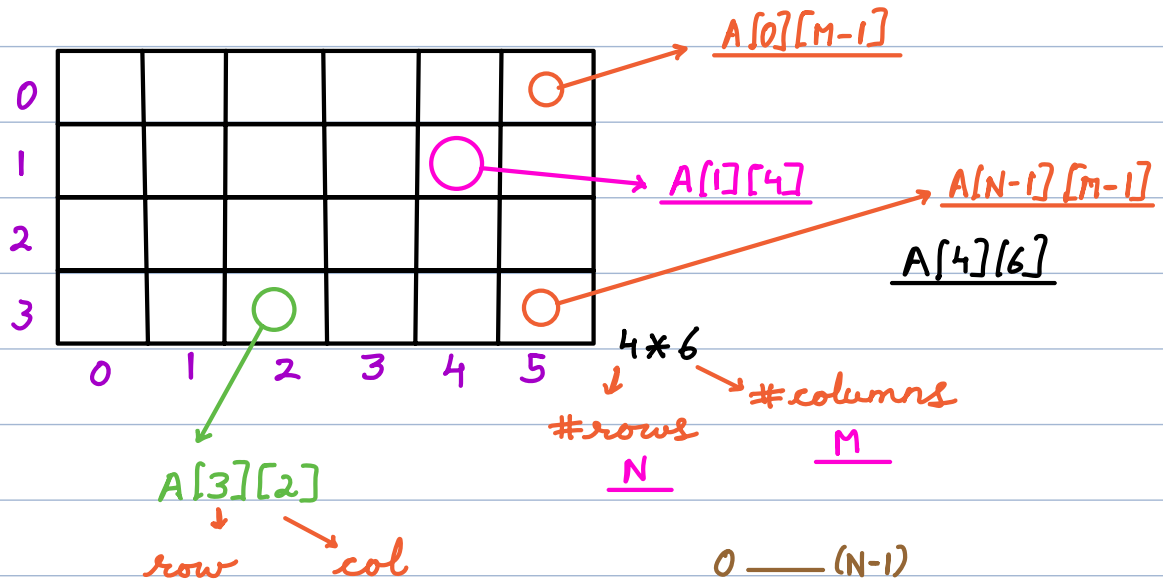
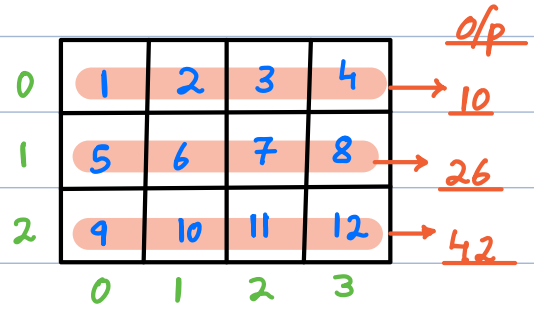


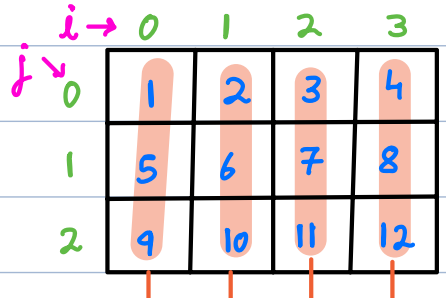
Q → Given a 2D-Matrix, print row wise sum.




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

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

Q → Given a 2D matrix, print column wise sum.



o/p → 15 18 21 24

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

TC = O(N*M)

SC = O(1)

(N=M)

Q → Given a 2d square matrix, print diagonal:

a) from top left to bottom right

b) from top right to bottom left

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

4*4

o/p → 1 6 11 16

i	j
0	0
1	1
2	2
3	3

i == j

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

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

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

(0, N-1)

4*4

o/p → 4 7 10 13

i	j
0	3
1	2
2	1

i++ j--

i + j = N-1

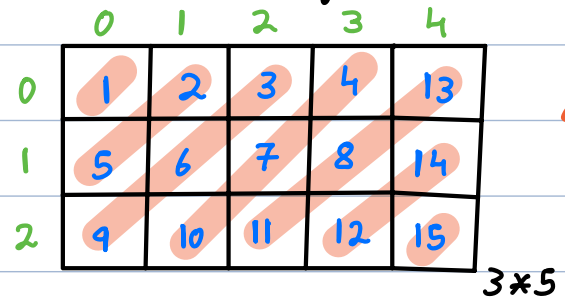
j = N-1-i

3 | 0

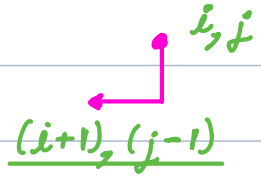
```
for i → 0 to (N-1) {
    j = N-1-i
    print (A[i][j])
}
```

