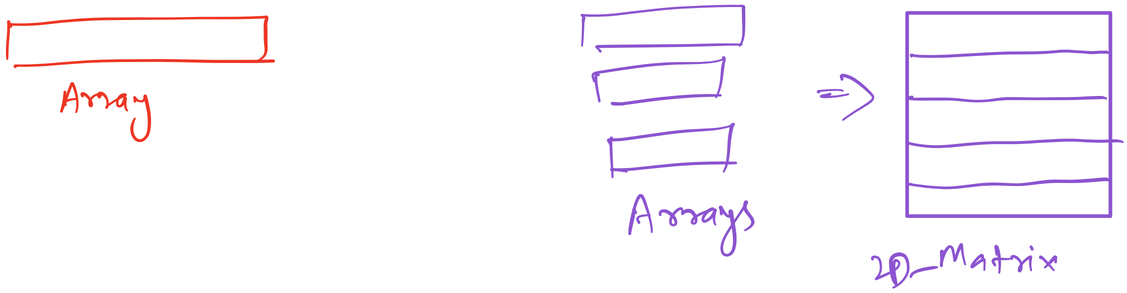
Arrays: 2D Matrices

2-D Matrix: Array of arrays



all arrays are of same size

So, size of each array in a matrices → M

Total no. of arrays → N

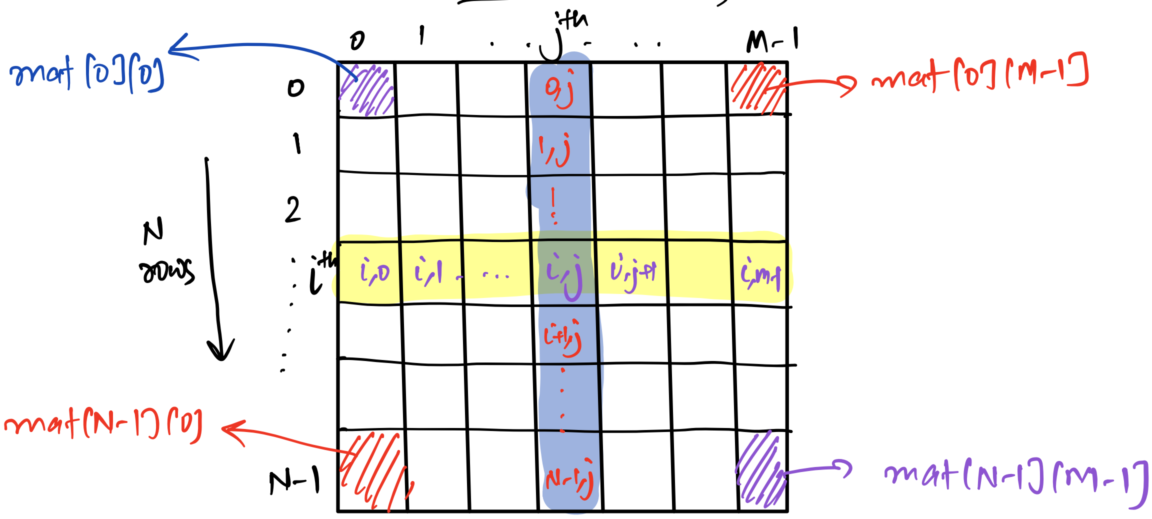
Declare: int mat[N][M] (TODO → learn in your own language)

mat[N] → no. of rows

[M] → no. of columns

Rows are horizontal

Columns are vertical



Observation

→ If we move in i^{th} row, column index will change from [0, M-1]

→ If we move in j^{th} column, row index will change from [0, N-1]

Question 1

Given mat [N][M], print row-wise sum.

eg

$$\begin{array}{c|ccc|l} & 0 & 1 & 2 & \\ \hline 0 & 1 & 5 & 9 & \rightarrow 15 \\ 1 & 2 & 8 & 2 & \rightarrow 12 \\ 2 & 7 & 6 & 2 & \rightarrow 15 \\ \hline \end{array}$$

Code

```
for ( i=0; i<N; ++i ) {
     // ith row
     sum =0
     for ( j=0; j<M; ++j ) {
          sum = sum + mat[i][j]
     }
     print (sum)
}
```

TC : O(N×M)

SC : O(1)

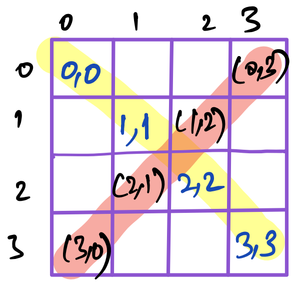
Question 2

Given square mat[N][N] , print diagonals.

left → right & right → left

Square matrix : (# of rows) = (# of colums)

eg



	0	1	2	3
0	0,0			(0,3)
1		1,1	(1,2)	
2		(2,1)	2,2	
3	(3,0)			3,3

```
i=0 , j=0                 optional
while ( i<N && j<N ){
    print( mat [i][j])
    i++ ,j++
}
```

left to right

```
for(i=0;i<N;++i) {
    for(j=0; j<N; ++j) {
        if (i==j)
            print(mat[i][j])
} }
```

