Arrays

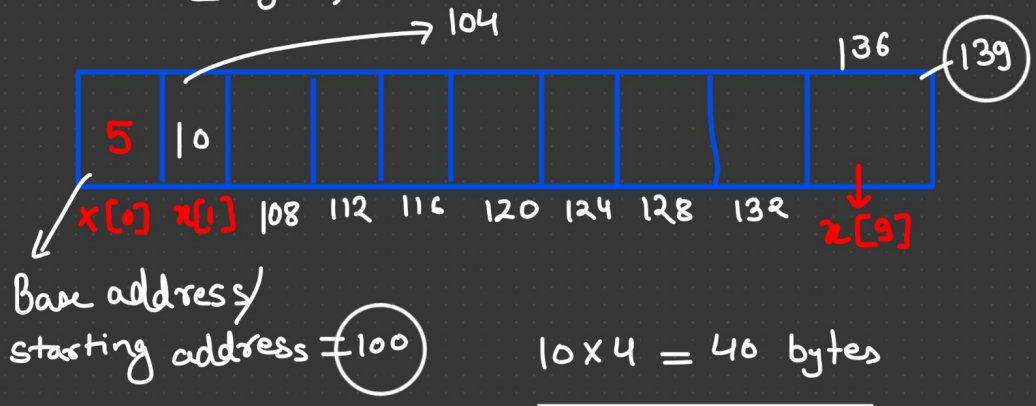
1) How to create an Array?

Data Type variable name [size];

Ex:- int x[10];

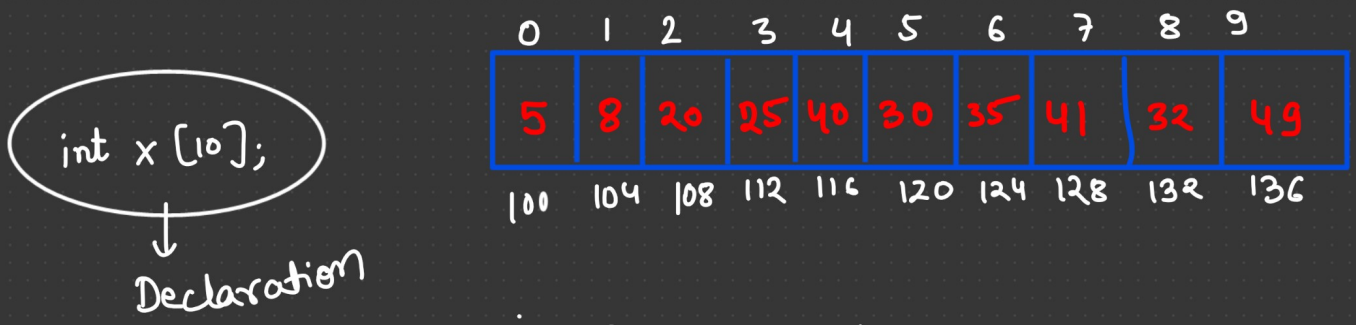
x[0] = 5;
x[1] = 10;
⋮

int = 4 byte



10 × 4 = 40 bytes

* Array is collection of similar type of Elements, stored in Contigous memory allocation.



