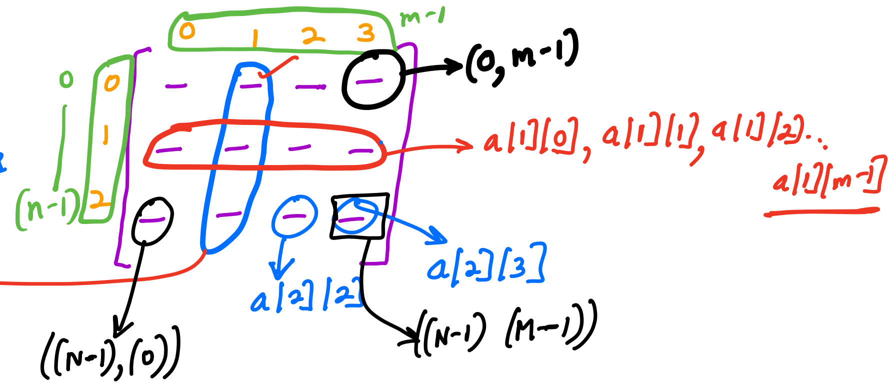
int a[3][4];

↑ row   ↑ columns



a[0][1],
a[1][1],
a[2][1]
⋮
a[n-1][1]

Q. Given a matrix a[n][m]. Print row wise sum.

Eg

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 3 & 8 & 9 & 2 \\ 1 & 1 & 2 & 3 & 6 \\ 2 & 4 & 10 & 11 & 13 \end{array}$$

3*4

⇒ 3+8+9+2 = 22
⇒ 1+2+3+6 = 12
⇒ 4+10+11+13 = 38

o/p

i = ~~0~~ 1
j = 0
sum = 0

n = 3
m = 4

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

```
for (i = 0 ; i < n ; i++) {
    sum = 0;
    for (j = 0 ; j < m ; j++) {
        sum = sum + a[i][j];
    }
    print (sum);
}
```

Q→ Given a matrix a[n][m], find max column wise sum

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 3 & 8 & 9 & 2 \\ 1 & 1 & 2 & 3 & 6 \\ 2 & 4 & 10 & 11 & 13 \end{array}$$

Eg

3	8	9	2
+1	+2	+3	+6
+4	+10	+11	+13
8	20	23	21

Ans = 23

a[row] [col] ✓

```
maxSum = INT_MIN;
for (i=0 ; i < m ; i++) {        → column ✓
    sum = 0;
    for (j = 0 ; j < n ; j++) {   → row ✓
        sum = sum + a[j][i];
    }
    maxSum = max(sum, maxSum);
}
return maxSum;
```

$TC = O(N \times M)$

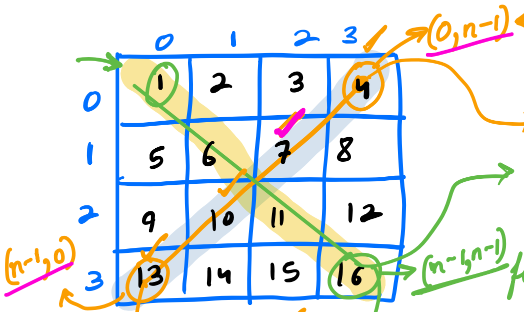
$SC = O(1)$

Q→ Given a square matrix a[n][n]. Print diagonals.

⇒ $\boxed{n = m}$ ✓

✓ Top Right → Bottom left ✓

✓ Top left → Bottom Right



4, 7, 10, 13

1, 6, 11, 16

a[0][0] ✓
a[1][1] ✓
a[2][2] ✓
a[3][3] ✓

i = 0, 1, 2, 3, 4

```
for (i=0 ; i < n ; i++) {
    print (a[i] [i]);
    (i = j)
}
```

$TC = O(N)$ ✓

$SC = O(1)$

→ (0, 3) ← (0, n-1) ✓
→ (1, 2) ✓
→ (2, 1)
→ (3, 0)

n = 4
n-1 = 3

i++
j--

$\boxed{i + j = n-1}$ → $\boxed{j = (n-1) - i}$

(i + j = 3)

(0,0) ✓
(1,1)
(2,2)
(3,3) ✓
(i == j) ✓
a[i][i]

```
for (i = 0 ; i < n ; i++) {
    → j = n-1-i ; ✓
    print (a[i] [j]); ✓
}
```

a[i] [n-1-i] ✓

$TC = O(N)$ ✓

$SC = O(1)$

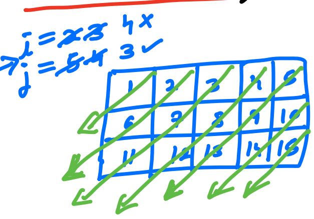
(0, n-1) → (1, n-2) → (2, n-3) - - - - → ((n-1), 0) ←

0 + n-1 = n-1
1 + n-2 = n-1
2 + n-3 = n-1
⋮
n-1 + 0 = n-1

$\boxed{i + j = n-1}$ ✓ → $\boxed{j = (n-1) - i}$ ✓

Q→ Given a matrix a[n][m]. Print all diagonals from right to left.

