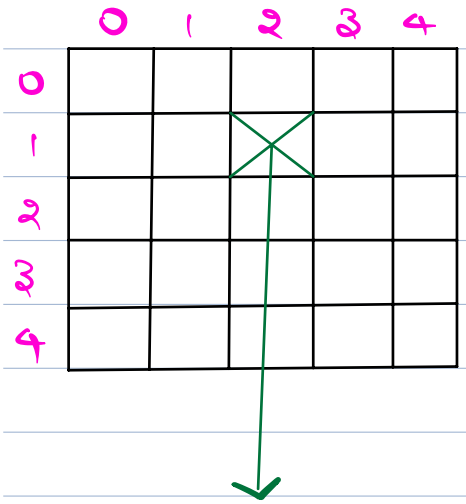
1 D array ⇒ A[i] → One index req.

2 D array ⇒ A[i][j]

2 Dimensional array.



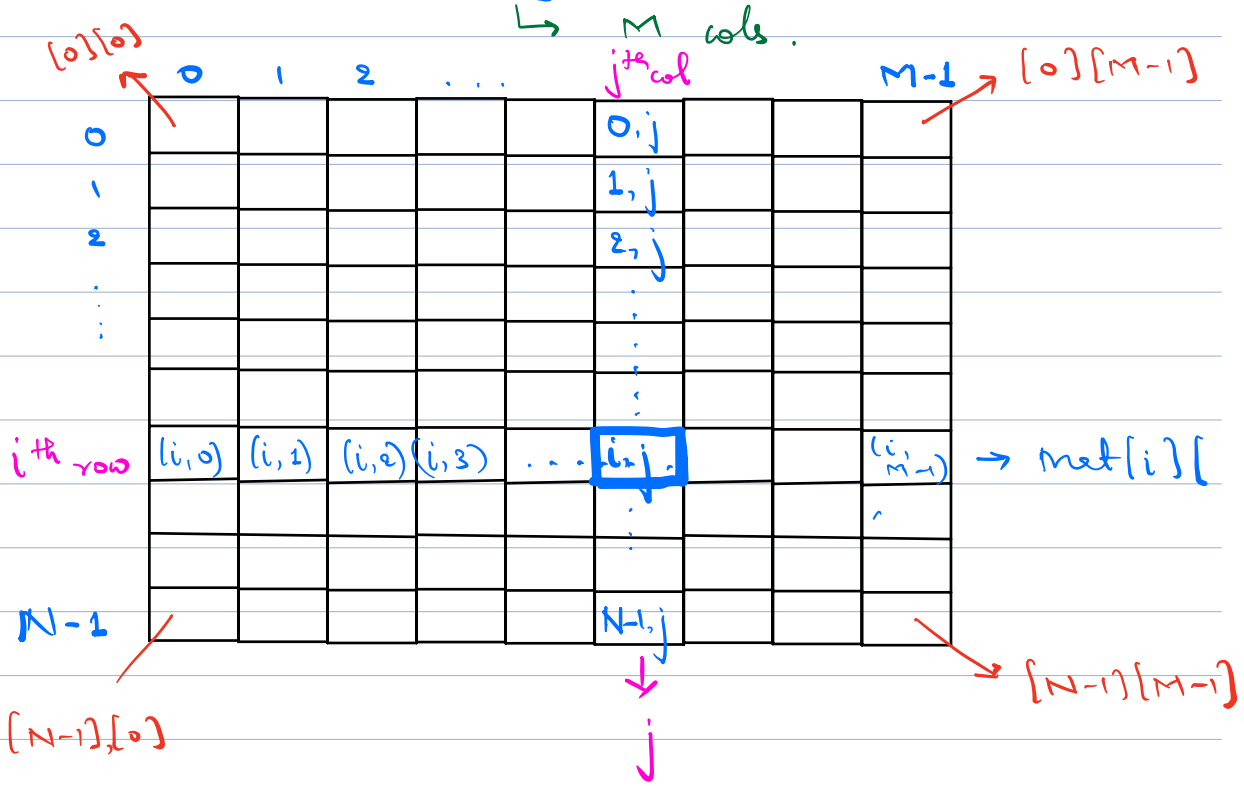
M[5][5]

(# of Rows) (# of columns)

int mat[N][M]

N rows

M cols.