NOT needed since TC: O(N²)

```
i=0, j=n-1                 optional
while (i<N && j>=0) {
    print(mat[i][j])
    i++, j--
}
```

right to left

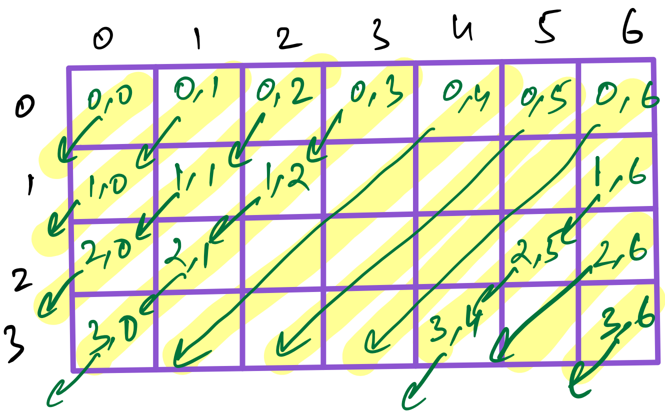
TC : O(N)

SC : O(1)

Question 3

Given mat[N][M], print all diagonals going from R → L.

Note: Diagonals starting from 0th row OR m-1th column.



	0	1	2	3	4	5	6
0	0,0	0,1	0,2	0,3	0,4	0,5	0,6
1	1,0	1,1	1,2				1,6
2	2,0	2,1				2,5	2,6
3	3,0				3,4		3,6

mat[4][7]

Diagonals starting point :

(0,0) (0,1) (0,2) (0,3) . . . . (0,6)

(1,6) (2,6) (3,6)

// print all diagonals from 0th row

```
for (K=0; K<M; ++K) {
    // starting : (0,K)
    i=0, j=K
    while (i<N && j>=0) {
        print (mat[i][j])
```

K=0 → (0,0)

K=1 → (0,1) (1,0)

K=2 → (0,2) (1,1) (2,0)

⋮

```
            i++ , j--
        }
        print (newline)
    }
```

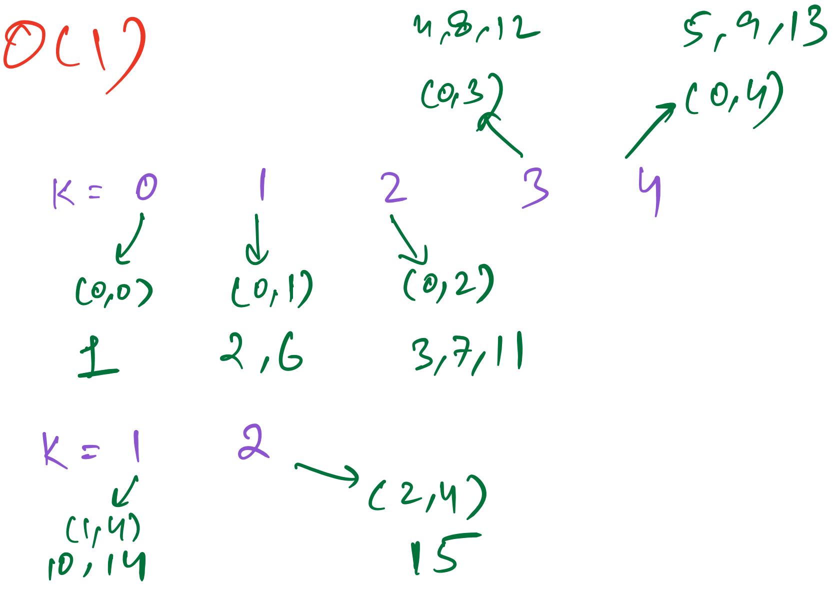
// print all diagonals from m-1th column

```
    for ( K=1;  K<N; ++i)  {
        i=K, j= m-1
        while(i<N && j>=0) {
            print (mat [i][j])
            i++, j--
        }
        print (newline)
    }
```

TC : O(N*M)

SC : O(1)

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



Question 4

Given a mat[N][N], calculate transpose of the matrix without extra space.

