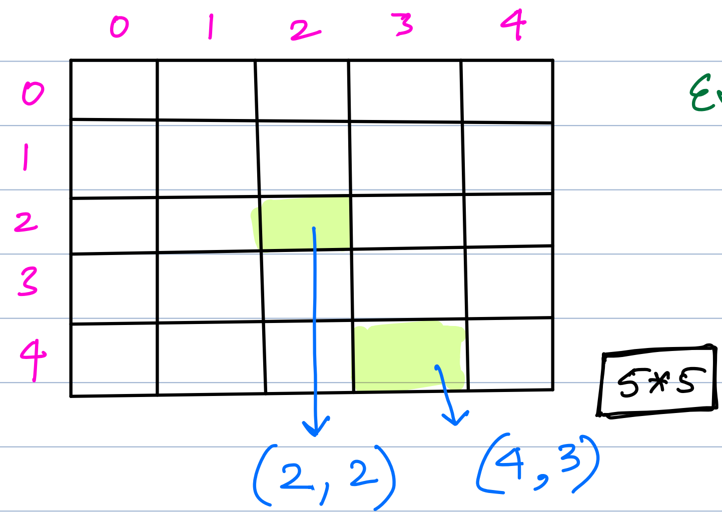
google sheets / calendar

2-D dimensional

↓

rows & columns

Arrays
- prefix sum
- carry forward
- subarrays
  - ↳ contribution




Every cell can be represented by (row, col)

no of cells →

5 * 5 = 25

# cells = no of rows * no of columns

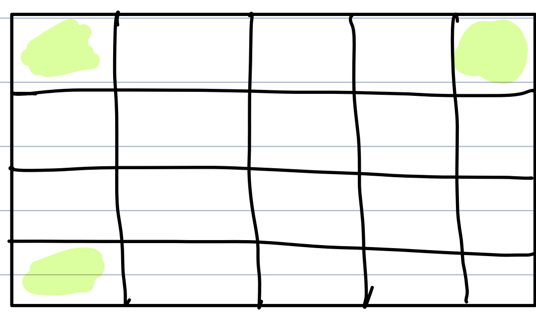
```
int mat[4][5];
```

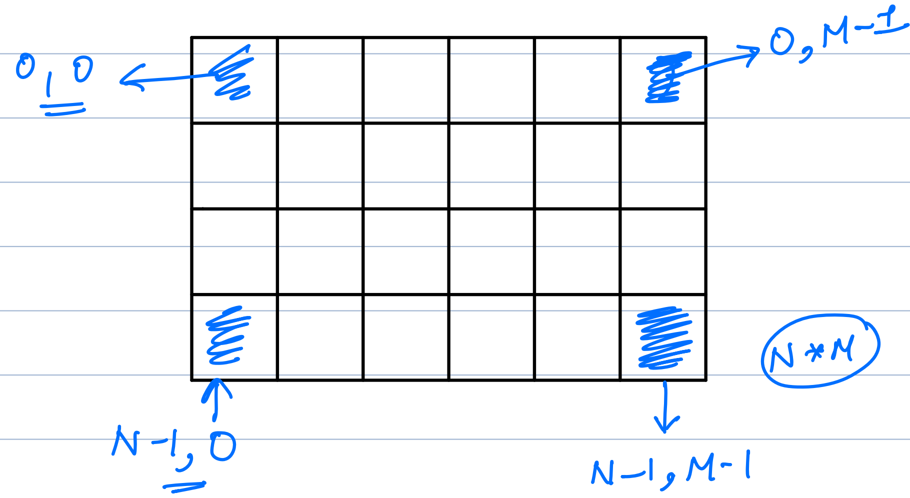
top left cell (0,0)

top right (0, 4)

bottom left (3,0)

bottom right (3,4)

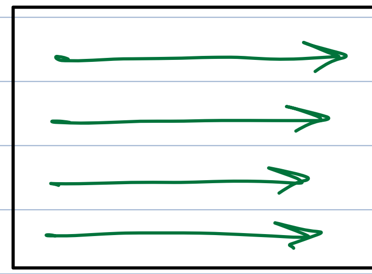





**Q** given a 2D matrix N*M. Print the elements row by row

3*4

| | | | |
|---|---|---|---|
| 0,0 | 0,1 | 0,2 | 0,3 |
| 1,0 | 1,1 | 1,2 | 1,3 |
| 2,0 | 2,1 | 2,2 | 2,3 |



```
for( i=0; i<n; i++)
{          ith row
    for( j=0; j<m; j++)
    {
        // i,j
        print(arr[i][j]);
    }
}
```

sum of each row

```
for( i=0; i<n; i++)
{          ith row     sum=0;
     for( j=0; j<m; j++)
     {
           // i,j
                 sum += arr[i][j];
     }
     print(sum);
}
```

