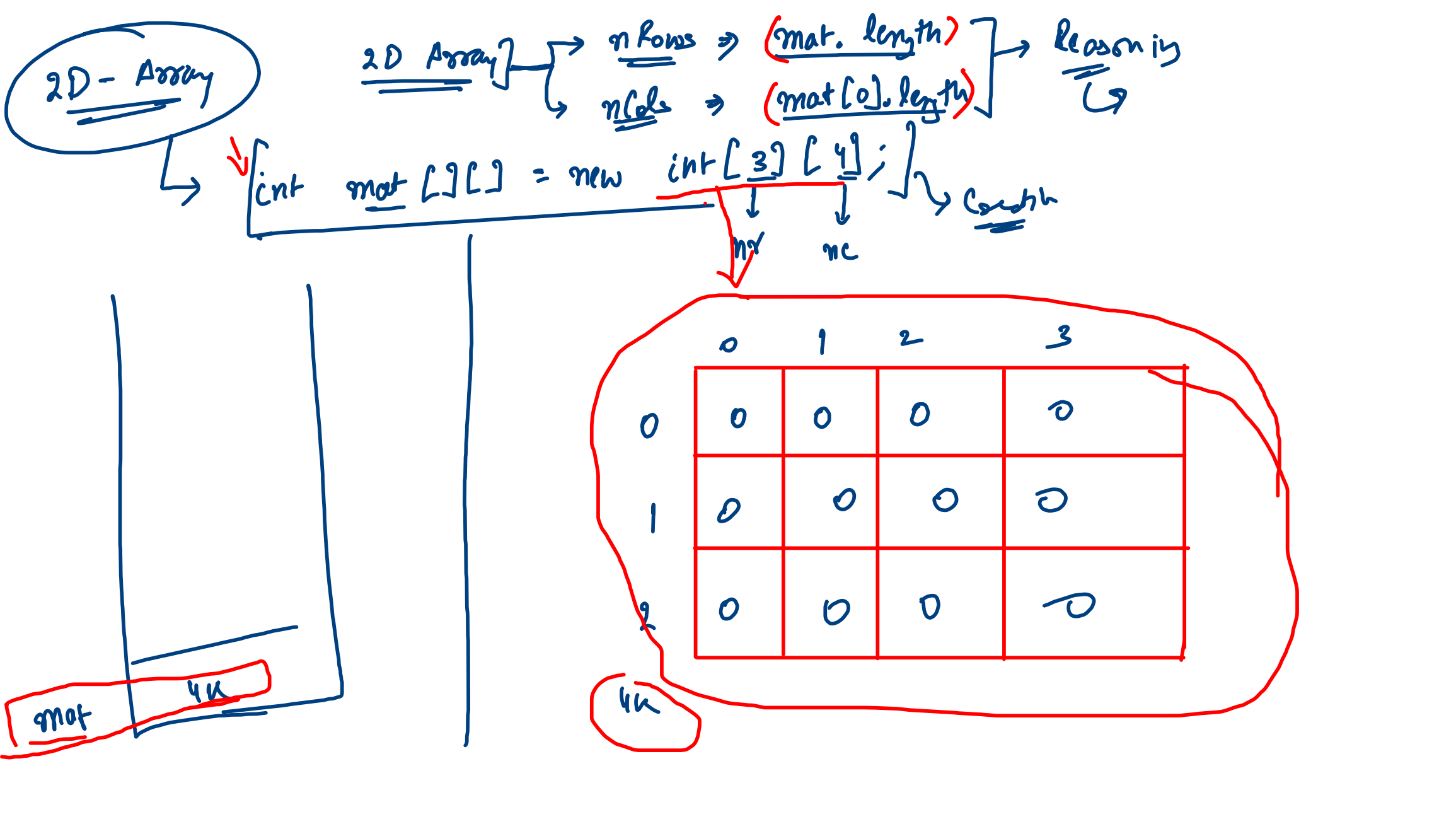
