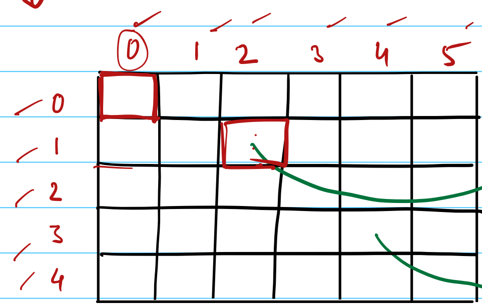
2D Matrices

→ int mat[][] = new int [5][6]

rows (horizontal lines)

columns (vertical lines)

mat[0][0]



mat[1][2]

mat[3][4]

[0, N-1]

int mat[N][M]

[0, M-1]

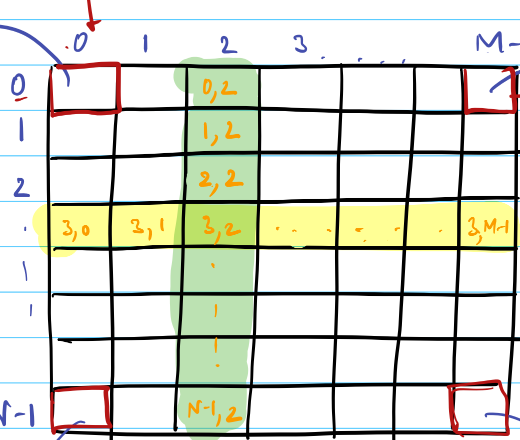
rows columns

TL

mat[0][0]

TR

mat[0][M-1]



BR

mat[N-1][M-1]

BL

mat[N-1][0]

N → Rows   M → column

$\boxed{N*M}$

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

Example
mat[3][4]

3	8	9	2
1	2	3	6
4	10	11	17

Output:
22
12
42

```
for (i=0; i<N; i++)
    sum = 0
    for (j=0; j<M; j++)
        sum += mat[i][j];
    print sum
```

mat[0][0] + mat[0][1] + mat[0][2] + mat[0][3]

→ $\boxed{O(N*M)}$

N=1   M = 10^5 N^2

b) Column-wise sum?

mat[0][0] + mat[1][0] + mat[2][0]

0,0
1,0
2,0

0,1
1,1
2,1

col

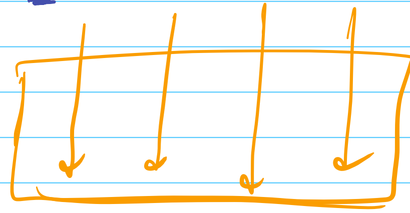
```
for (i=0; i<M; i++)
    sum=0
    for (j=0; j<N; j++)
        sum += mat[j][i]
    print sum
```

i=0
j: 0 → N-1
[0][0]
[1][0]

TC → N×M

Sc → O(1)

N × M <=> mat[N][M]
↓ Rows   ↓ Colu



Q Given a mat [N][N] . Print diagnols.

square matrix

	0	1	2	3	4
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

1 7 13 19 25

(0,0) (1,1)
(2,2) (3,3) (4,4)

i == j

```
for ( K = 0 ; K < N ; K++ )
    print( mat[K][K] )
```

(4,0) (3,1) (2,2) (1,3) (0,4)

a + b = K

?

N-1

i (n-i-1)

n-1

```
for ( c = 0 ; c < N ; c++ )
    r = n-1-c
    print ( mat [r] [c] )
```

4,0
3,1
2,2
1,3
0,4

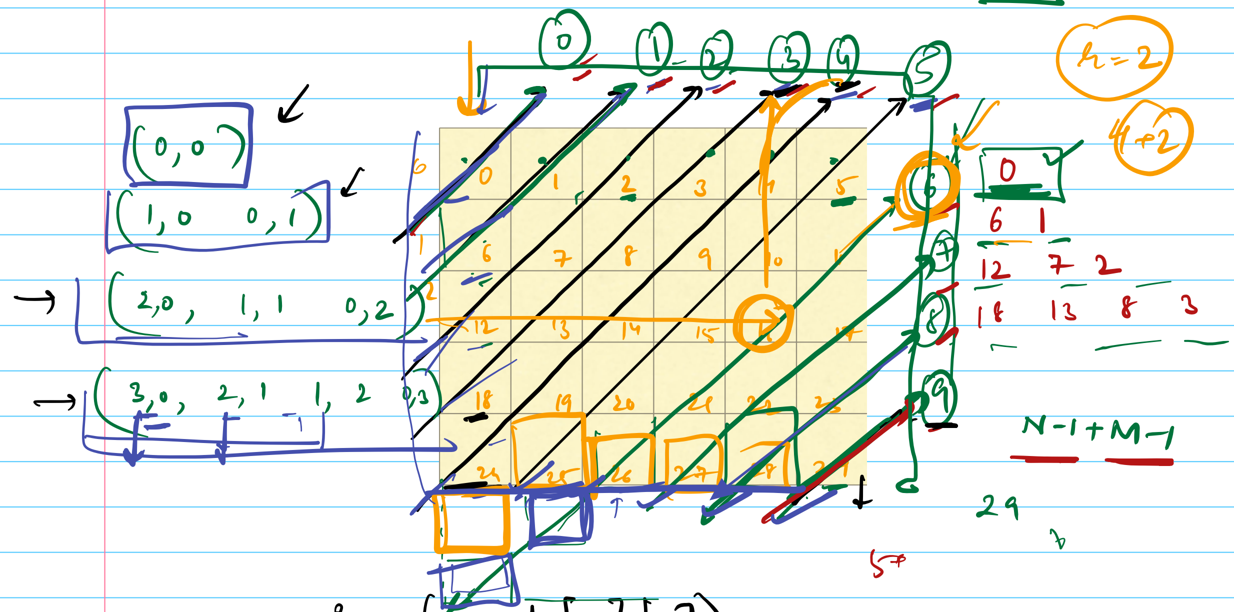
r + c = constant

c = sum - r

Rectangular Matrix !