- print sum column-wise

| | | |
|---|---|---|
| 3 | -2 | 1 |
| 8 | 6 | 5 |
| 1 | 2 | 4 |
| ⇓ | ⇓ | ⇓ |
| 12 | 6 | 10 |

```
for( j=0; j<m; j++)      → fix the column
{
              sum=0;     → fix the row
     for( int i=0; i<n; i++)
     {
           sum += arr[i][j];
     }
}
```

Q: Given 2 matrices A and B. Add 2 matrices.

$A$: $N*M$, $B$: $N*M$

$$A = \begin{matrix} 4 & 1 & 3 \\ 1 & 2 & 6 \\ 9 & 1 & 7 \end{matrix} \qquad B = \begin{matrix} 2 & 1 & 3 \\ 4 & 1 & 2 \\ 9 & 2 & 1 \end{matrix}$$

$A = N*M$
$3*4$

$B = M*N$
$4*3$

$$\boxed{res[i][j] = A[i][j] + B[i][j]}$$

$$\boxed{\begin{matrix} 6 & 2 & 6 \\ 5 & 3 & 8 \\ 18 & 3 & 8 \end{matrix}}$$

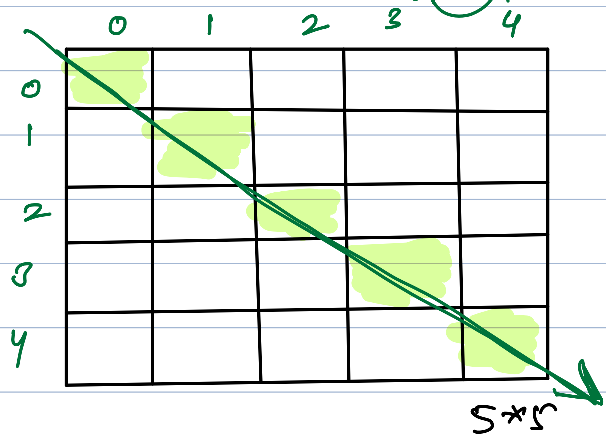
to perform add^n of matrices, dimensions have to be exactly same

T.C: $O(n*m)$
S.C: $O(n*m)$

Q. N * N (Square) matrix. Print the diagonal elements.

( top left, bottom right)

$i = j$




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

arr[i][i]

```
for (i=0; i<n; i++)
    print ( arr[i][i] );
```

$$\boxed{\begin{array}{l} T.C : O(N) \\ S.C : O(1) \end{array}}$$

N * N

top right → bottom left

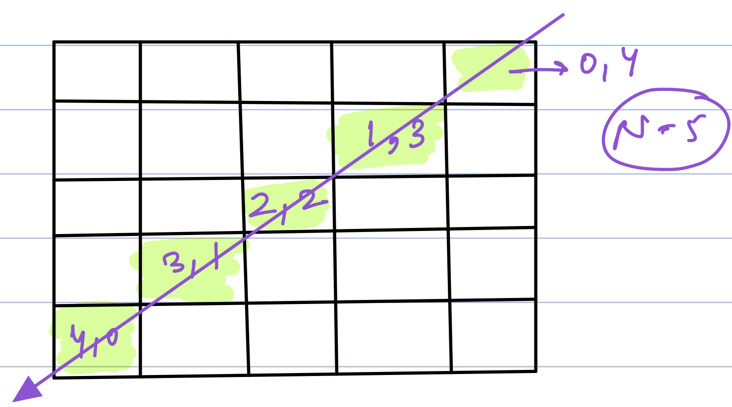
$\boxed{0, n-1}$ → i, j

↓

i+1, j-1

$i = 0, \; j = N-1$

```
while ( i<N && j>=0)
{       print(arr[i][j]);
        i++;
        j--;
}
```