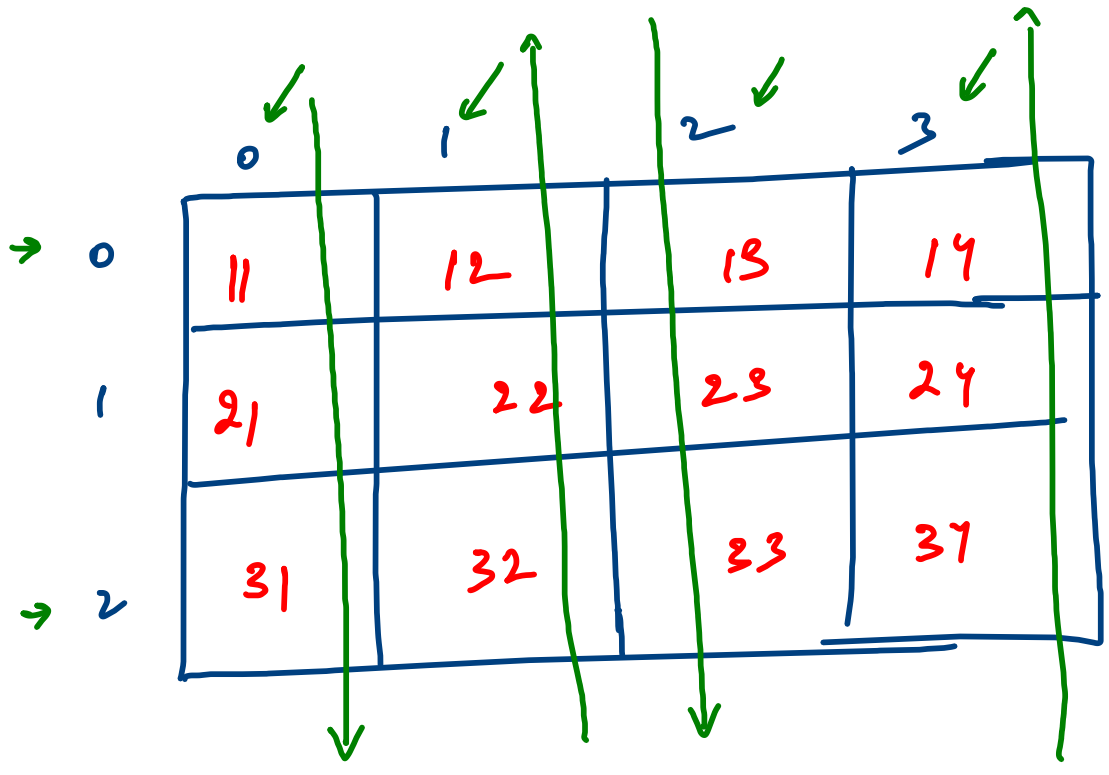
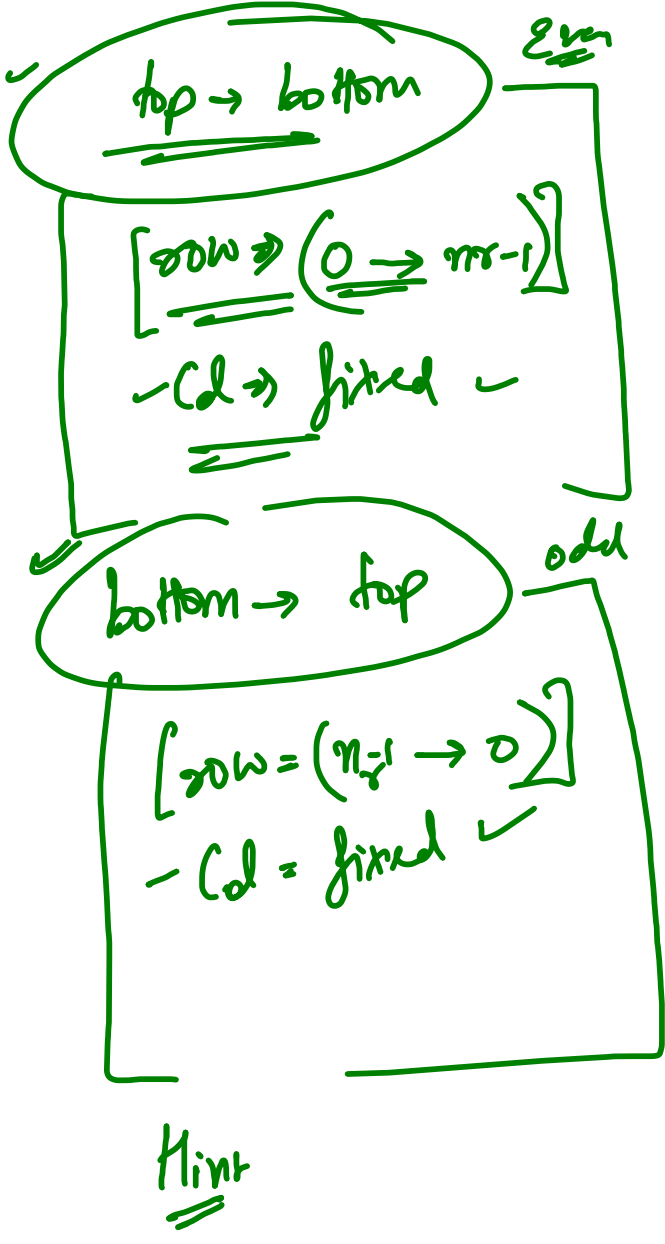