X = Base address = 100
X[0] = Value at 100. = 5

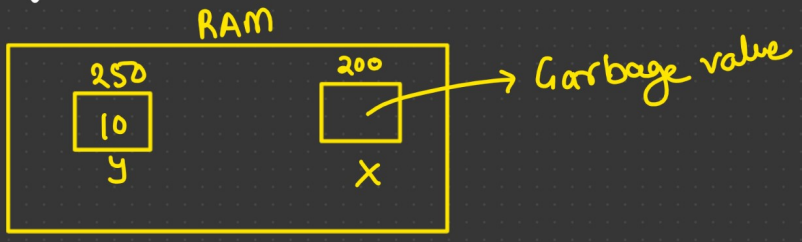
X+1 = 104
X[1] = value at 104 = 8

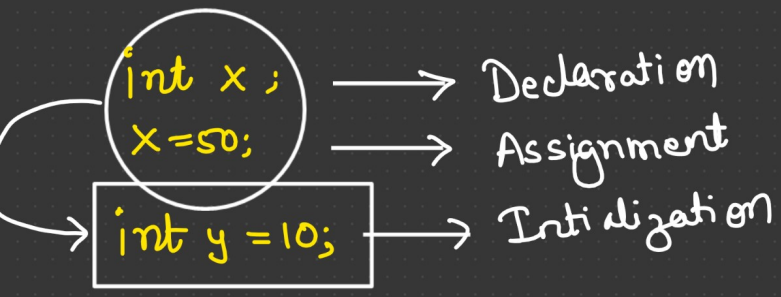
X+6 = 124
X[6] = value at 124 = 35

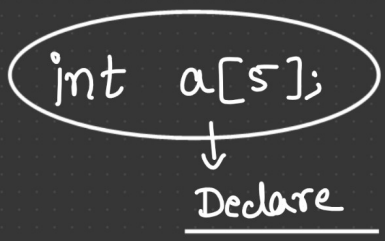
x[0] = x + 0X4 = 100 + 0 = 100
x[2] = 100 + 2X4 = 108
x[6] = 100 + 6X4 = 124

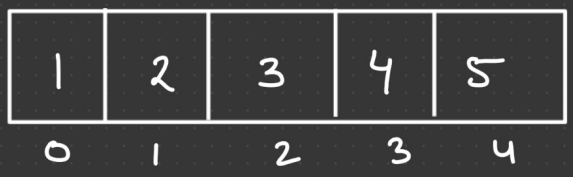
Address = Base Address + (index x size)

Intilialization of an Array:-










```
int a[5] = {1,2,3,4,5};  // Intialization
int a[5] = {1,2,3};      // 1, 2, 3, 0, 0
int a[ ] = {1,2,3,4,5};  //size = 5.
```

---

2) Accessing Array Elements:-

```
for(i=0 ;i<n ;i++)
{
    printf(" %d", a[i]);

}
```

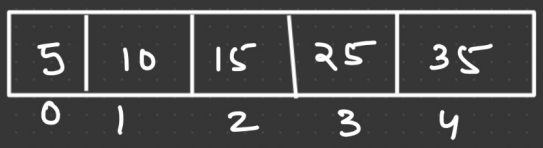
i = 0 , a[0] = 5

i = 1 , a[1] = 10

i = 2 , a[2] = 15

i = 3 , a[3] = 25

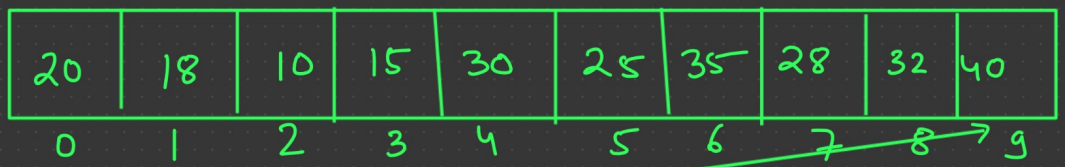
i = 4 , a[4] = 35

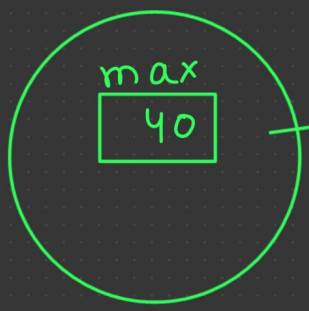


Q:- WAP to find maximum element from the array.

↓

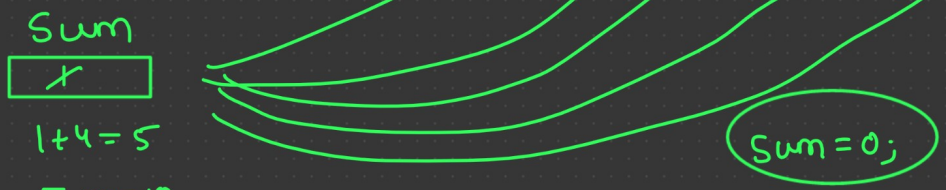
Write a program






```
max = a[0];
for (i= 1 ; i< 10 ; i++)
{
    if (max < a[i])
        max = a[i];
}
printf(" Maximum is %d", max);
```

Q:- Find the sum of all the elements in the given array.