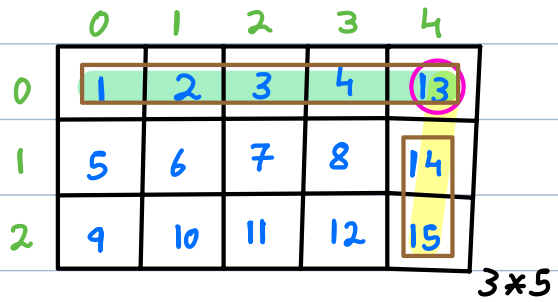
TC = O(N) SC = O(1)

Q → Given a 2d matrix print all diagonals from right to left, starting from (0,0).



o/p → 1 ✓
2 5 ✓
3 6 9 ✓
4 7 10 ✓
13 8 11 ✓
14 12 ✓
15 ✓





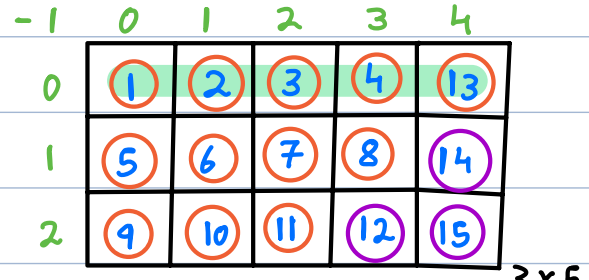
5 + 3 - 1 = 7

diagonals = N + M - 1

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

c = 0 1 2 3 4 5 stop

r = 1 2 3 stop

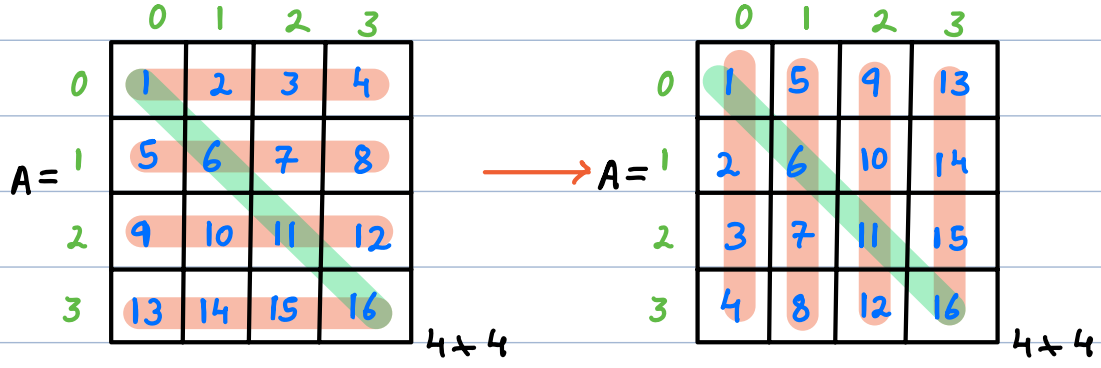


```
}
for r → 1 to (N-1) {   // row, c = M-1
    i = r   j = M-1
    while ( i < N && j >= 0 ) {
        print (A[i][j])
        i++     j--
    }
    print ("\n")
}
```

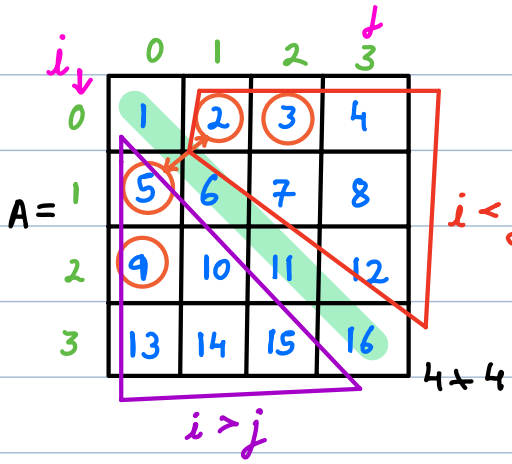
TC = O(N * M) SC = O(1)

Q → Given a **square matrix**,
convert the given matrix to it **transpose**.

row ↔ col



SC = O(1)



(i, j) ⟷ (j, i)

(row, col) ⟷ (col, row)

i j
(0, 1) ⟷ (1, 0)
(1, 0) ⟷ (0, 1)

```
for i → 0 to (N-2) {
    for j → (i+1) to (N-1) {
        swap (A[i][j] , A[j][i])
    }
}
```

}

}

TC $\frac{N^2 - N}{2} \rightarrow O(N^2)$

SC = O(1)

Q → Rotate the given square matrix 90° clockwise.

A =

	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

reverse every row

A =

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

Sol → 1) transpose ✓

2) reverse every row ✓

TC = O(N²) SC = O(1)