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

$i + j = N-1$

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

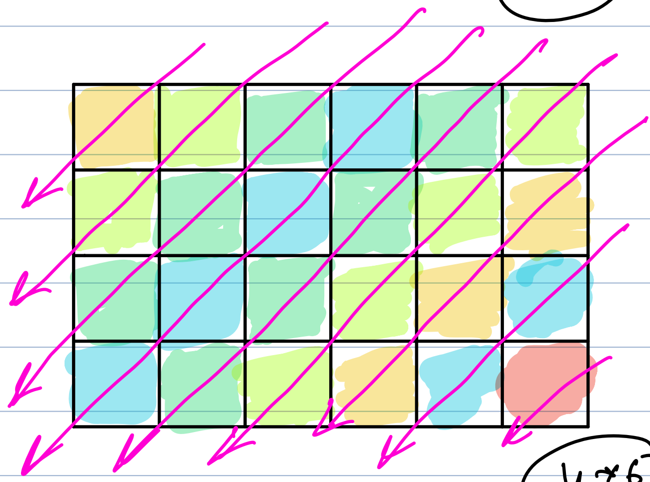
```
for( i=0; i<n; i++)
    pr(arr[i][N-i-1])
```

# Rectangular Matrix

$n != m$

$4 * 5$

$4 * 5 \equiv$ 8 diagonals



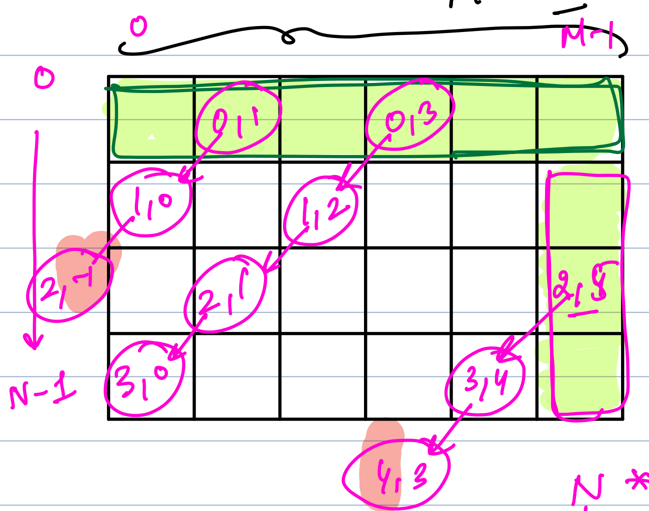
$4 * 6$

total diagonal = 9

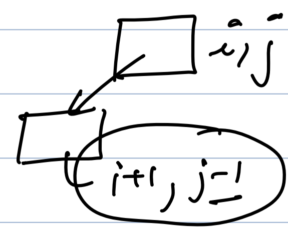
rows

$N * M$ col

M elements




N-1 elements




$N * M$

rows cols

diagonals → $\boxed{M + N - 1}$

$s_i, s_j$ → $s_i + 1, s_j - 1$

$s_i < n$ && $s_j >= 0$

```
printdiagonal( int si , int sj, int n, int m, int arr)
{
        while( si < n && sj >=0)
        {
            pri (arr[si][sj]) ;
                    si++;
                    sj--;
        }
}

for( j=0; j<M; j++)
{
        printdiagonal( 0, j ...)
}
for( i = 1; i<n ; i++)
{
        printdiagonal ( i, m-1 ..)
}
```

T.C: O(N*M)