1+4 = 5

5+5 = 10

10+3 = 13

13+4 = 17

```
for(i=0 ; i<5 ; i++)
{
    sum = sum + a[i];
}
```

Q:- WAP to sort the given array in increasing order

{ 2, 6, 4, 9, 8 } ⟶ { 2, 4, 6, 8, 9 }

a[0]

4	6	8	10	12	9	14
0	1	2	3	4	5	6

swap(10,6);	swap(10,6);
swap(6,4);	swap(10,8);

i=0 [1, 2, 3, 4, 5, 6]

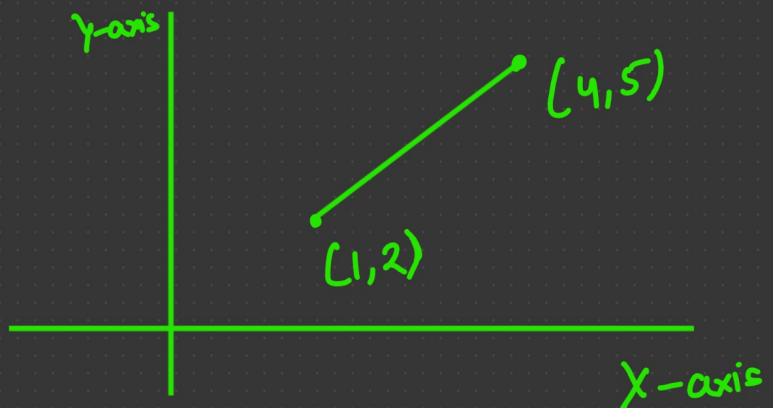
i=1 [2, 3, 4, 5, 6]

i=2 [3, 4, 5, 6]

```
for(i=0 ; i<6 ; i++)
{
   for(j=i+1 ; j<7 ; j++)
   {
      if(a[i] > a[j])
         swap(a[i], a[j]);
   }
}
```

2D Array :-

row	0	1	2	3 (Columns)
→ 0	00	01	02	03
→ 1	10	11	12	13
→ 2	20	21	22	23
→ 3	30	31	32	33



Matrix = 4 x 4 = 16 cells

(Row, Column)

int a[4][4];

Row (0-3)

Column (0-3)

```
int a[5][4];
```

→ 5 rows

→ 4 columns

(0-4) (0-3)

	0	1	2	3
0	0	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	6
4	4	5	6	7

← i

↑ j

```
a[0][0] = 1;
a[0][2] = 4;
a[2][2] = 6;
```

```
for(i=0; i<=4; i++)
{
    for(j=0; j<=3; j++)
    {
        a[i][j] = i+j;
    }
}
```

i = 0 1

j = 0 1 2 3

Memory Representation of 2D Array -

```
int a[4][4];
```

	0	1	2	3	
100 → 0	1	2	3	4	→ 16 byte
116 → 1	5	6	7	8	→ 16 byte
132 → 2	9	10	11	12	→ 16 byte
3	13	14	15	16	

4×4

Row 0 | Row 1 | Row 2 | Row 3

100	104	108	112	116	120	124	128	132	136	140	144				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

$a[1][2] = 100 + [(4 \times 4) \times 1] + (4 \times 2) = 124$

$a[2][3] = 100 + [(4 \times 4) \times 2] + (4 \times 3) = 144$

(no. of column) Sizeof(int)

int a[R][C]; a[i][j]

Address = Base Address + [C x (sizeof Data Type) x i] + (sizeof DT) x j

Q:- Addition of two 3x3 Matrix.

