Welcome ☺

Agenda : 2D arrays.

Q Given a row-wise & column wise sorted matrix, find out whether element k is present or not.

$$A: \begin{bmatrix} -5 & -2 & 1 & 13 \\ -4 & 0 & 3 & 14 \\ -3 & 2 & 6 & 18 \end{bmatrix}$$

Bruteforce

⇒ Iterate over all the rows and columns

T.C ⇒ O(N*M)

$$A: \begin{matrix} 0 \\ 1 \\ 2 \end{matrix} \begin{bmatrix} -5 & -2 & 1 & 13 \\ -4 & 0 & 3 & 14 \\ -3 & 2 & 6 & 18 \end{bmatrix} \quad \begin{matrix} 0 & 1 & 2 & 3 \end{matrix}$$

✗ TL → Top left

✗ BR → Bottom Right

✓ TR → Top Right

✓ BL → Bottom Left

$$A: \begin{bmatrix} -5 & -2 & 1 & 13 \\ -4 & 0 & 3 & 14 \\ -3 & 2 & 6 & 18 \end{bmatrix}$$

k = 0

Code

```
int i = 0        int j = M-1   // Top right
while ( i<N && j>=0 )   // within range of matrix
{
    if ( arr[i][j] == K )
        return true.

    else if ( arr[i][j] < K )
        i++          // move down

    else
        j--          // move left

}
return false.
```

T.C ⇒ O(N+M)

S.C ⇒ O(1)

A:

-5	-2	1	13
-4	0	3	14
-3	2	6	18

K = 4

13	4 < 13	left
1	4 > 1	down
3	4 > 3	down
6	4 < 6	left
2	4 > 2	down

false

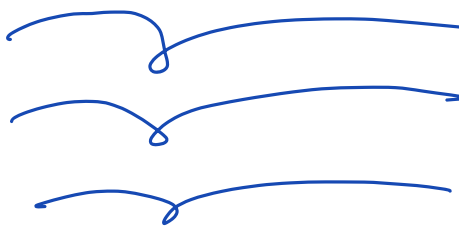
Q Print Boundary Elements N*N matrix clockwise direction

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

o/p ⇒ 1 2 3 4 5 10 15 20 25 24 23 22 21 16 11 6

Approach

Print N-1 elements of 0th row from left to right



last column from top to down

last row from right to left

first column from down to top.

T.C ⇒ O(N)

Code

```
printBoundary ( A[][] , N )
{
    i=0   j=0
    for ( idx = 1 ; idx < N ; idx++)   // first row N-1 ele
    {
        print ( A[i][j] )
        j++
    }

    for ( idx = 1 ; idx < N ; idx++)
    {
        print ( A[i][j] )
        i++
    }
```

```
for ( idx = 1 ; idx < N ; idx ++)
{
    print ( A[i][j] )
    j--
}

for ( idx = 1 ; idx < N ; idx ++)
{
    print ( A[i][j] )
    i--
}
}
```

Q Spiral Matrix.

Given a integer A, generate a square matrix filled with elements from 1 to A^2 in spiral order and return generated square matrix.

N = 4

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

1	2	3
8	9	4
7	6	5

```
Boundary (ans[][], N)
{
  i=0   j=0   count = 1
  while (N>1)
  {
      for (idx = 1 ; idx < N ; idx++)
      {
          ans[i][j] = count
          j++ , count++
      }

      for (idx = 1 ; idx < N ; idx++)
      {
          ans[i][j] = count
          i++ , count++
      }

      for (idx = 1 ; idx < N ; idx++)
      {
          ans[i][j] = count
          j-- , count++
      }

      for (idx = 1 ; idx < N ; idx++)
      {
          ans[i][j] = count
          i-- , count++
      }
      i++   j++     N = N-2
  }
  if (N==1)
  {
      ans[i][j] = count
  }
```

T.C = $O(N^2)$

S.C = $O(1)$

Submatrix

↳ contigous part of a matrix

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

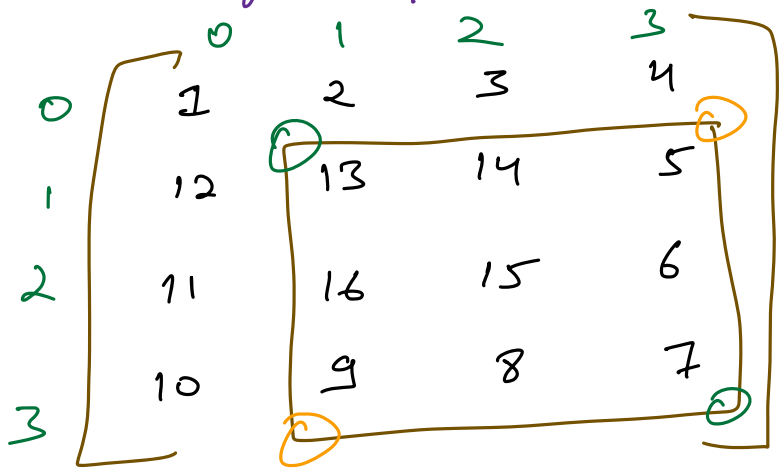
$$\begin{bmatrix} 15 & 6 \\ 8 & 7 \end{bmatrix}$$ ✓

$$\begin{bmatrix} 1 & 3 \\ 12 & 14 \end{bmatrix}$$ ✗

How to identify a sub-matrix.