10:30

• square matrix (N*N). Find $\boxed{\text{transpose}}$ of a matrix

| | | |
|---|---|---|
| 1 | 4 | 5 |
| 6 | 3 | 2 (1,2) |
| 9 | 8 | 0 |

$\Longrightarrow$

| | | |
|---|---|---|
| 1 | 6 | 9 |
| 4 | 3 | 8 |
| 5 | 2 (2,1) | 0 |

$0^{th}$ row → $0^{th}$ col

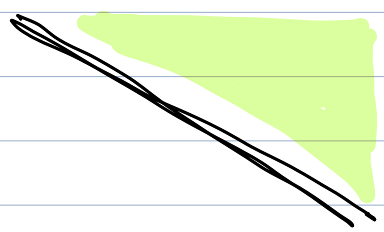
$1^{st}$ row → $1^{st}$ col

$2^{nd}$ row → $2^{nd}$ col

row — col

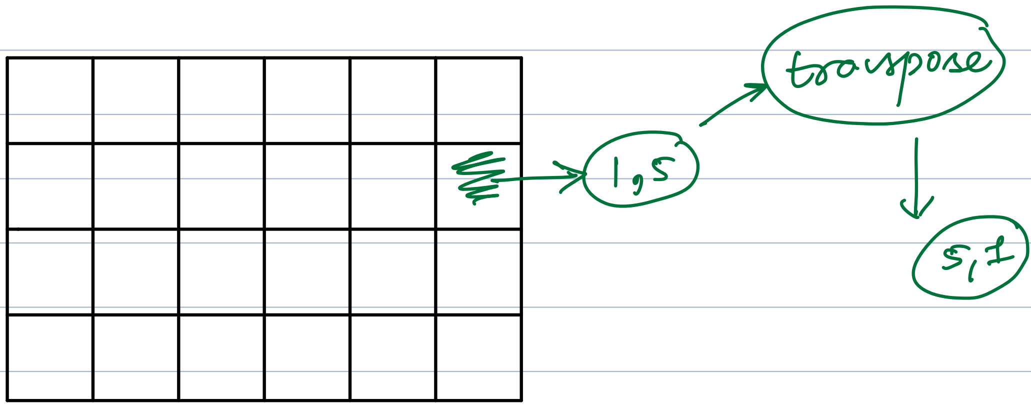
col → row

$\boxed{i, j \Longrightarrow j, i}$



```
for ( i=0 → n )
{
    for ( j= i+1 ; j <n ; j++ )
        swap(arr[i][j],
             arr[j][i]);
```

$\boxed{\begin{array}{l} T.C: O(N^2) \\ S.C: O(1) \end{array}}$




n*m → ~~transpose~~ → m*n

3*4 4*3

we can't do direct swapping.

| | | | |
|---|---|---|---|
| 4 | 1 | 3 | 9 |
| 8 | 2 | 6 | 7 |

A  2*4

| | |
|---|---|
| 4 | 8 |
| 1 | 2 |
| 3 | 6 |
| 9 | 7 |

res

res[j][i] ← A[i][j]

T.C: O(n*m)
S.C: O(n*m)

Rotate a matrix by $90^\circ$ clockwise!

$0^{th}$ row $\equiv 0^{th}$ col

$0^{th}$ row $\equiv n-1$ col

$$\begin{bmatrix} 1 & 4 & 5 & 9 \\ 8 & 11 & 13 & 12 \\ 16 & 3 & 0 & -1 \\ 18 & 15 & 2 & 3 \end{bmatrix} \Longrightarrow \begin{bmatrix} 18 & 16 & 8 & 1 \\ 15 & 3 & 11 & 4 \\ 2 & 0 & 13 & 5 \\ 3 & -1 & 12 & 9 \end{bmatrix}$$

$90^\circ$ clockwise

$\Downarrow$ transpose

$$\begin{matrix} 1 & 8 & 16 & 18 \\ 4 & 11 & 3 & 15 \\ 5 & 13 & 0 & 2 \\ 9 & 12 & -1 & 3 \end{matrix}$$

reverse each row

| T.C: $O(N*M)$ |
|---|
| S.C: $O(N*M)$ |

$\Downarrow$ reverse columns

$$\begin{bmatrix} 9 & 12 & -1 & 3 \\ 5 & 13 & 0 & 2 \\ 4 & 11 & 3 & 15 \\ 1 & 8 & 16 & 18 \end{bmatrix}$$

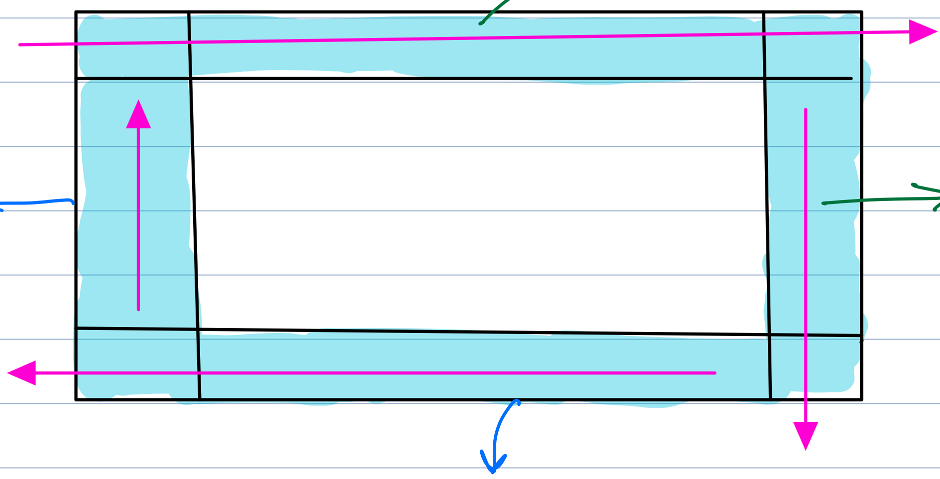
$90^\circ$ anti

HW

$\boxed{i, j \Rightarrow ?}$

Q $N \times M$, print boundary elements

$N \times M$




Into to Problem
↓
2D Matrices

$2\frac{1}{2}$ hr

(optional)
Problem solving
↓
next weekend

$16^{th}/17^{th}$