$$A = \begin{bmatrix} 2 & 1 & -1 \\ 4 & 5 & 2 \end{bmatrix}_{3\times3}, \quad B = \begin{bmatrix} 2 & 0 & 6 \\ -1 & 2 & 4 \end{bmatrix}_{3\times3}$$

$$A+B = \begin{bmatrix} 4 & 3 & 7 \\ 4 & 1 & 5 \\ 3 & 7 & 6 \end{bmatrix}$$

```
int a[3][3]= {1,2,3,2,1,-1, 4,5,2};
int b[3][3]= { 3,1,4, 2,0,6, -1,2,4};
 int c[3][3];
for (i=0 ; i<3 ; i++)
{  for(j=0;j<3 ; j++)
   {
     c[i][j] = a[i][j] + b[i][j];
   }
}
```

i=0 j= 0, 1, 2

[0][0] [0][1] [0][2]

Q:- Matrix multiplication 2x2.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix}$$

A B

$$\begin{bmatrix} 00 & 01 \\ 10 & 11 \end{bmatrix} \times \begin{bmatrix} 00 & 01 \\ 10 & 11 \end{bmatrix}$$

$$A \times B = \begin{bmatrix} 1\times1 + 2\times2 & 1\times(-1) + 2\times(3) \\ 3\times1 + 4\times2 & 3\times(-1) + 4\times(3) \end{bmatrix}$$

Res $\begin{bmatrix} 00 & 01 \\ 10 & 11 \end{bmatrix}$

R=0 x C1 R=0 x C2

R=1 x C1 R=1 x C2

```
for(i=0; i<2; i++)
{
    for(j=0; j<2; j++)
    {
        for(k=0; k<2; k++)
        {
            sum = sum + a[i][k]*b[k][j];
        }
        c[i][j] = sum;
    }
}
```

i=0 → i=0

j=0 → j=0

k=0, 1 → k=(0,1)

i=0, j=0 → sum = (00) x (00) + (01) x (10)

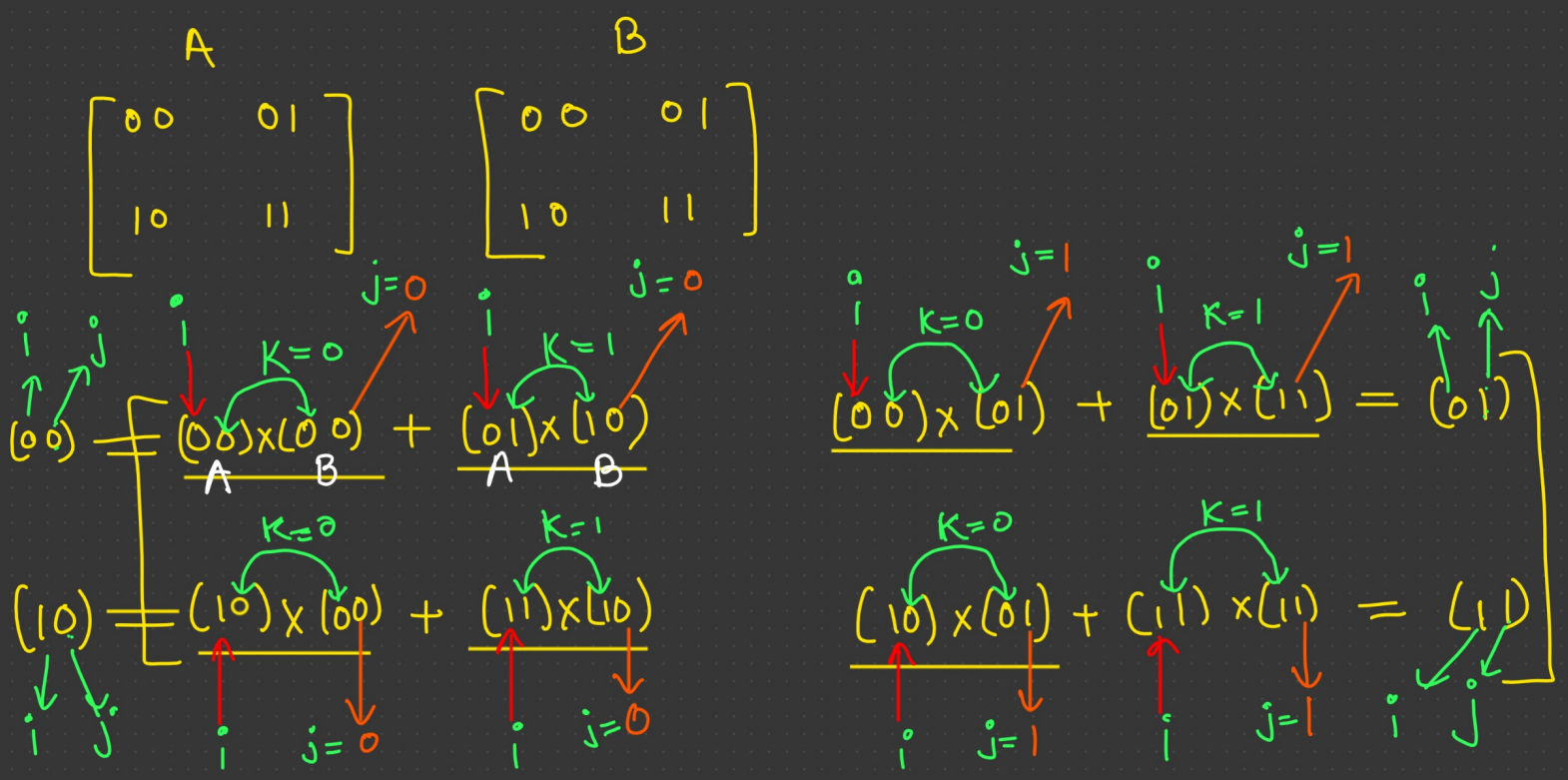
(00) x (01) + (01) x (11)