Q Iterate on the ith row from left → right. (N × M)

Sol^n

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

Q Iterate over the entire matrix of size N × M

row by row (Top to Bottom)
(left to right)

Code

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

S.C. = O(1)

T.C. = O(N × M)

Q Iterate over the matrix (N×M) from top to Bottom

H.W left to right

Q Print the row wise sum of the entire matrix.

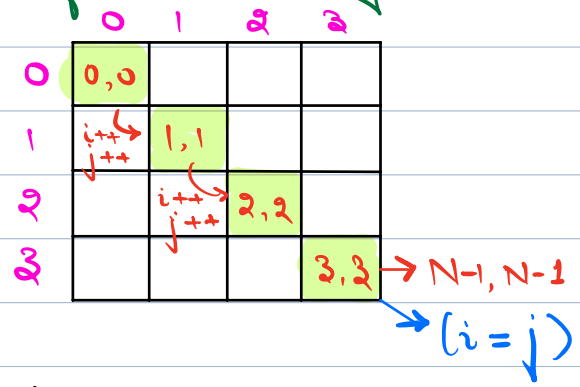
Soln Code

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

T.C. = O(N×M)

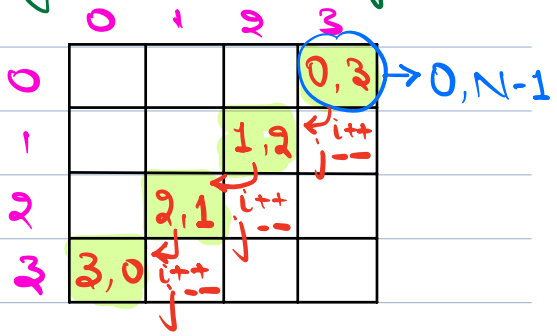
Q Given a square matrix M[N][N]
Print the diagonal.

Left → Right




```
i = 0, j = 0;
while (i<N && j<N) {
    print (M[i][j]);
    i++;
    j++;
}
```

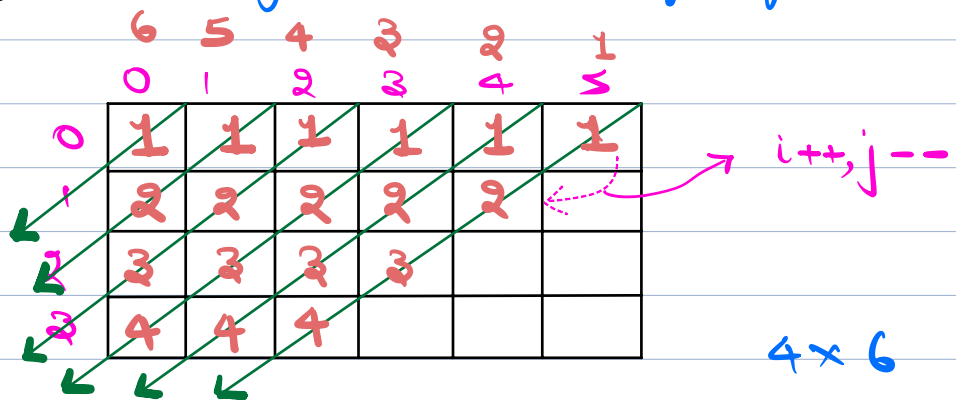
Right → Left




```
i = 0, j = N-1;
while (i<N && j>=0) {
    print (M[i][j]);
    i++;
    j--;
}
```

T.C. = O(N)

Given a matrix of size N×M
Print all diagonals (R → L)
Diagonals starting from row 0.



Code

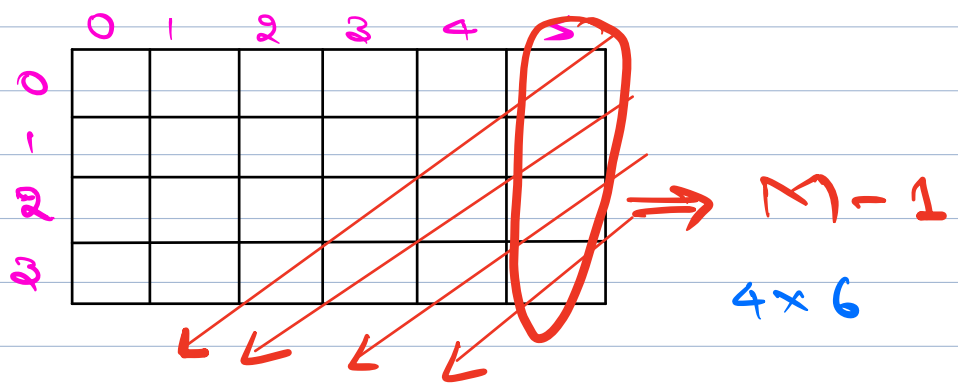
i , j

```
for (col = M-1 ; col >= 0; col--) {
    i = 0, j = col;
    while ((i < N) && (j >= 0)) {
        print (M[i][j]);
        i++;
        j--;
    }
}
```