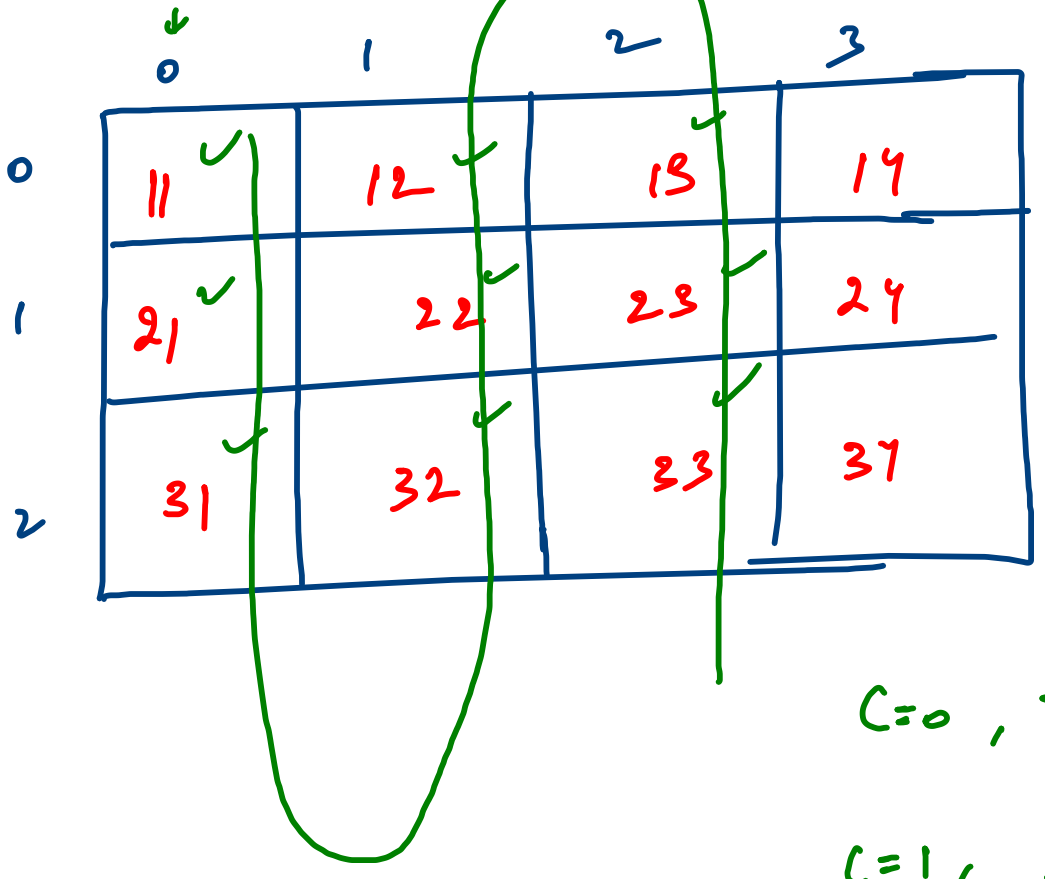