i=1 | j=1 | k=(0,1)

j=0, j=1 | k=(0,1)

a[i][k]*b[k][j] → sum

sum = sum + a[i][k]*b[k][j]




```
for(i=0 ; i<2 ; i++)
{
   for (j=0; j<2; j++)
   {
      sum=0;
     for(k=0; k<2; k++)
     {
       sum= a[i][k] * b[k][j] + sum;
     }c[i][j] = sum;
   }
}
```

Q:- Find Transpose of the given matrix?

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

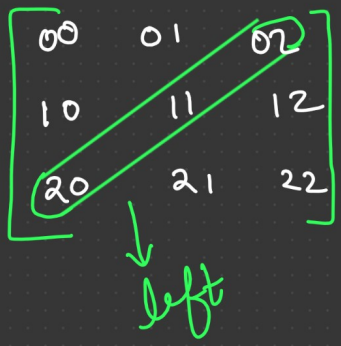
$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

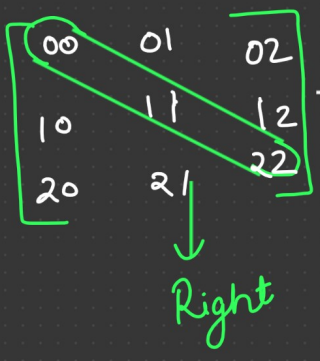
$$\begin{bmatrix} 00 & 01 & 02 \\ 10 & 11 & 12 \\ 20 & 21 & 22 \end{bmatrix} \rightarrow \begin{bmatrix} 00 & 10 & 20 \\ 01 & 11 & 21 \\ 02 & 12 & 22 \end{bmatrix}$$

```
for(i=0; i<3; i++)
{
  for(j=0; j<3; j++)
  {
    b[i][j] = a[j][i];
  }
}
```

$$\begin{bmatrix} 00 & 01 & 02 \\ 10 & 11 & 12 \end{bmatrix}_{2\times3} \qquad \begin{bmatrix} 00 & 10 \\ 01 & 11 \\ 02 & 12 \end{bmatrix}_{3\times2}$$

Q:- Sum of right diagonal / left diagonal?






```
for(i=0; i<3; i++)
{ sum = a[i][i] + sum;
}
```

