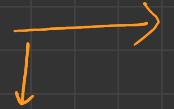


Rows, 6 columns

2D ARRAY → 2 Dimension



Excel Sheet

| Name | Class | Roll | Reg | Addl | cont |
|--------|--------------------|------|------|-------|------|
| Arpit | 10 th A | 24 | 2338 | xyz 2 | 987 |
| Kabir | 12 th B | 35 | 6218 | xyz 2 | 1, |
| Raj | 11 th C | 44 | 0104 | xyz 2 | 1, |
| Soniya | 9 th A | 03 | 7763 | xyz 2 | " |

inf

arr [4] [6];

column



row



2 0

Row →



column

| 0 | 1 | 2 | 3 | 4 | 5 | |
|---|-----|-----|-----|-----|-----|-----|
| 0 | 0,0 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 |
| 1 | 1,0 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 |
| 2 | 2,0 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 |
| 3 | 3,0 | 3,1 | 3,2 | 3,3 | 3,4 | 3,5 |

Chessboard

| | | | | | | |
|----|----|----|----|----|----|----|
| | | | | | | |
| 00 | 01 | 02 | 03 | 04 | 05 | 10 |
| 11 | 12 | 13 | 14 | 15 | 21 | 24 |

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|-----|-----|-----|-----|-----|-----|
| 0 | 0,0 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 |
| 1 | 1,0 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 |
| 2 | 2,0 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 |
| 3 | 3,0 | 3,1 | 3,2 | 3,3 | 3,4 | 3,5 |

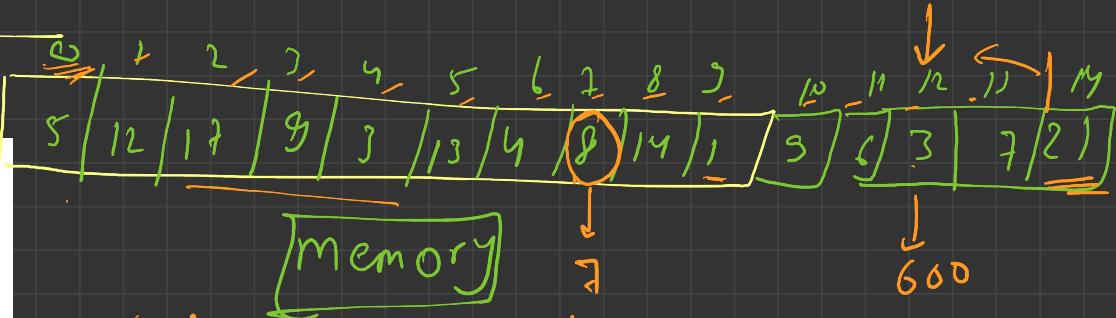
① Row major

② Column major

① Row major

| Columns → | | | | | |
|-----------|----|----|----|----|----|
| Rows ↓ | 0 | 1 | 2 | 3 | |
| 0 | 5 | 12 | 17 | 9 | 3 |
| 1 | 13 | 4 | 8 | 14 | 1 |
| 2 | 9 | 6 | 3 | 7 | 21 |

2D Array of size 3 x 5



$$\text{arr}[i][j] = \text{base address} + \underbrace{\text{index} \times \text{size}}_{\text{of element}}$$

$$\text{row index} - \frac{\text{index}}{\text{column}}$$

$$\text{Column index} = \text{index \% columns}$$
$$600 + 12 \times 4 = 648$$

$$\text{Index} = \text{row index} \times \text{column} + \text{col index}$$

$$\text{index} = \underline{\text{rowindex}} \times \cancel{\text{col}} + \text{colindex}$$

0, 1, 2, 3, 4, 5

$$\boxed{0 < \text{colindex} < \text{col}}$$

$$\frac{\text{index}}{\cancel{\text{col}}} = \cancel{\text{rowindex}} \times \cancel{\text{col}} + \frac{\text{colindex}}{\cancel{\text{col}}}$$