2D - Array
2D Array
nRows ⇒ (mat. length)
nCols ⇒ (mat[0]. length)
Reasoning
int mat [][] = new int [3] [4];
nr
nc
Length
0 1 2 3
0 0 0 0
1 0 0 0
2 0 0 0
mat
4k
4k

3
4
11 12 13 14
21 22 23 24
31 32 33 34





nr : 3




```
for(int c = 0 ; c < nc ; c++){
    if(c % 2 == 0){
        // even : top -> bottom
        for(int r = 0 ; r <= nr-1 ; r++){
            System.out.println(mat[r][c]);
        }
    }else{
        // odd : bottom -> top
        for(int r = nr-1 ; r >= 0 ; r--){
            System.out.println(mat[r][c]);
        }
    }
}
```

c=0 , r=0,1,2

c=1 , r=2,1,0

c=2 , r=0,1,2

[0][0] → 11

[1][0] → 21

[2][0] → 31

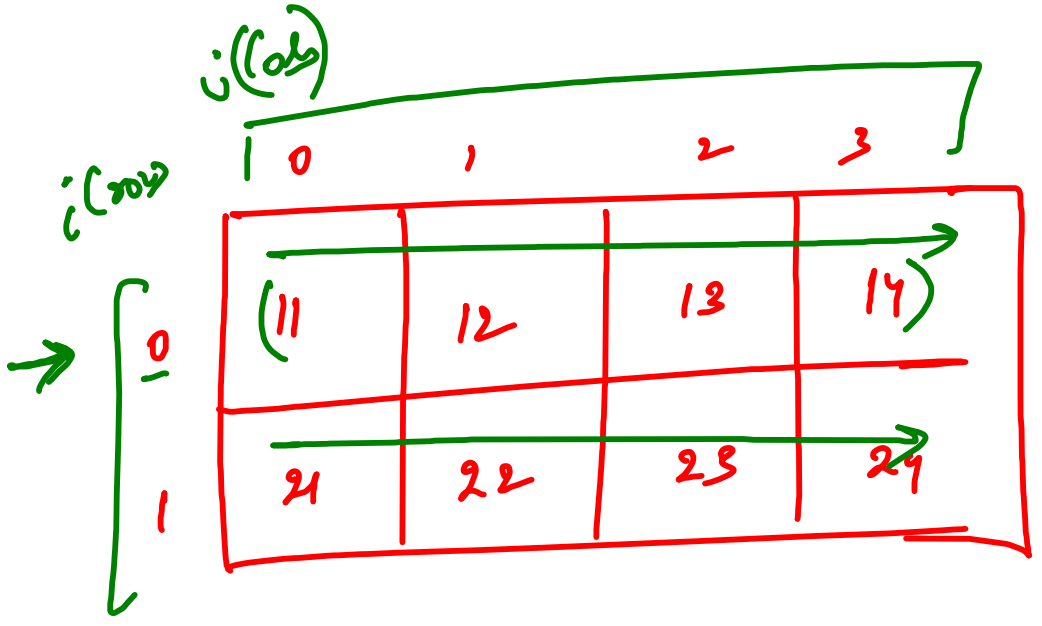
[2][1] → 32

[1][1] → 22

[0][1] → 12

2 → nr
4 → nc

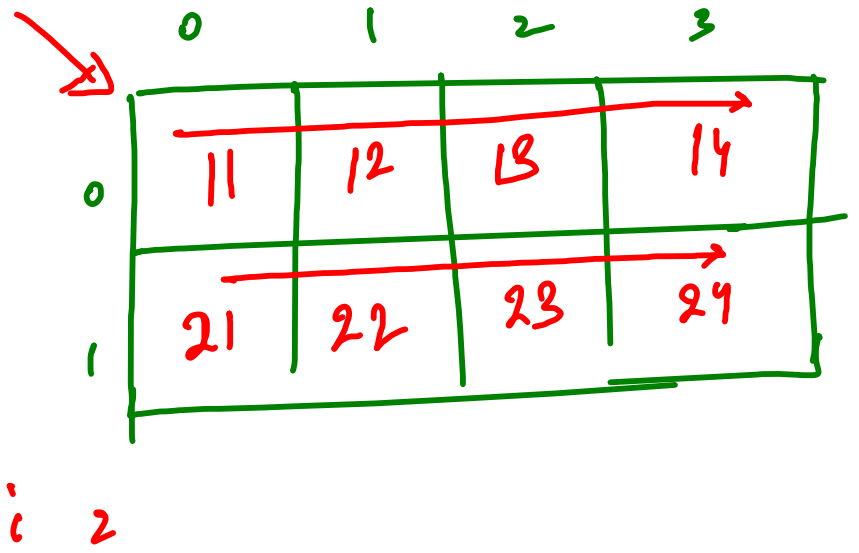
11
12
13
14
21
22
23
24



nr = 2

nc = 4

2
4
11
12
13
14
21
22
23
24



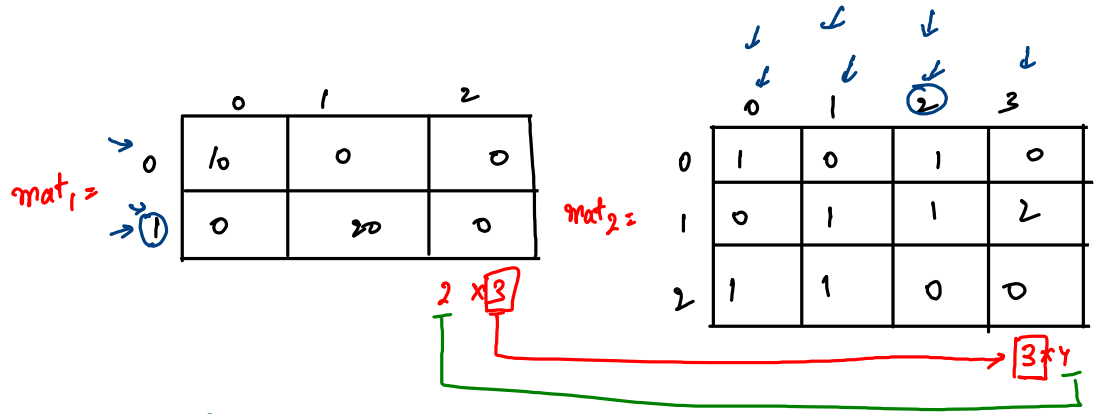
i 2

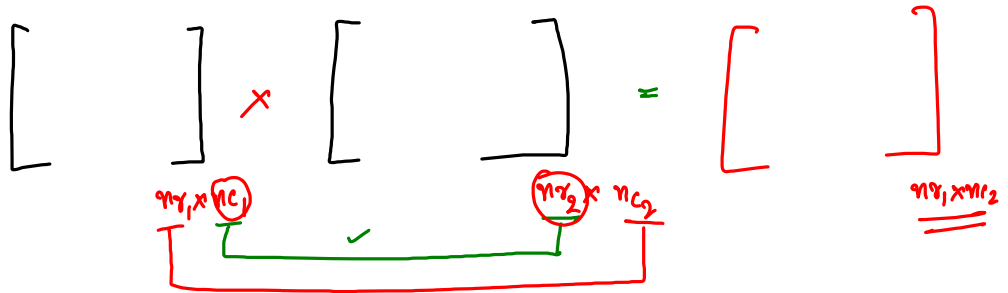
j

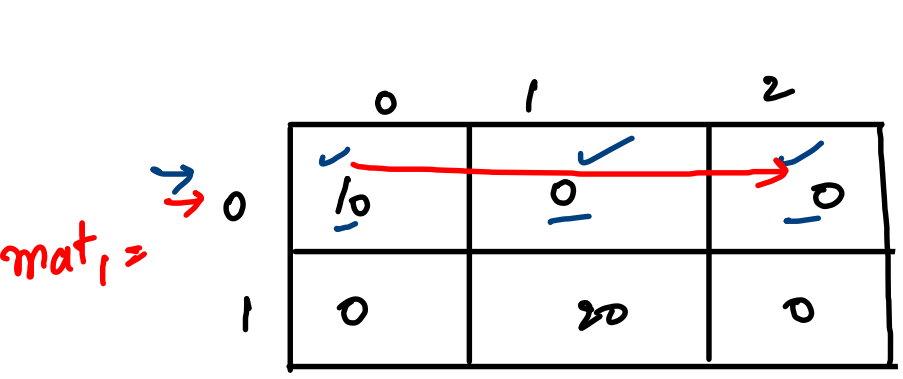
```
int nr = scn.nextInt();
int nc = scn.nextInt();

int mat[][] = new int[nr][nc];

for(int i = 0 ; i < nr ; i++){
    for(int j = 0 ; j < nc ; j++){
        mat[i][j] = scn.nextInt();
    }
}
```