(L→R)

Q Given a matrix [N][M]. Print all diagonals



print (mat [0][0])

print(mat [1][0]) print (mat [0][1]

Observation

decrement row increment col.

	starts	end	
0	0,0	(0,0)	→ 0
1	1,0	0,1	→ 1
2	2,0	0,2	→ 2
3	(3,0)	0,3	→ 3
4	4,0	0,4	→ 4
5	(4,1) (3,2) ...	0,5	→ 5

(5,0) ?

out of bounds

~~6,0~~   ~~5,1~~   (4,2)

X   X

NOTE: To print a diagnol, you decrement row and increment col

$r \quad c$

$\downarrow \quad \downarrow$

$r-1 \quad c+1$

$r+c$

$\boxed{\begin{array}{ll} r >= 0 & c >= 0 \\ r < N(5) & c < M(6) \end{array}}$

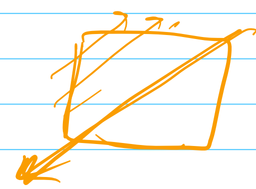
clear ← $\boxed{sum = 0}$

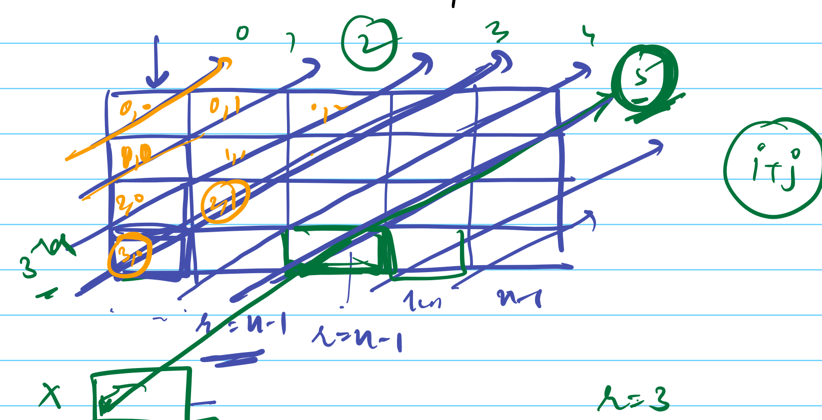
$\boxed{r = 0 \qquad c = 0}$ print

← $\boxed{sum = 1}$

$r = 1 \qquad c = 0$ print.

$r = 0 \qquad c = 1$ print





X $\boxed{r = sum}$

5th index

$r = 3$

$5 < 3$

$r = 3$

r

$\boxed{\begin{array}{c} sum > r \\ \downarrow \\ r = n-1 \end{array}}$

Iterates over all diagonals, sum = r+c

r+c = sum
c = sum - r

```
for ( sum = 0 ; sum <= N+M-2 ; sum++ )
    ( r = sum ; r >= 0 ; r-- )
        ( c = sum - r
        if ( r < 0 or c < 0 or r >= N
             or c >= M )
            continue ;
        print ( mat [r][c] )
```

(c, r)

r + c = sum

sum = 5

r = min(n-1 , sum)

sum → r + c

r → sum, c → 0

Observations:

→ Diagonals start from $\boxed{\text{col index} = 0}$

→ Elements index sum (r + c) is constant across a diagonal.

→ Diagonal sum increases by 1 as we move on to next diagonal

→ Kadanes
→ Sliding Window (2 pointers

Alternative subarray

Q. Given a mat[N][N] (square matrix). Find transpose matrix inplace (SC: O(1))

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

$(i, j) \rightarrow j, i$

$r = 0 \rightarrow n-1$

$c = 0 \rightarrow r$

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

output

$\boxed{i == j}$

4, 0

(4, 1)

TC = N^2

SC O(1)

swap (1, 4) with (4, 1)

```
for ( r = 0 ; r < n ; r++ )
    for (c = r ; c < n ; c++ )
        tmp = mat[r][c]
        mat[r][c] = mat[c][r]
        mat[c][r] = tmp
```

Q

Rotate a matrix 90° clockwise (N×N)

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

0,4

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

1,4

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

Steps

1. Transpose
2. Reverse every Row

4 x 3

	0	1	2
0	5	6	1
1	2	3	4
2	1	0	-5
3	2	1	5

0, 0	0, 1	0, 2
1, 0	1, 1	1, 2
2, 0	2, 1	2, 2
3, 0	3, 1	3, 2

1st for

```
for(j=0; j<M; j++)
    sum=0
    for ( i = 0 ; i < N ; i++)
        sum += mat[i][j]
    print(sum)
```