T.C. = O (N × M)

H.W.

Print diagonals starting on last column



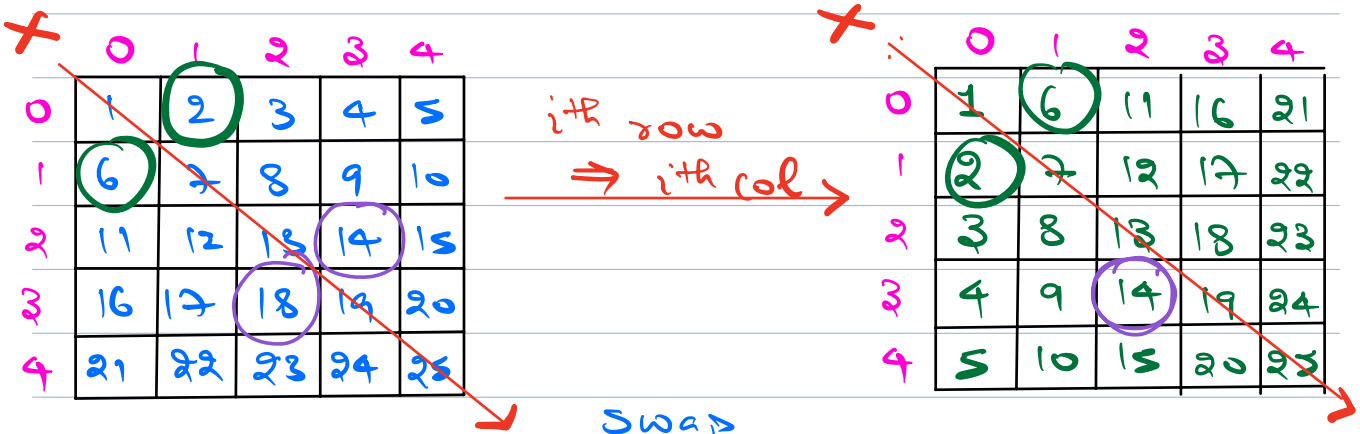
Print all diagonals of a square matrix

H.W.

- Left → Right
- Right → Left

Given a square matrix M[N][N] Convert the matrix to its transpose.

S.C. = O(1)



swap

(M[0][1] → M[1][0]) (M[1][0] → M[0][1])

(M[2][3] → M[3][2]) (M[3][2] → M[2][3])

swap (M[i][j] & M[j][i]).

Code

```
for (i = 0; i < N; i++) {
    for (j = i+1; j < N; j++) {
        temp = M[i][j];
        M[i][j] = M[j][i];
        M[j][i] = temp;
    }
}
```

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

S.C. = $O(1)$

Q Is it possible to convert a rectangular matrix of size NXM (N != M) to its transpose w/o using extra space.

Ex A[2][5]

$$\begin{matrix} 0 \\ 1 \end{matrix}\begin{bmatrix} 1, & 2, & 3, & 4, & 5 \\ 6, & 7, & 8, & 9, & 10 \end{bmatrix} \longrightarrow \begin{bmatrix} 1, & 6 \\ 2, & 7 \\ 3, & 8 \\ 4, & 9 \\ 5, & 10 \end{bmatrix}$$

