* (1069)

= 400 + 4276

= 4676

Q1.