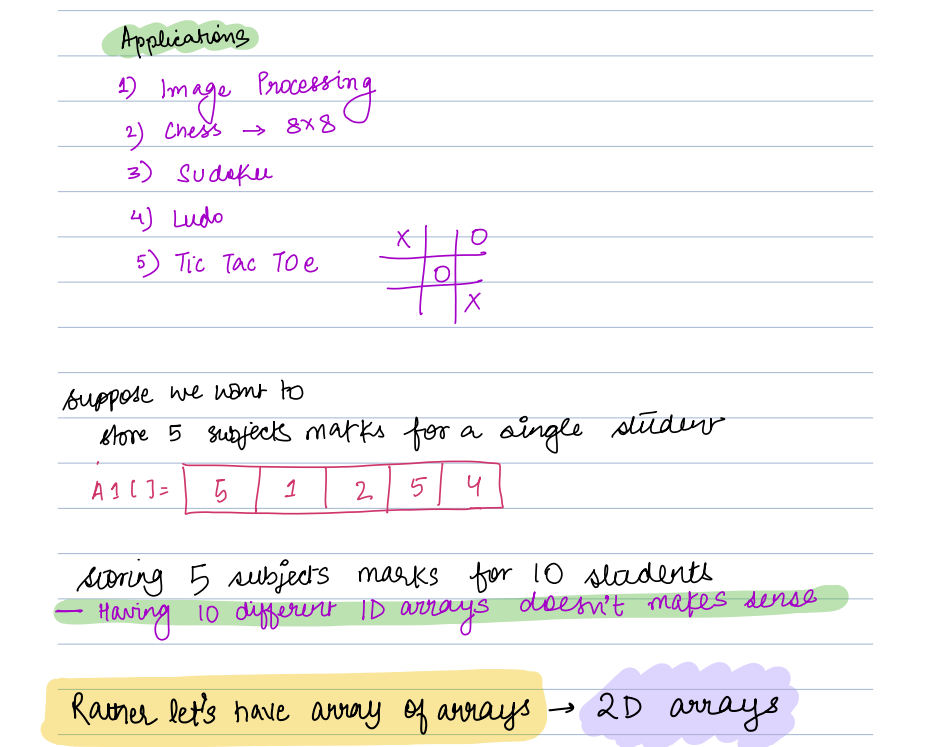
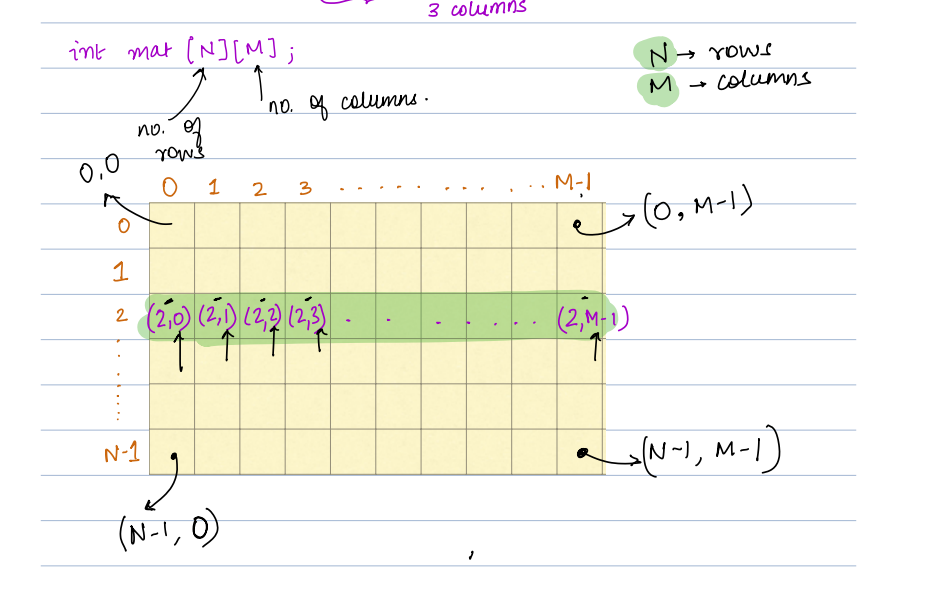
2D Array





Q: Rotate 2D array at 90 degree clock wise

1 ,2

3, 4

o/p

3,1

4,2

1,2,3

4,5,6

7,8,9

o/p

0 1 2

0 7 4 1

1 8 5 2

2 9 6 3

(0,0) => (0,2)

(0,1) => (1,2)

(0,2) => (2,2)

(1,0) => (0,1)

(1,1) => (1,1)

(1,2) => (2,1)

(2,0) => (0,0)

(2,1) => (1,0)

(2,2) => (2,0)

Transpose

1 4 7

2 5 8

3 6 9

for (i =0 ; i< n; i++)

temp = n;

for (j = 0; j < j++)

