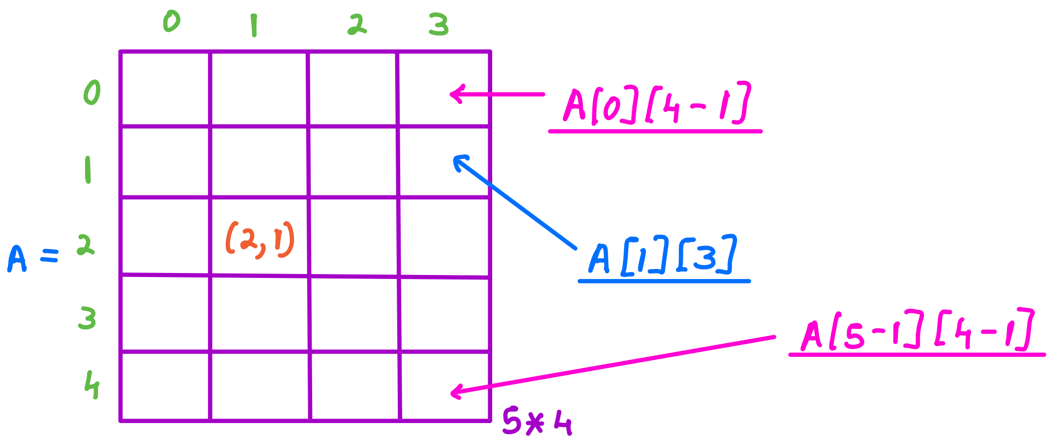


Q1 → Given a 2D matrix, print row wise sum.

$$A = \begin{array}{c|cccc|c} i\backslash j & 0 & 1 & 2 & 3 & \text{o/p} \\ \hline 0 & 1 & 2 & 3 & 4 & \rightarrow 10 \\ 1 & 5 & 6 & 7 & 8 & \rightarrow 26 \\ 2 & 9 & 10 & 11 & 12 & \rightarrow 42 \end{array}$$

```
for i → 0 to (N-1) {
        sum = 0
        for j → 0 to (M-1) {
            sum += A[i][j]
        }
        print (sum)
}
```

TC = O(N*M)

SC = O(1)

Q2 → Given a 2D matrix, print row wise sum.

$$A = \begin{array}{c|cccc} i\backslash j & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 & 8 \\ 2 & 9 & 10 & 11 & 12 \\ \hline \text{o/p} \rightarrow & 15 & 18 & 21 & 24 \end{array}$$

```
for i → 0 to (M-1) {   // i → col
    sum = 0
    for j → 0 to (N-1) {   // j → row
        sum += A[j][i]
    }
    print (sum)
}
```

TC = O(N*M)

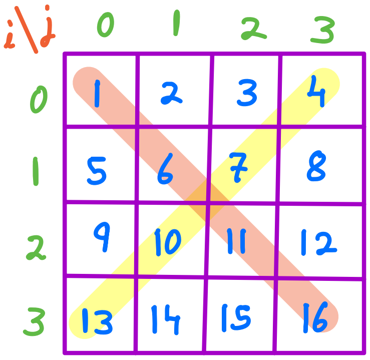
SC = O(1)

Q3 → Given a 2D square matrix. (N == M)

Print the diagonal elements from →

a) top-left to bottom-right

b) top-right to bottom-left



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

```
for i → 0 to (N-1) {
    print (A[i][i])
}
```

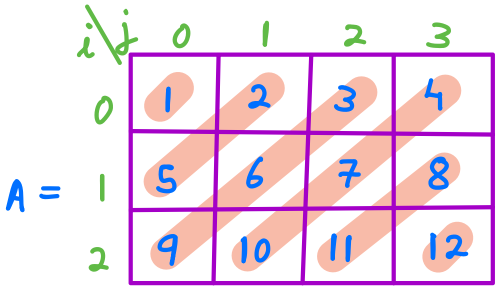
TC = O(N)   SC = O(1)

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

TC = O(N)   SC = O(1)

$\boxed{i + j = N-1 \Rightarrow j = N-1-i}$

Q4 → Given a matrix print all the elements diagonally (right to left).