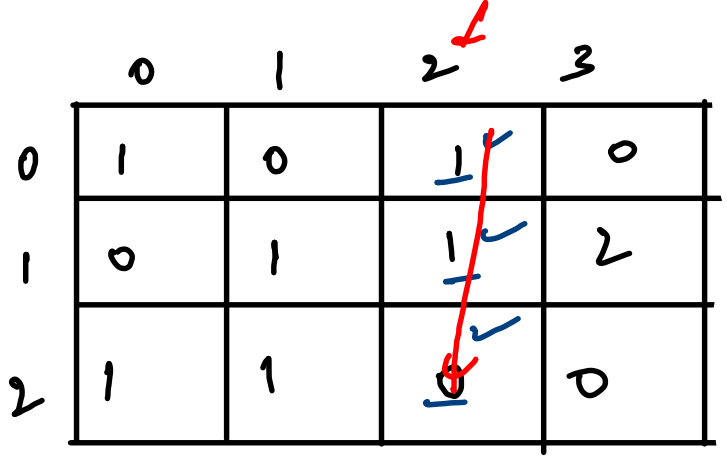
row

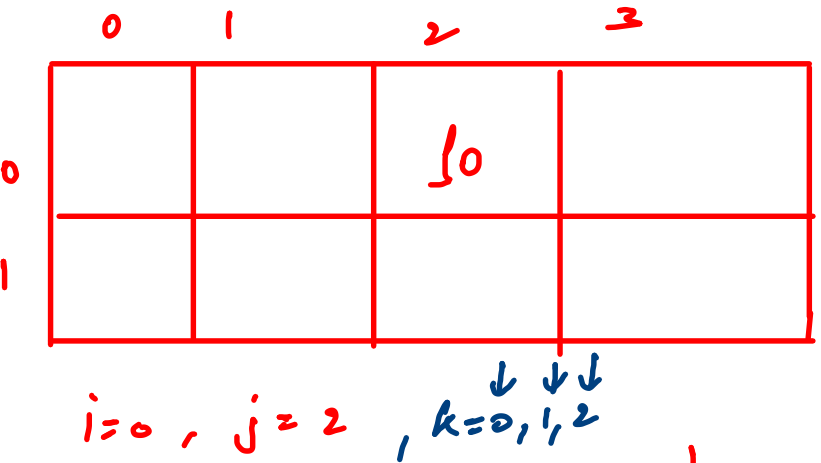
col


mat1 =
mat2 =
2 x 3
3 x 4
res =
res[1][2] = (0.1) + (20.1) + (0.0)
= 20


nr1 x nc1
nr2 x nc2
nr1 x nr2

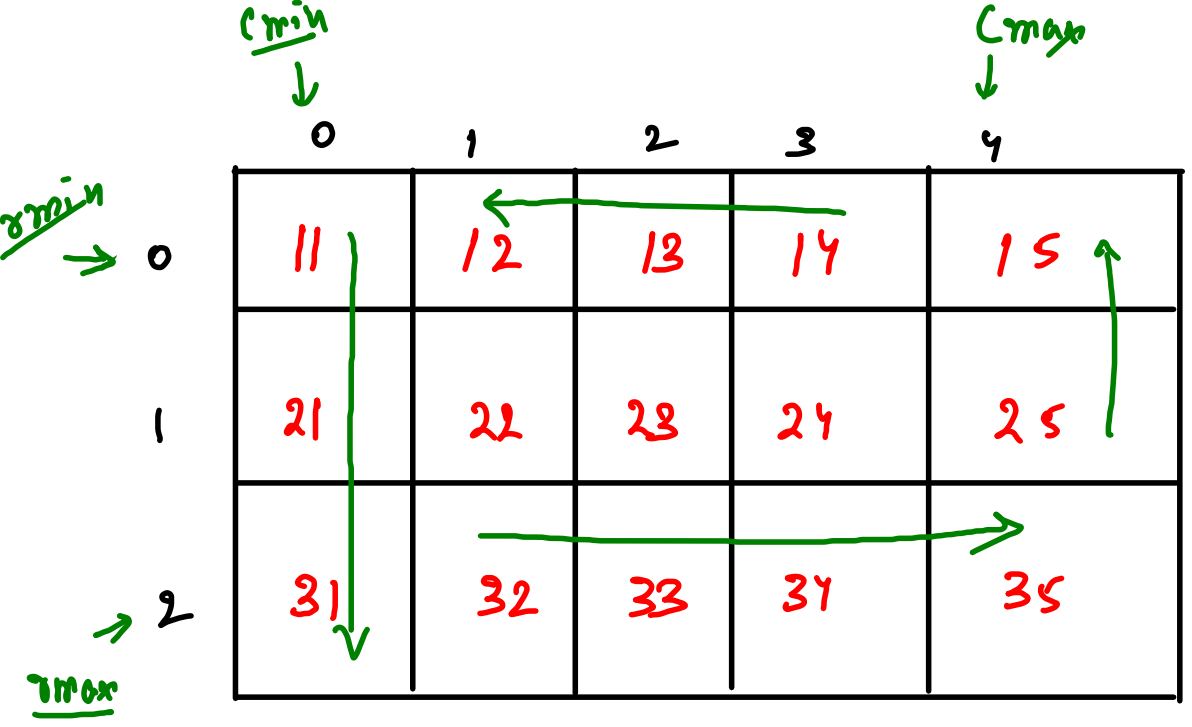


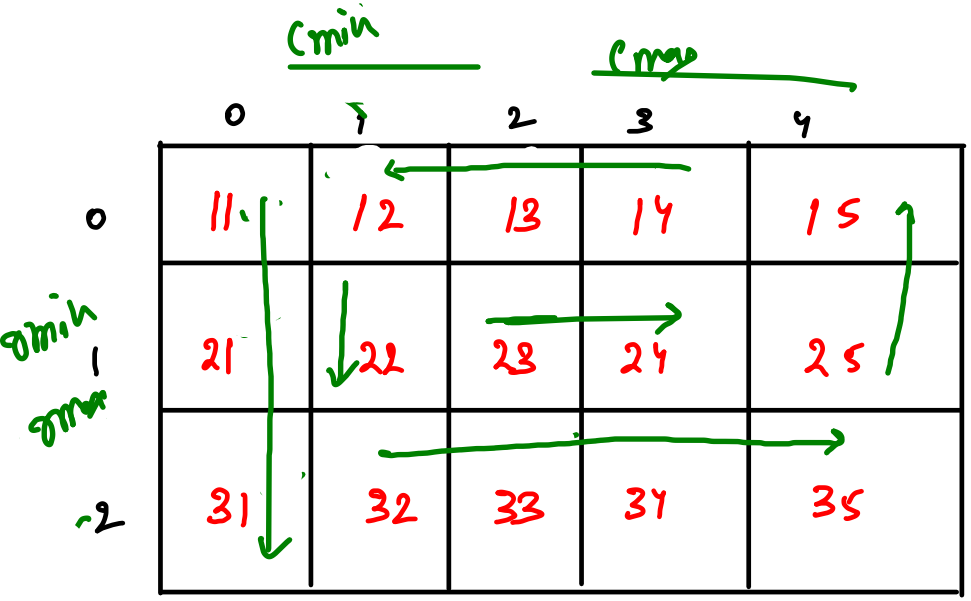


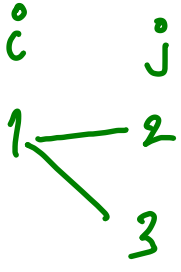



```
int res[][] = new int[nr1][nc2];

// matrix multiplication

for(int i = 0 ; i < res.length ; i++){
    for(int j = 0 ; j < res[0].length ; j++){

        for(int k = 0 ; k < nc1 ; k++){
            res[i][j] = res[i][j] + (mat1[i][k] * mat2[k][j]);
        }
    }
}
```

res[0][2] = (m1[0][0] . m2[0][2]) + (m1[0][1] . m2[1][2]) + (m1[0][2] . m2[2][2])