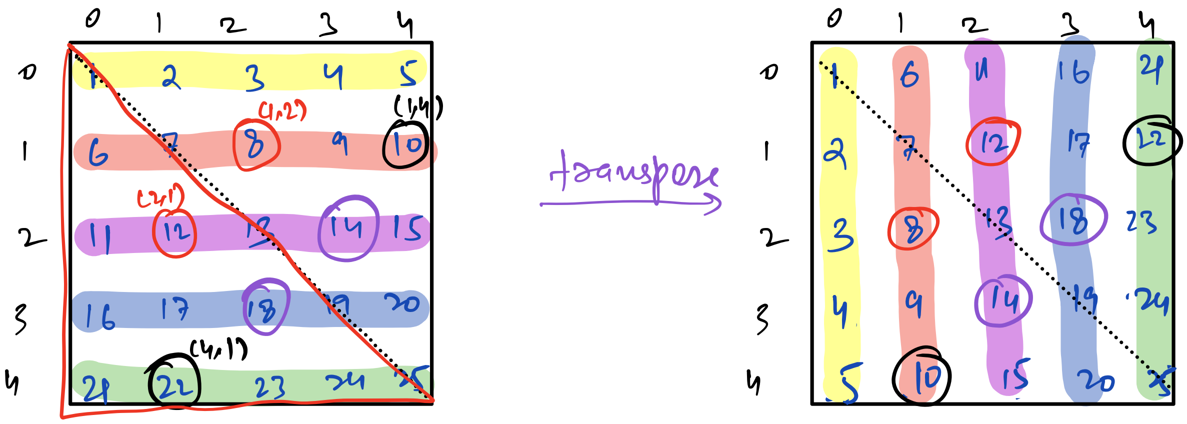
Transpose is :

0th row → 0th column

1st row → 1st column

⋮

n-1th row → n-1th column



$$
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 1 & 2 & 3 & 4 & 5 \\
1 & 6 & 7 & 8 & 9 & 10 \\
2 & 11 & 12 & 13 & 14 & 15 \\
3 & 16 & 17 & 18 & 19 & 20 \\
4 & 21 & 22 & 23 & 24 & 25
\end{array}
\xrightarrow{\text{transpose}}
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 1 & 6 & 11 & 16 & 21 \\
1 & 2 & 7 & 12 & 17 & 22 \\
2 & 3 & 8 & 13 & 18 & 23 \\
3 & 4 & 9 & 14 & 19 & 24 \\
4 & 5 & 10 & 15 & 20 & 25
\end{array}
$$

mat[5][5]

[i,j] → [j,i]

Code

```
for (i=0; i<N; ++i) {
    for (j=0; j<N; ++j) {
        temp = mat[i][j]
        mat[i][j] = mat[j][i]
```

X DONT WORK

```
        mat[j][i] = temp
    }
}
```

i=1 , j=3

swap (mat[1][3] , mat[3][1])

⋮

i=3 , j=1

swap (mat[3][1] , mat[1][3])

we are swapping 2 times

⇓ SOLUTION

either swap when i<j OR i>j

```
for (i=0; i<N; ++i) {
    for (j=i+1; j<N; ++j) {
        temp = mat[i][j]
        mat[i][j] = mat[j][i]
        mat[j][i] = temp
    }
}
```

✓ WORK

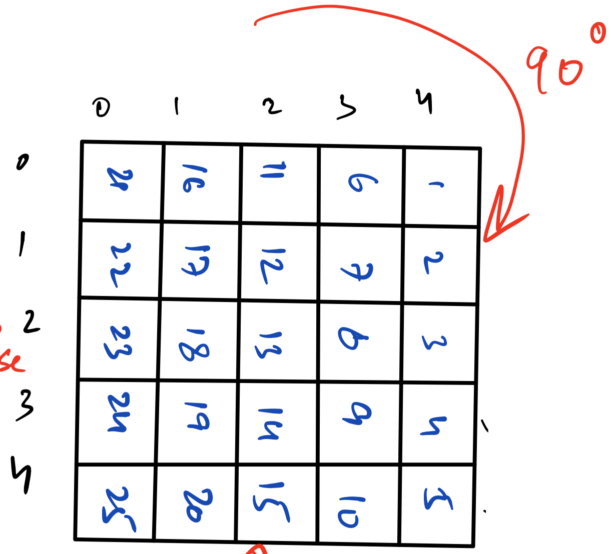
TC : $O(N^2)$

SC : $O(1)$

Question 5

Given square mat[N][N], rotate 90° clockwise from top-right. SC: O(1)

$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 6 & 7 & 8 & 9 & 10 \\ 2 & 11 & 12 & 13 & 14 & 15 \\ 3 & 16 & 17 & 18 & 19 & 20 \\ 4 & 21 & 22 & 23 & 24 & 25 \end{array} \xrightarrow[\text{clockwise}]{90^\circ}$$



$\downarrow$ transpose

$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 1 & 6 & 11 & 16 & 21 \\ 1 & 2 & 7 & 12 & 17 & 22 \\ 2 & 3 & 8 & 13 & 18 & 23 \\ 3 & 4 & 9 & 14 & 19 & 24 \\ 4 & 5 & 10 & 15 & 20 & 25 \end{array} \xrightarrow[\text{each row}]{\text{reverse}} \begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 21 & 16 & 11 & 6 & 1 \\ 1 & 22 & 17 & 12 & 7 & 2 \\ 2 & 23 & 18 & 13 & 8 & 3 \\ 3 & 24 & 19 & 14 & 9 & 4 \\ 4 & 25 & 20 & 15 & 10 & 5 \end{array}$$

90° clockwise rotation => transpose + reverse

TC : $N^2 + N^2$

: $O(N^2)$

SC : O(1)

DOUBT