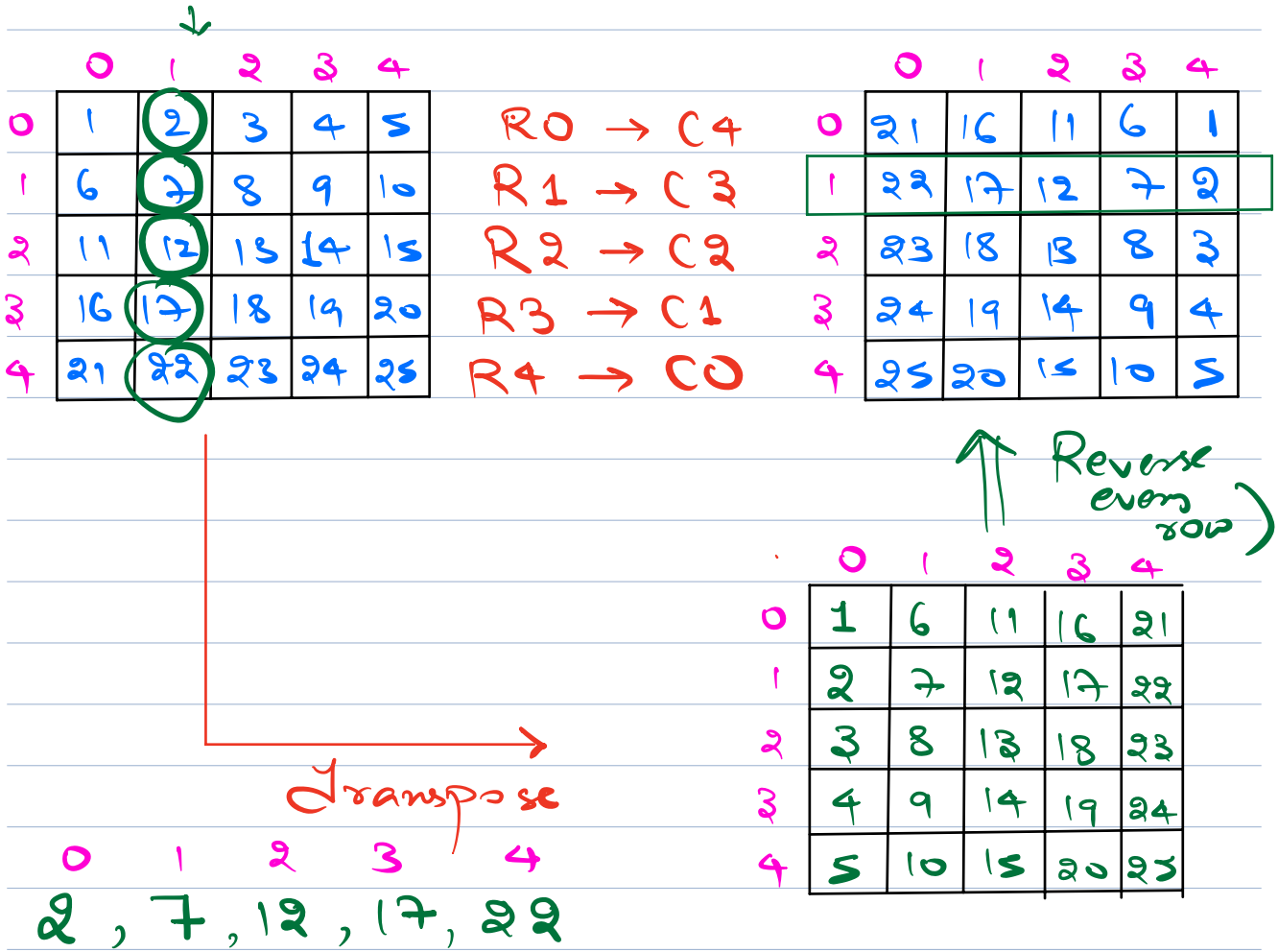
(NXM) 2x5 — 5x2 (MXN)

H.W. Try to fill the new Transpose matrix.

Q Given a matrix of size NXM

Rotate the matrix 90° clockwise from Top-Right corner.

S.C. = O(1)



90° Rotated Matrix = Transpose + Reverse Each Row.

Transpose → $O(N^2)$

Reverse Each Row. → $O(N^2)$

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

S.C. = $O(1)$

Q Is it possible to rotate a rectangular matrix in const space

$$\begin{bmatrix} 1, & 2, & 3 \\ 4, & 5, & 6 \end{bmatrix} \longrightarrow \begin{bmatrix} 4, & 1 \\ 5, & 2 \\ 6, & 3 \end{bmatrix}$$

Not poss. since dimensions change.

ArrayList < ArrayList <> >

vector < vector < > >

list < list < > >

Dynamic Array in _____

↑

language of your choice.