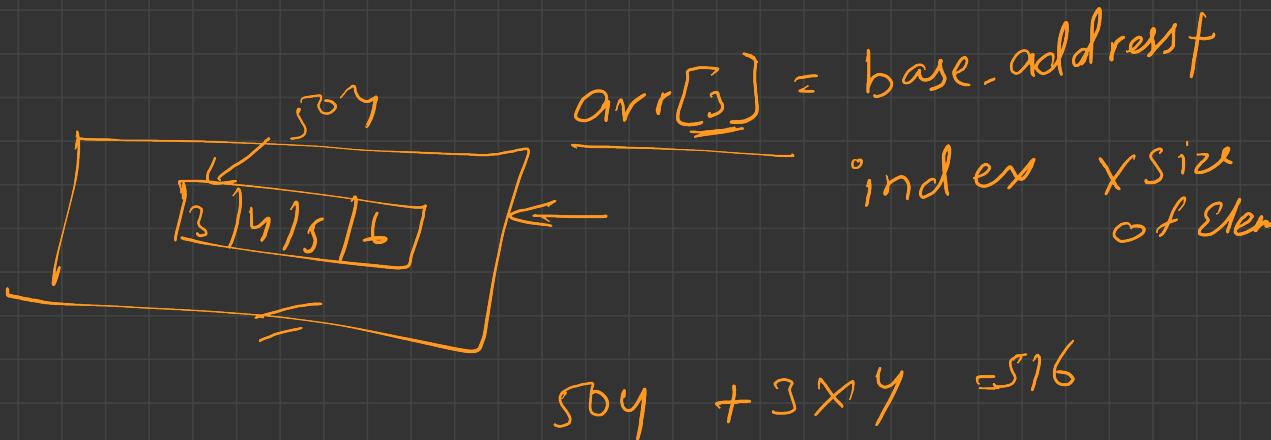
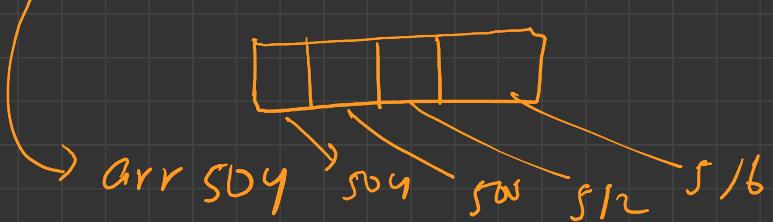
$$\boxed{\frac{\text{index}}{\text{col}} = \text{rowindex}}$$

$$\text{index} \circ \omega | \left(\underbrace{\text{rowindex} \times \text{col}}_{=} + \underbrace{\text{colindex}}_{=} \right) \circ \text{col}$$

$$(\text{rowindex} \times \text{col}) \circ \omega = 0$$

$$\boxed{\text{colindex} = \text{index} \circ \omega |}$$

int arr [4]



$\boxed{\text{arr}[i][j] = \text{base_add} + (i \times \text{col} + j) \times \text{size of Element}}$

① `int arr[4][3]`

$$4 \times 3 = 12$$

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

| 0 | 1 | 2 |
|---|----|----|
| 0 | 1 | 2 |
| 1 | 4 | -5 |
| 2 | 7 | 8 |
| 3 | 10 | 11 |
| | | 12 |

arr[2][2] → 12

③ update or user input :

$\text{arr}[1][2] = 16$

$\text{cin} \gg \text{arr}[0][2]$

④ Print

$\text{cout} \ll \text{arr}[i][j];$

$\text{for } (i=0; i < \text{row}; i++) \{$

$\text{for } (j=0; j < \text{col}; j++) \{$

| | Column 0 | Column 1 | Column 2 |
|-------|----------|----------|----------|
| Row 0 | x[0][0] | x[0][1] | x[0][2] |
| Row 1 | x[1][0] | x[1][1] | x[1][2] |
| Row 2 | x[2][0] | x[2][1] | x[2][2] |

Search Element in Array

∴ int arr [4][3]

