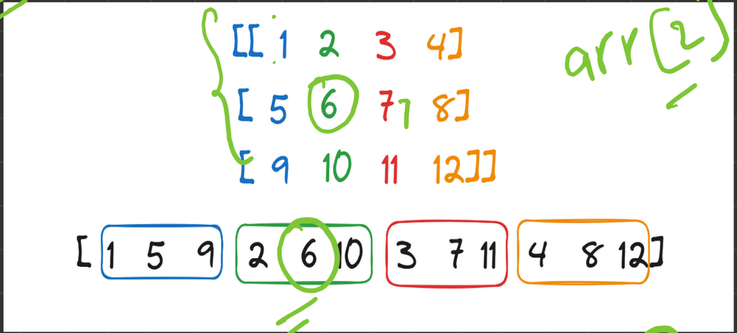
Column Major Order

C++

Java

row



$$\left[\begin{array}{cccc}[1 & 2 & 3 & 4] \\ [5 & \boxed{6} & 7 & 8] \\ [9 & 10 & 11 & 12]\end{array}\right]$$

$$[\underbrace{1\ 5\ 9}\ \underbrace{2\ \boxed{6}\ 10}\ \underbrace{3\ 7\ 11}\ \underbrace{4\ 8\ 12}]$$

arr[2][2] ?

Row

H.W

Vector in 2D Array

// create

vector<vector<int>> name;

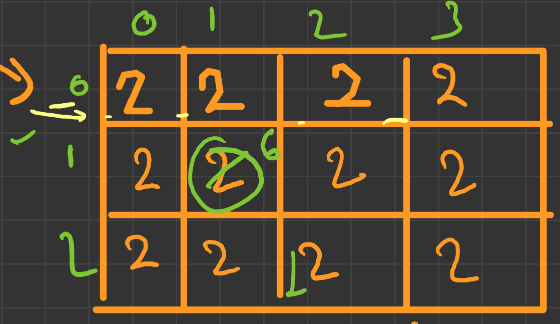
row

column

?

vector<vector<int)name (rows, vector<int> col, init)

3 2 4



	0	1	2	3
0	2	2	2	2
1	2	~~2~~ 6	2	2
2	2	2	2	2

name

name[1][1] = 6

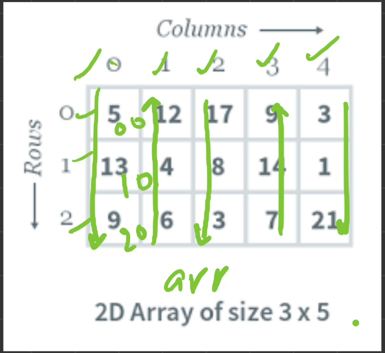
cin >> name[2][2]

row = name.size()

col = name[0].size()

find row, col

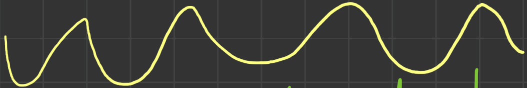
Wave form



	0	1	2	3	4
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

→arr



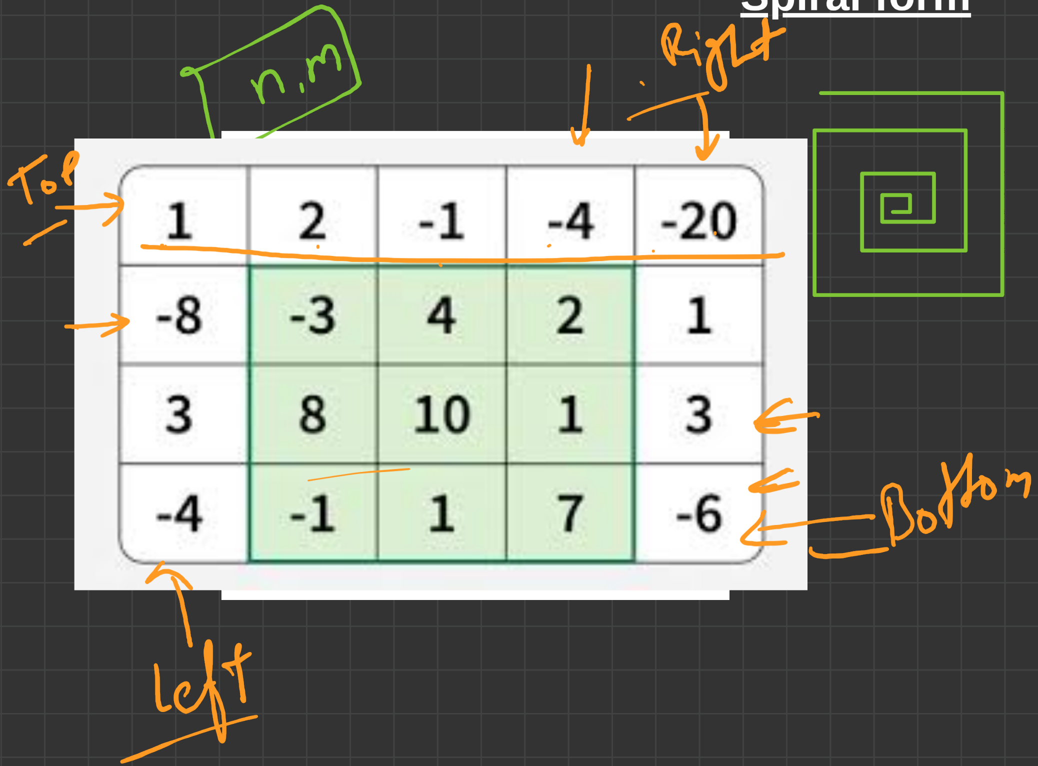
col = Even (up -> down)
col = odd (down -> up)

```
for (int j= 0;j< col;j++)
    if (j%2==0)
        for(i= 0; i<row; i++)
            cout<< arr[i][j]

    else
        for(i=row-1; i >=0 ;i--)
            cout << arr [i] [j]
```

arr[0][j]
arr[1][j]
arr[2][j]

Spiral form



1	2	-1	-4	-20
-8	-3	4	2	1
3	8	10	1	3
-4	-1	1	7	-6

Top = 0	→	Row
Right = 4	→	Col
Bottom = 4	→	Row
left = 0	→	Col

```
for (j = left ; j<= Right ; j++)          0
      ✓ Cout << arr [top][j]
          top++;
   for(i= top ; i<= bottom ;i++)
       cout <<arr [i][Right]
       Right --;
```

```
for (k= Right ; k>= left ;k--) {
    cout << arr[bottom][k];
    Bottom -- ;
for ( l= bottom; l >= top; l--)
    cout << arr [l][left)

    left ++
```