SCALER

2D Arrays

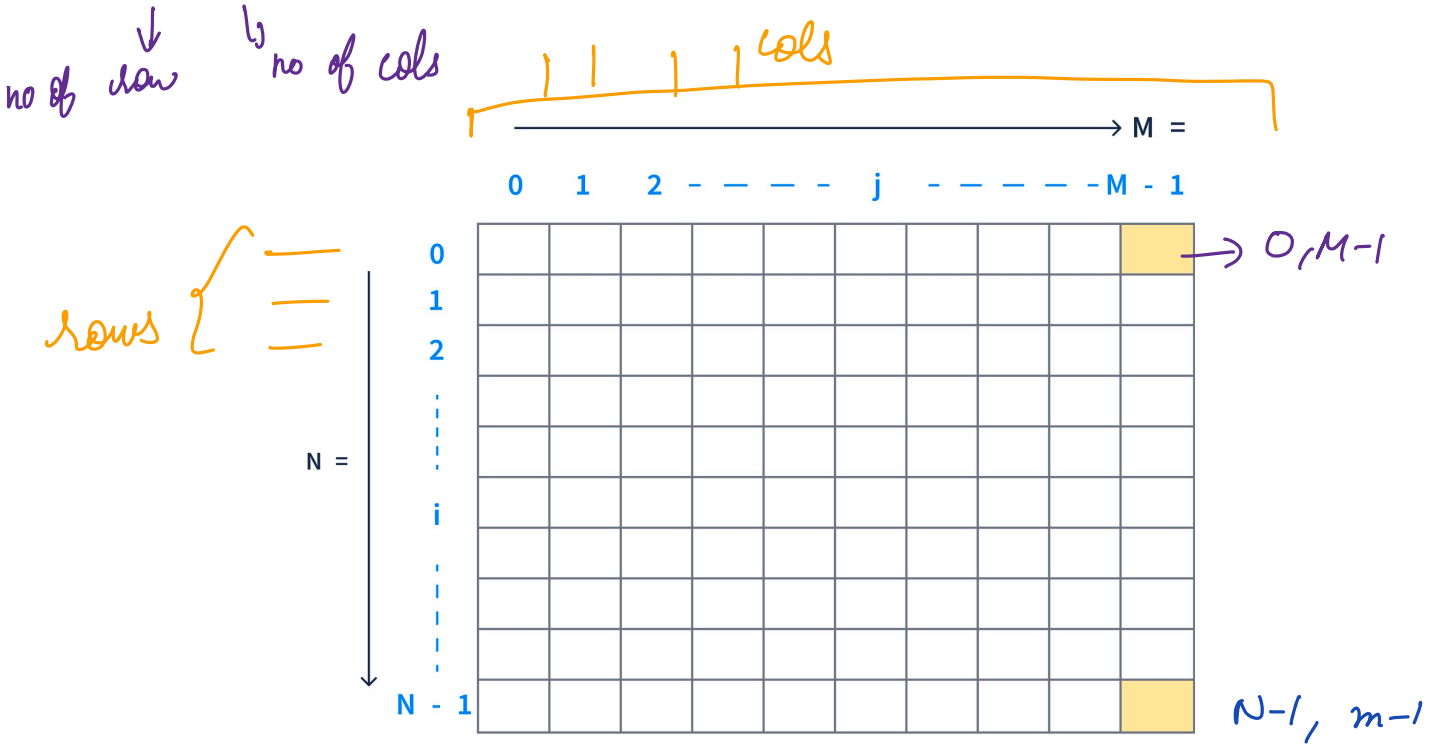
7 June Friday

TABLE OF CONTENTS

Notes

2 D array

- int arr[N][M] , total number of elements →



arr[row_no][col_no]

How is it actually stored?

< Question > : Given arr[N][M], print row-wise sum.

- arr [3] [4] →

no of cols

no of rows

	0	1	2	3	
0	10	2	7	3	→ 22
1	9	5	-1	8	→ 21
2	3	11	15	20	→ 49

usually row ⇒ i col ⇒ j

< / > Code

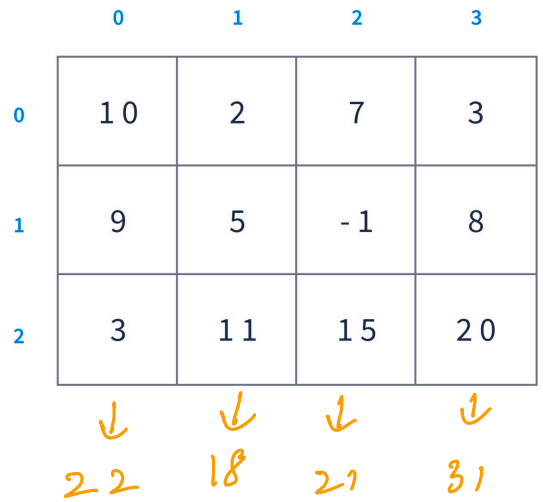
```
void printRow-wise sum(int arr[N][M]){
    for( i: 0 → N-1 ){                    N
        sum = 0
        for( j: 0 → M-1 ){                M
            sum += arr[i][j]
        }
        print(sum)
    }
}
```

TC: O(NM)

SC: O(1)

< *Question* > : Given arr[N][M], print column-wise sum.