Target = 17

Yes No

int main () {

int arr [4][3] = { - - - - }

int target = 17;

for (i=0 ; i<4; i++) {

 for (j=0 ; j<3; j++) {

 if (arr[i][j] == target) {

 cout << "Yes"

 cout << "No" } return 0;

```
void printCol ( int arr [ j [ 3 ], int row, int col ) {  
    for ( i = 0 ; i < row ; i++ )  
        for ( j = 0 ; j < col ; j++ )  
            cout << arr [ i ] [ j ]
```

```
int main () {  
    int arr [ 4 ] [ 3 ] = { - - - }  
    printCol ( arr, 4, 3 )
```

row $\overleftarrow{\text{Ar1}}[3][4] \rightarrow \text{col}$

| | | | |
|----|----|----|----|
| 1 | 3 | 5 | 7 |
| 10 | 11 | 16 | 20 |
| 23 | 30 | 34 | 60 |

$\text{Ar2}[3][s]$

$O(\text{row} \times \Theta)$

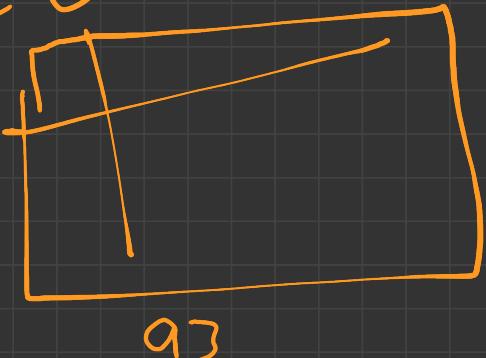
$O(\text{row} \times \omega)$.

| | | | | |
|---|---|---|---|--|
| 0 | 0 | 0 | 1 | |
| 0 | 1 | 0 | 0 | |
| 0 | 0 | 0 | 0 | |

Ar1

$a_1[i][j] + a_2[i][j]$

$a_3[i][j]$



Ar2

$(a_1[0][0] + a_2[0][0])$

$a_3[0][0]$

int arr [2][2] = { {1, 2}, {2, 4} } }

int arr [3][4] = { {1, 2, 3}, {3, 4, 5, 6}, {7, 8, 11} }

| | | | |
|---|---|----|--|
| 1 | 2 | 3 | |
| 3 | 4 | 5 | |
| 4 | 5 | 6 | |
| 7 | 8 | 11 | |

Print row index with max sum

| | 0 | 1 | 2 | 3 |
|---|----|----|----|----|
| 0 | 1 | 3 | 5 | 7 |
| 1 | 10 | 11 | 16 | 20 |
| 2 | 23 | 30 | 34 | 60 |

$\rightarrow 16$

$\rightarrow 47$

$\rightarrow 147$

$arr[3][q]$

$total = \emptyset$

$min < 167$

$sum = 167$

$index = 2$

$i = \emptyset$

$j = \emptyset$

$int sum = INT-MIN,$
 $index = -1;$

```

for (i=0 ; i<row ; i++)
{
    int total = 0;
    for (j=0 ; j<col ; j++)
        total += arr[i][j];
    if (sum < total)
        sum = total;
        index = i;
}
cout << index;

```

Sum of diagonal Element

| | | | |
|---------|----------|----------|---------|
| 1 00 | 3 | 5 | 7 00 |
| 10 | 11 | 16 12 | 20 |
| 23 | 30 21 | 34 22 | 60 |

row = col

i = 0

j = col - 1

i = j

4
30 6 3 2
33

$$1 + 11 + 34 + 2 \rightarrow \text{sum}$$

first = 0
for (i = 0; i < row; i++)
 first += arr[i][i]

first + arr[i][j] → sum

03 12 21 30

{ Reverse each row of Matrix }

| | | | |
|----|----|----|----|
| 1 | 3 | 5 | 7 |
| 10 | 11 | 16 | 20 |
| 23 | 30 | 34 | 60 |

7 5 31
20 16 11 10
60 34 30 23

for (j=0; j < row; j++)

{ Reverses i-th row }

start = 0 end = col - 1
while (start < end)
 swap (arr[i][start], arr[i][end]);

start++
end--;
}

$N \cdot \omega$

- ① multiply of 2×20 matrix
- ② Reverse each Array of Matrix