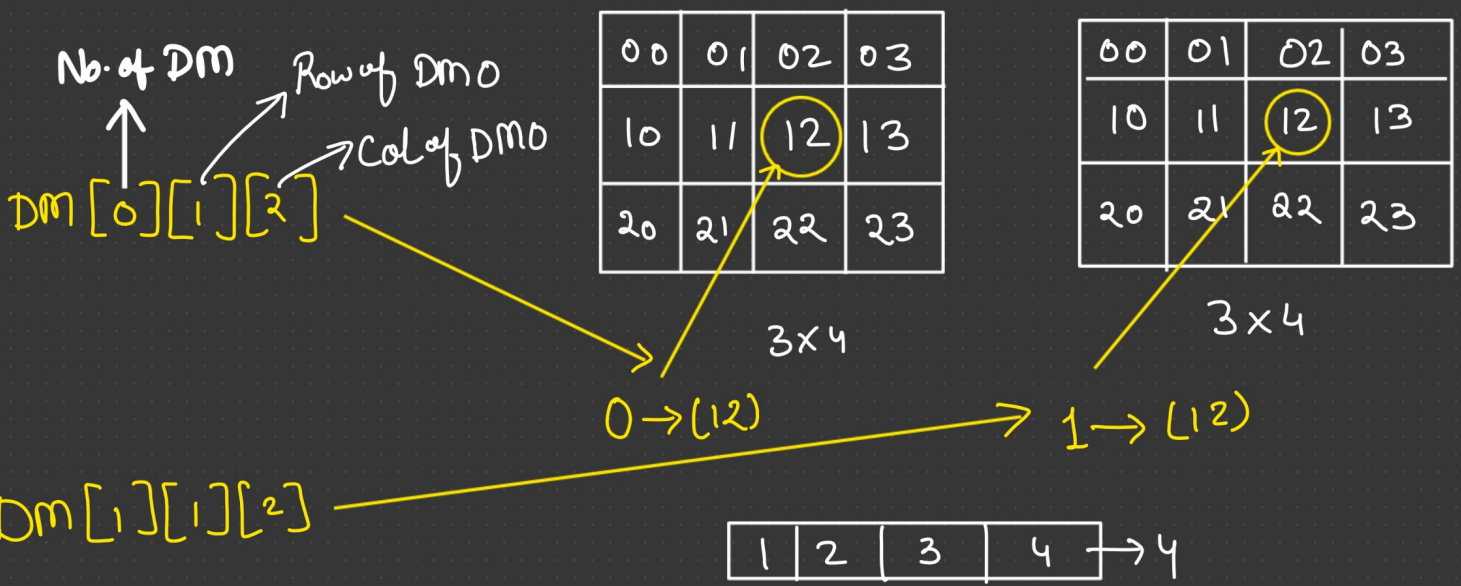
```
if(i+j==2)
sum = a[i][j] + sum;
```

```
for(i=0, j=2; i<3; i++, j--)
  sum = a[i][j] + sum;
```

3D Array :-

DaryMilk = 0

DaryMilk = 1



1DM ⟶ 12 pieces (each row 4 piece)

→ 4
→ 4
12

$\frac{12}{4} = 3$

Box ⟶ 4DM

Size of Box = ?

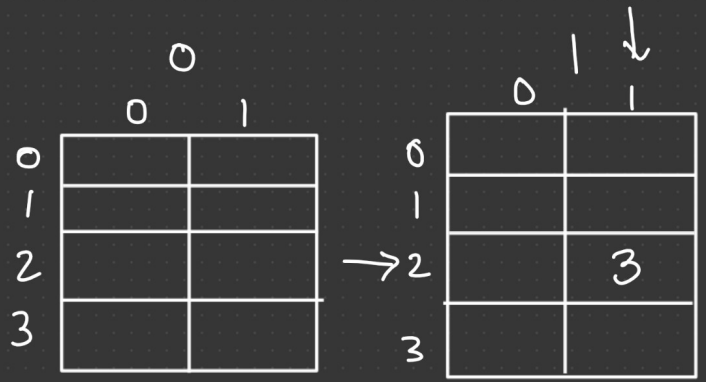
Box[4][3][4]; 4×3×4

no. of DM

no. of Row

no. of col

int a[3][4][2];



a[1][2][1] = 3;