- arr [3] [4] →



< / > *Code*

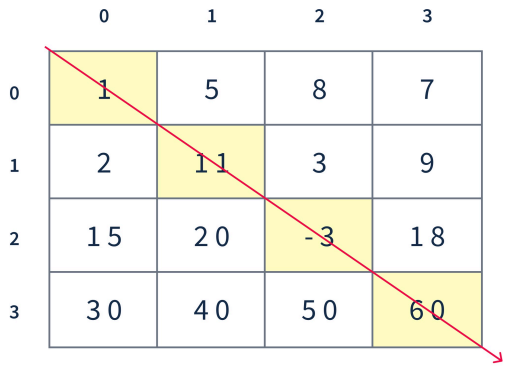
```
void printColumn-wise sum(int arr[N][M]){
    for ( j : 0 → M-1 ) {
        sum = 0
        for ( i : 0 → N-1 ) {
            sum += arr[i][j]
        }
        print (sum)
    }
}
```

N

j = 0

0, 0
1, 0
2, 0
:
n-1, 0

< Question > : Given arr[N][N] Square matrix

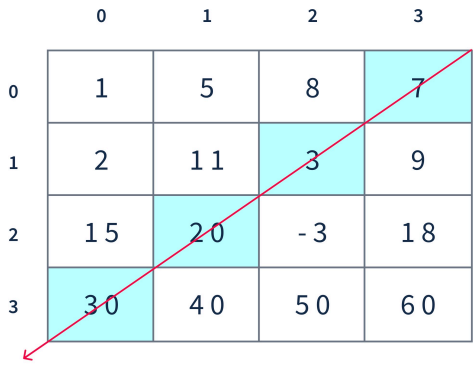


4*4

main diagnol

0,0
1,1
2,2
3,3

```
for(i: 0 -> N-1){
    print(arr[i][i])
}
```



4*4

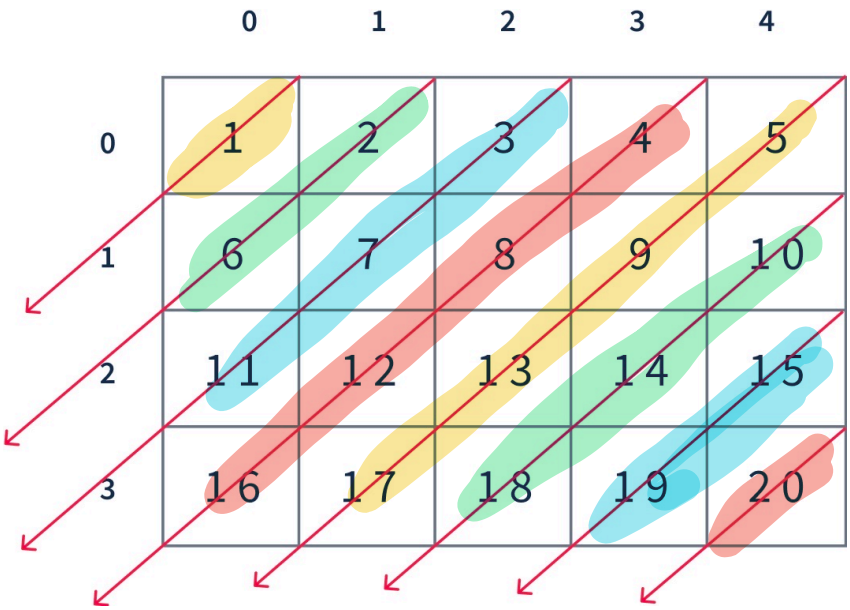
anti diagnol

0, 3
1, 2
2, 1
3, 0

```
i = 0    j = n-1
while(i < N){
    print(arr[i][j])
    i++
    j--
}
```

< *Question* > : Given arr[N][N]. Print all the elements diagonally from right to left.

- arr [4] [5] →



0,0

0,1 1,0

0,2 1,1 2,0

0,3 1,2 2,1 3,0

0,4 1,3 2,2 3,1

1,4 2,3 3,2

2,4 3,3

3,4

N = 4 M = 5

diag = 8

diag = N+M-1

< / > **Code**

```
void printAllDiagonals(int arr[N][M]){

    //print the diagonals starting from 0th row
    for(j: 0 → M-1){
        row = 0
        col = j
        while(row < N && col >= 0){
            print(all[row][col])
            row++
            col--
        }
    }
    //print the diagonals starting from last column
    for(i: 1 → n-1){
        row = i
        col = m-1
        while(row < N && col >= 0){
            print(all[row][col])
            row++
            col--
        }
    }
}
```

0,0
0,1 1,0
0,2 1,1 2,0

1,4 2,3 3,2
2,4 3,3
3,4

TC: NM

Transpose of a Square Matrix

$i, j \longleftrightarrow j, i$

$0,1 \longleftrightarrow 1,0$
$0,2 \longleftrightarrow 2,0$
$0,3 \longleftrightarrow 3,0$
$1,2 \longleftrightarrow 2,1$
$1,3 \longleftrightarrow 3,1$
$2,3 \longleftrightarrow 3,2$

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

Transpose →

	0	1	2	3
0	1	5	9	13
1	2	6	10	14
2	3	7	11	15
3	4	8	12	16

```
for (i: 0 → n-1) {
    for(j: i+1 → n-1) {
        swap(arr[i][j], arr[j][i])
    }
}
```

TC: $O(n^2)$

SC: $O(1)$

Rotate a mat[N][N]

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

Rotate by
90° Clockwise

	0	1	2	3
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4

Step 1 ⇒ Create a transpose

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

⇒

1 5 9 13
2 6 10 14
3 7 11 15
4 8 12 16

reverse each row