Submatrix

↳ contiguous part of matrix.



$\begin{bmatrix} 15 & 6 \\ 8 & 7 \end{bmatrix}$ ✓

$\begin{bmatrix} 1 & 3 \\ 12 & 14 \end{bmatrix}$ ✗

Submatrix ⇒ T.L , B.R

T.R , B.L

Q Sum of all submatrices. sum.

$\begin{bmatrix} 4 & 9 & 6 \\ 5 & -1 & 2 \end{bmatrix}$

[4] → 4
[9] → 9
[6] → 6
[5] → 5
[-1] → -1
[2] → 2

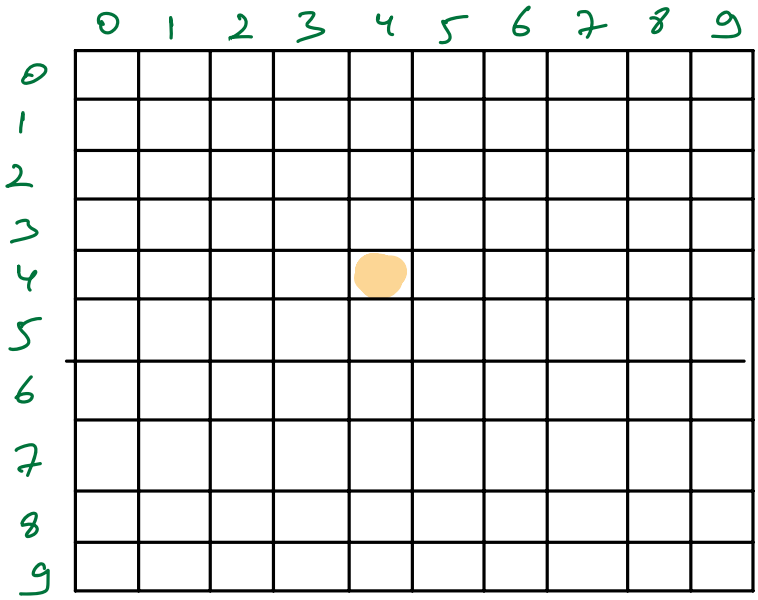
[4, 9] ⇒ 13
[9, 6] ⇒ 15
[5, -1] ⇒ 4
[-1, 2] ⇒ 1

[4 9 6] ⇒ 19
[5 -1 2] ⇒ 6

$\begin{bmatrix} 4 & 9 \\ 5 & -1 \end{bmatrix}$ ⇒ 17 $\begin{bmatrix} 9 & 6 \\ -1 & 2 \end{bmatrix}$ = 16

$\begin{bmatrix} 4 \\ 5 \end{bmatrix}$ ⇒ 9 $\begin{bmatrix} 9 \\ -1 \end{bmatrix}$ = 8 $\begin{bmatrix} 6 \\ 2 \end{bmatrix}$ ⇒ 8 $\begin{bmatrix} 4 & 9 & 6 \\ 5 & -1 & 2 \end{bmatrix}$ ⇒ 25

= 166



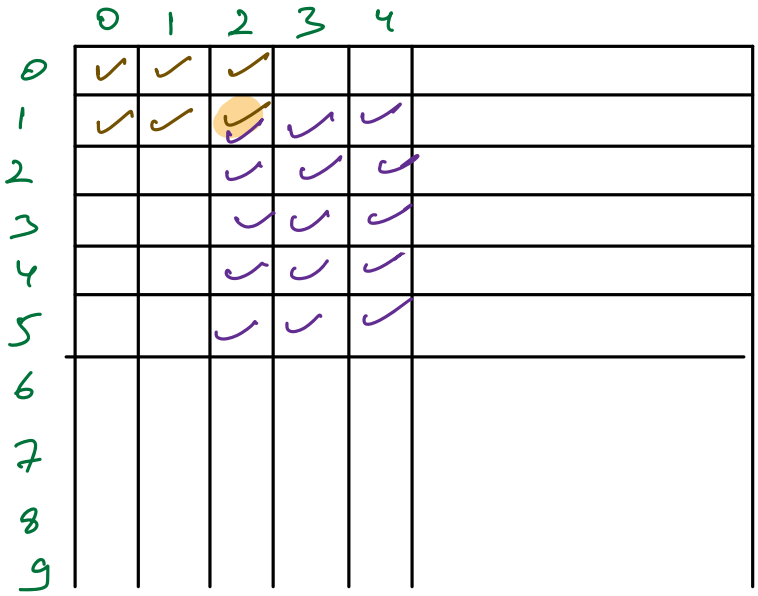
T.L ⇒

B.R ⇒

Starting point = $(i+1) * (j+1)$

Ending point = $(N-i) * (m-j)$

submatrices ⇒ $\left((i+1) * (j+1)\right) * \left((N-i) * (m-j)\right)$



$6 * 15 = 90$

Code

```
totalSum = 0
for( int i=0 ; i< N ; i++)
{
    for( int j = 0 ; j<M ; j++)
    {
        topLeft = (i+1) * (j+1)
        bottomRight = (N-i) * (M-j)
        contribu^n = topLeft * bottomRight * A[i][j]
        totalSum += contribu^n
    }
}
```

T.C = O(N * M)

S.C = O(1)