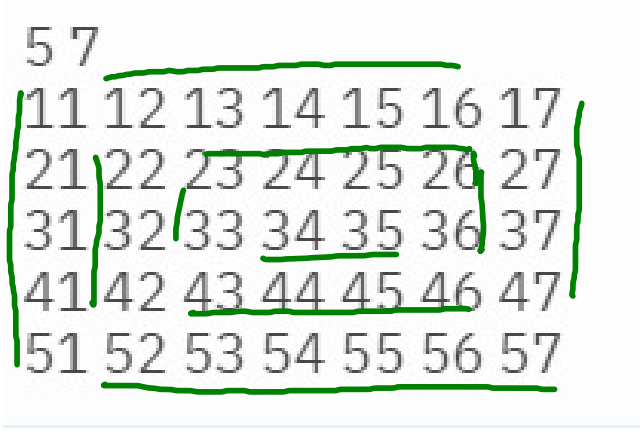

cmin
cmax
0
1
2
3
4
rmin
0
1
2
rmax
11
12
13
14
15
21
22
23
24
25
31
32
33
34
35



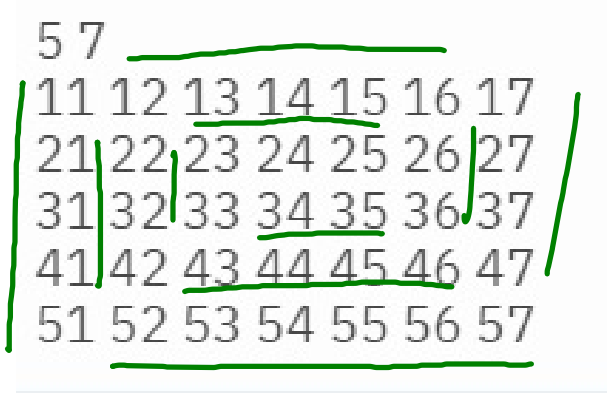


11
21
31
32
33
34
35
25
15
14
13
12
22
23
24

```java
public static void spiralDisplay(int mat[][]){
    int rmin = 0, cmin = 0, rmax = mat.length-1,cmax = mat[0].length-1;

    while(){
        for(int i = rmin , j = cmin ; i <= rmax ; i++){
            System.out.println(mat[i][j]);
        }

        for(int i = rmax , j = cmin+1 ; j <= cmax ; j++){
            System.out.println(mat[i][j]);
        }

        for(int i = rmax-1 , j = cmax ; i >= rmin; i--){
            System.out.println(mat[i][j]);
        }

        for(int i = rmin , j = cmax-1 ; j >= cmin+1 ; j--){
            System.out.println(mat[i][j]);
        }

        rmin++;
        cmin++;
        rmax--;
        cmax--;
    }
}
```

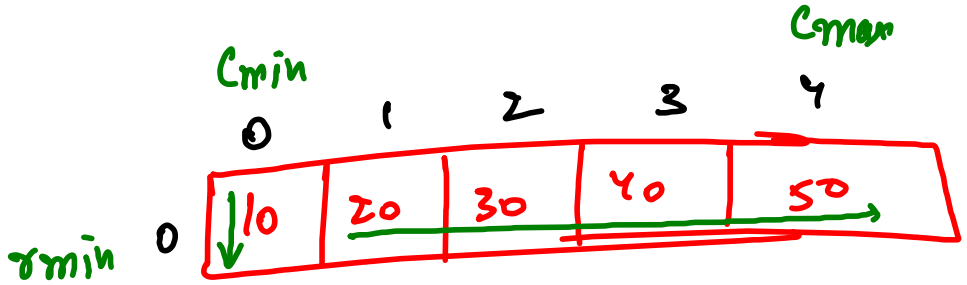


11
21
31
41
51
52
53
54
55
56
57
47
37
27
17
16
15
14
13
12
22
32
42
43
44
45
46
36
26
25
24
23
33
34
35



11
21
31
41
51
52
53
54
55
56
57
47
37
27
17
16
15
14
13
12
22
32
42
43
44
45
46
36
26
25
24
23
33
34
35 34

Count = 5 x 2 = 10 | total Element = 5



0 0 → 10

0 1 → 20

0 2 → 30

0 3 → 40

0 4 → 50

```java
int rmin = 0, cmin = 0, rmax = mat.length-1,cmax = mat[0].length-1;
int count = 0;
int totalEle = mat.length * mat[0].length;
while(count < totalEle){
    for(int i = rmin , j = cmin ; i <= rmax && count < totalEle ; i++){
        System.out.println(mat[i][j]);
        count++;
    }

    for(int i = rmax , j = cmin+1 ; j <= cmax && count < totalEle ; j++){
        System.out.println(mat[i][j]);
        count++;
    }

    for(int i = rmax-1 , j = cmax ; i >= rmin && count < totalEle ; i--){
        System.out.println(mat[i][j]);
        count++;
    }

    for(int i = rmin , j = cmax-1 ; j >= cmin+1 && count < totalEle; j--){
        System.out.println(mat[i][j]);
        count++;
    }

    rmin++;
    cmin++;
    rmax--;
    cmax--;
}
```