4+6−1 = 9



TC = O(N*M)

i++
j−−

Diagonal can start from
- ∀ elements in first row = M
- ∀ elements in last column = N

Total starting positions? (N+M−1)

(0,0) (0,1) (0,2) - - - (0, m−1), (1, m−1), (2, m−1) - - - - (n−1, m−1)

M

```
for (col=0 ; col < m ; col++) {
    i=0;  j = col;
    while ( i < n && j >= 0) {
        print(a[i][j]);
        i++;   j--;
    }
    print("\n");
}
```

N

N*M

a[r][c]

col = 2   n = 4   m = 6

i = ~~0~~ 1 ~~2~~ 3   j = ~~2~~ 1 ~~0~~ −1

⇒ a[0][2] = 3

⇒ a[1][1] = 8

⇒ a[2][0] = 13

TC = O(N*M)

SC = O(1)

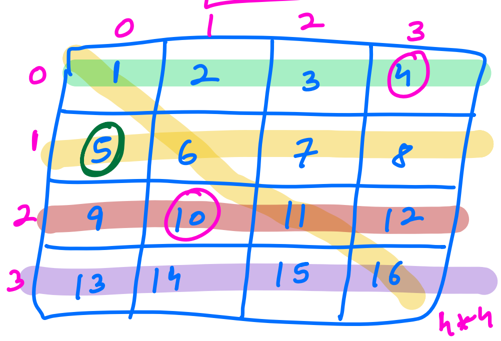
```
N → for( row=1; row<n; row++){
        i = row;     j = m−1;
        while ( i < n && j >= 0) {
            print (a[i][j]);
            i++;   j--;
        }
        print("\n");
    }
```

M

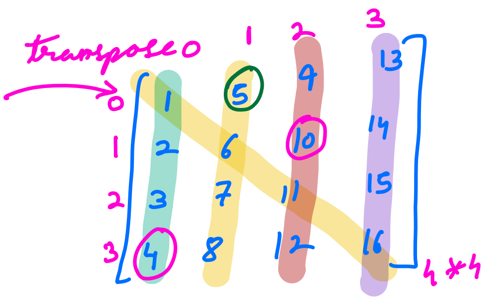
N*M

Q→ Given a square matrix. Convert the matrix to its transpose.
No extra space is allowed ⇒ $SC = O(1)$ ✓

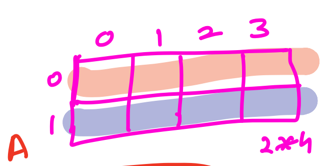
$n = m$

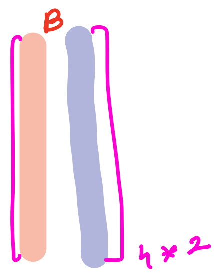


transpose



(1,0) → (0,1) ✓
(0,3) → (3,0) ✓
(2,1) → (1,2) ✓
⋮
$\boxed{(i,j) \to (j,i)}$ ← ✓





B[j][i] = A[i][j]

```
for ( i = 0 ; i < n; i++) {
  ✓ for ( j = 0; j < i; j++) {
      // swap (a[i][j], a[j][i]);   ✓
  }
}
```

j < i

(0,2)
i=0 j=2
a[0][2] ⇄ a[2][0] ✓

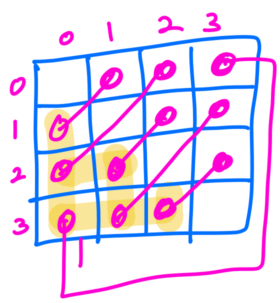
i=2 j=0
a[2][0] ⇄ a[0][2] ✓

i = ~~0~~ j = 0
~~1~~
~~2~~
~~3~~
4 ✓

$1 + 2 + 3 + \dots N$

$\frac{N(N+1)}{2}$ $TC = O(N^2)$

✓ a[1][0] ⇄ a[0][1] ✓
a[2][0] ⇄ a[0][2]
a[2][1] ⇄ a[1][2]
a[3][0] ⇄ a[0][3]
a[3][1] ⇄ a[1][3]
a[3][2] ⇄ a[2][3]



$SC = O(1)$

Q→ Rotate a square matrix by 90° clockwise.

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 & 8 \\ 2 & 9 & 10 & 11 & 12 \\ 3 & 13 & 14 & 15 & 16 \end{array} \xrightarrow{90^\circ \text{ clockwise}} \begin{bmatrix} 13 & 9 & 5 & 1 \\ 14 & 10 & 6 & 2 \\ 15 & 11 & 7 & 3 \\ 16 & 12 & 8 & 4 \end{bmatrix}$$

↓ transpose

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

✓ 1) Find Transpose → $TC = O(N * N) = O(N^2)$

✓ 2) Reverse every row. → $TC = O(N * N) = O(N^2)$

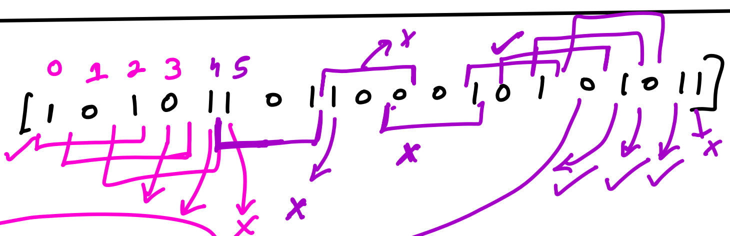
swap

[1 | 2 | 3 | 4]

swap

[1 | 2 | 3 | 4 | - - - | N]

swap (1, N)
(2, N-1)
(3, N-2)
⋮
(N/2, $N - \frac{N}{2}$)



→ a[i] != a[i-1]

B = 1

$\boxed{2 * B + 1 \le N}$

$2*1+1 = \boxed{3}$ ✓

count valid center

→ count alternating subarray of length 2*B+1.

ans = ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9

st = end − 2*B

st end $\frac{st+end}{2}$ ✓

[-5 0]

ans = 0 ✓