o/p →

1

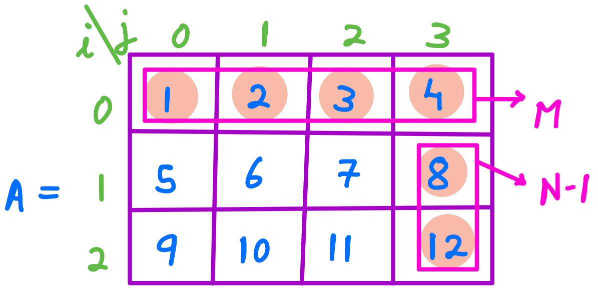
2 5

3 6 9

4 7 10

8 11

12



Total diagonals = M+N-1

```
for i → 0 to (M-1) {
        r=0      c=i
        while (r < N && c >= 0) {
                print (A[r][c])
                r++      c--
        }
        print ("\n")
}
for i → 1 to (N-1) {
        r = i      c = M-1
        while (r < N && c >= 0) {
                print (A[r][c])
                r++      c--
        }
        print ("\n")
}
```

TC = O(N*M)

SC = O(1)

10:10 PM

Q5 → Find transpose of the given square matrix.
(update i/p to transpose)

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 & 8 \\ 2 & 9 & 10 & 11 & 12 \\ 3 & 13 & 14 & 15 & 16 \end{array} \longrightarrow \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 5 & 9 & 13 \\ 1 & 2 & 6 & 10 & 14 \\ 2 & 3 & 7 & 11 & 15 \\ 3 & 4 & 8 & 12 & 16 \end{array}$$

A[2][1] → A[1][2]

A[i][j] → A[j][i]

```
// A → global
void swap (r, c) {   // A[r][c] ⟷ A[c][r]
    t = A[r][c]
    A[r][c] = A[c][r]
    A[c][r] = t
}
```

TC = $O(N^2)$ SC = $O(1)$

```
for i → 1 to (N-1) {   // i > j
    for j → 0 to i-1 {
        swap (i, j)
    }
}
```

i = 1, j = 3
A[1][3] ⟷ A[3][1]

i = 3, j = 1
A[3][1] ⟷ A[1][3]

same as input

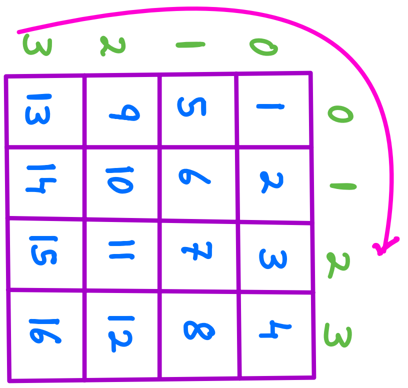
Q6 → Rotate the given square matrix by 90° clockwise. ↻

I/P →

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 & 8 \\ 2 & 9 & 10 & 11 & 12 \\ 3 & 13 & 14 & 15 & 16 \end{array}$$

↓ transpose ✓

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 5 & 9 & 13 \\ 1 & 2 & 6 & 10 & 14 \\ 2 & 3 & 7 & 11 & 15 \\ 3 & 4 & 8 & 12 & 16 \end{array}$$



↓

$$\begin{array}{cccc} 13 & 9 & 5 & 1 \\ 14 & 10 & 6 & 2 \\ 15 & 11 & 7 & 3 \\ 16 & 12 & 8 & 4 \end{array}$$

Sol → 1) Take transpose

```
2) for i → 0 to (N-1) {
       reverse A[i][0] to A[i][N-1]
   }
```

TC = O(N²)   SC = O(1)

Q7 → Print all boundary elements of the matrix clockwise starting from (0,0).

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 & 8 \\ 2 & 9 & 10 & 11 & 12 \\ 3 & 13 & 14 & 15 & 16 \end{array}$$

(Try print inner boundary)

O/P → 1 2 3 4 8 12 16
15 14 13 9 5

```
for i → 0 to (M-2) {
    print (A[0][i])
}
```

$TC = O(2 * (N+M))$

$= \underline{O(N+M)}$

$SC = \underline{O(1)}$

```
for i → 0 to (N-2) {
        print (A[i][M-1])
}
for  i → (M-1) to 1 {
        print ( A[N-1][i])
}
for  i → (N-1) to 1 {
        print(A[i][0])
}
```