No. of element

= 3×4×2

= 24

```
int a[2][2][3] = {1,2,3,4,5,6,7,8,9,10,11,12};
int a[2][2][3] = { {{1,2,3},{4,5,6}},{{7,8,9},{10,11,12}} };
```

1	2	3
4	5	6

0

7	8	9
10	11	12

1

```
int a[ ][2][3] = {1,2,3,4,5,6,7,8,9,10,11,12};
                  0 0 1 1
```

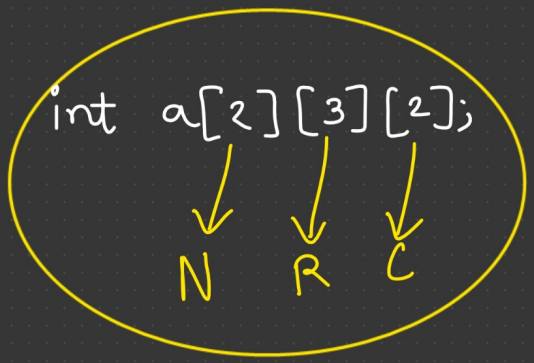
1 matrix, 2 Matrix

```
int a[ ][3][2] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17};0
                  0 0 0 1 1 1 2 2 2
```

a[3]

① ② ③

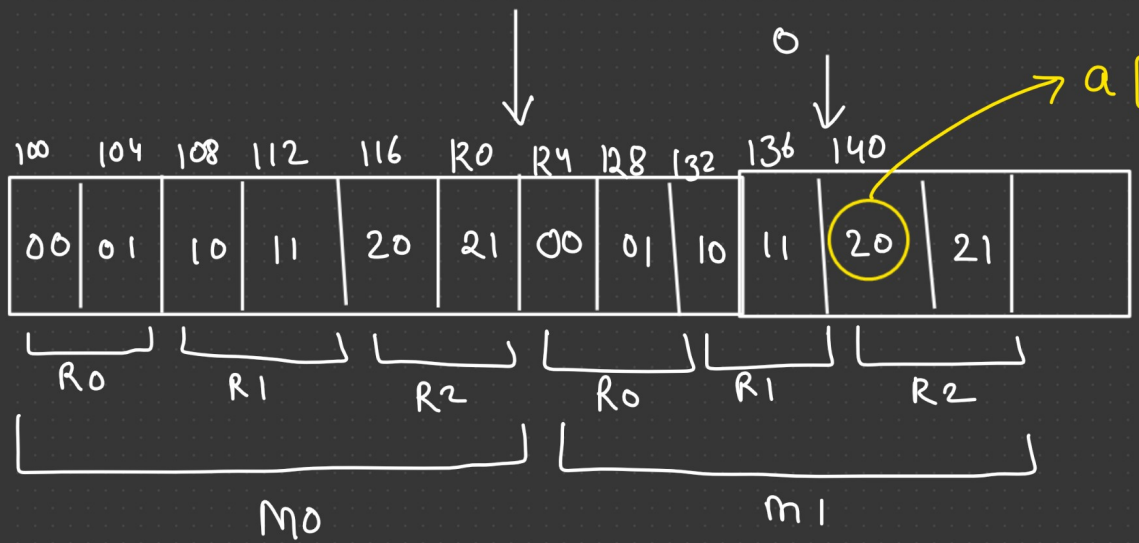
```
for(i=0; i<3; i++)
{
  for(j=0; j<3; j++)
  {
    for(k=0; k<2; k++)
    {
      a[i][j][k] = i+j+k;
    }
  }
}
```



100

00	01
10	11
20	21

00	01
10	11
20	21



$$a[1][2][0] = 100 + [1 \times (3) \times (2) \times 4] + [2 \times (2) \times 4] + (0 \times 4)$$

$$= 100 + 24 + 16 + 0 = 140$$

$$\text{Address} = \text{Base Address} + [i \times (R \times C) \times \text{size}] + [j \times (C \times \text{size})] + (K \times \text{size})$$