{

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

temp--

}

}

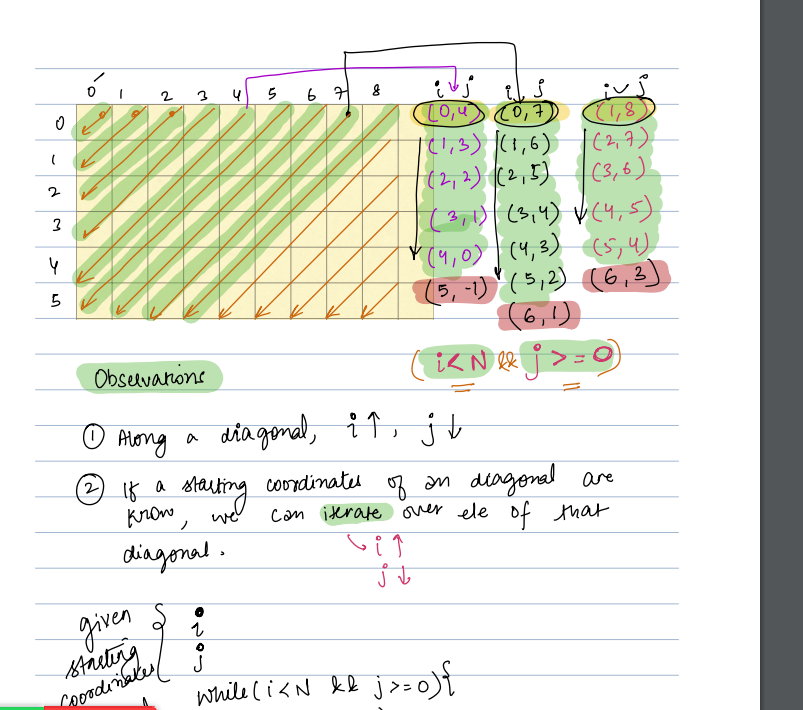
for new array

i j temp A

for same transpose first

reverse rowwise

Q2: print matrix diagonally



(0,0)

(0,1)(1,0)

(0,2)(1,1)(2,0)

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

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

(0,5)(1,4)(2,3)(3,2)(4,1)(5,0)

(0,6)(1,5)(2,4)(3,3)(4,2)(5,1)

(0,7)(1,6)(2,5)(3,4)(4,3)(5,2)

for(var i = 0; i< m; i++)

{

temp = i;

for(var j = 0; j< n && j <= i; j++)

{

Print(A[j][temp]);

temp--

}

var temp1= 0;

for (k = 6 + i; k > i+1 ;k--)

{

print A[temp1][k] ;

temp1++;

}

}

i j temp A[j][temp]

0 0 0 (0,0)

1 0 1 (0,1)

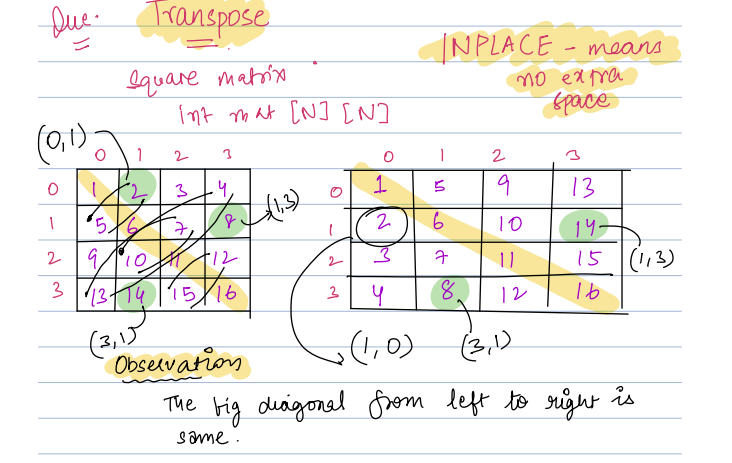
1 1 0 (1,0)

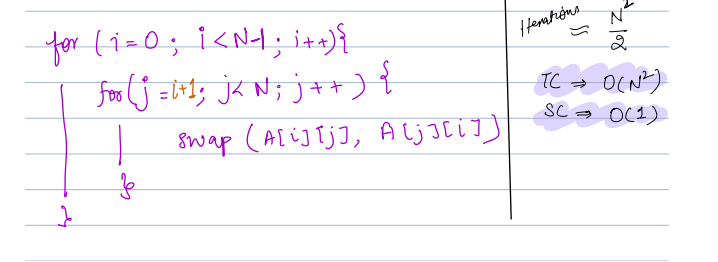
2 0 2 (0,2)

1 1 (1,1)

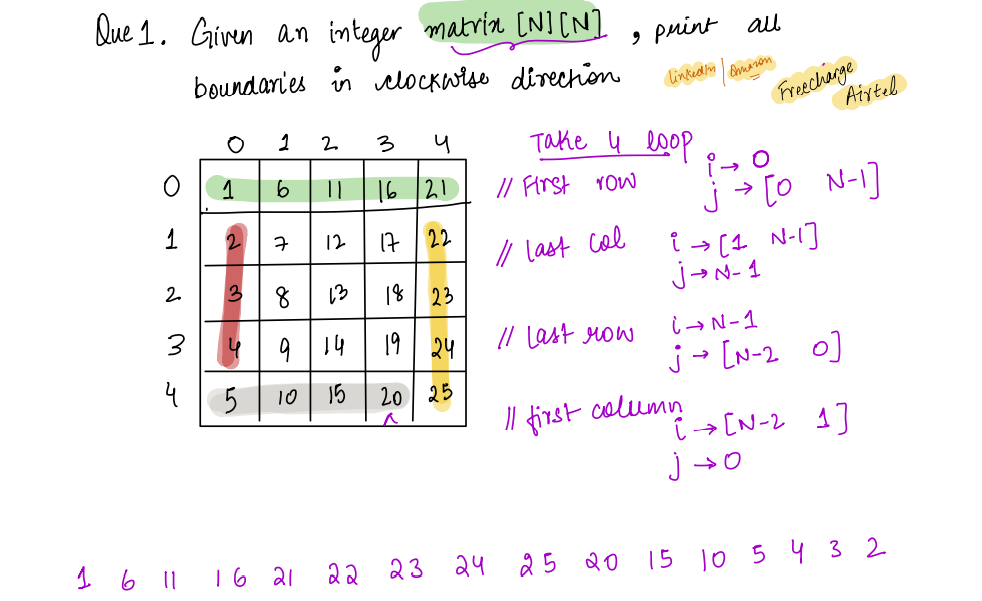
2 0 (2,0)

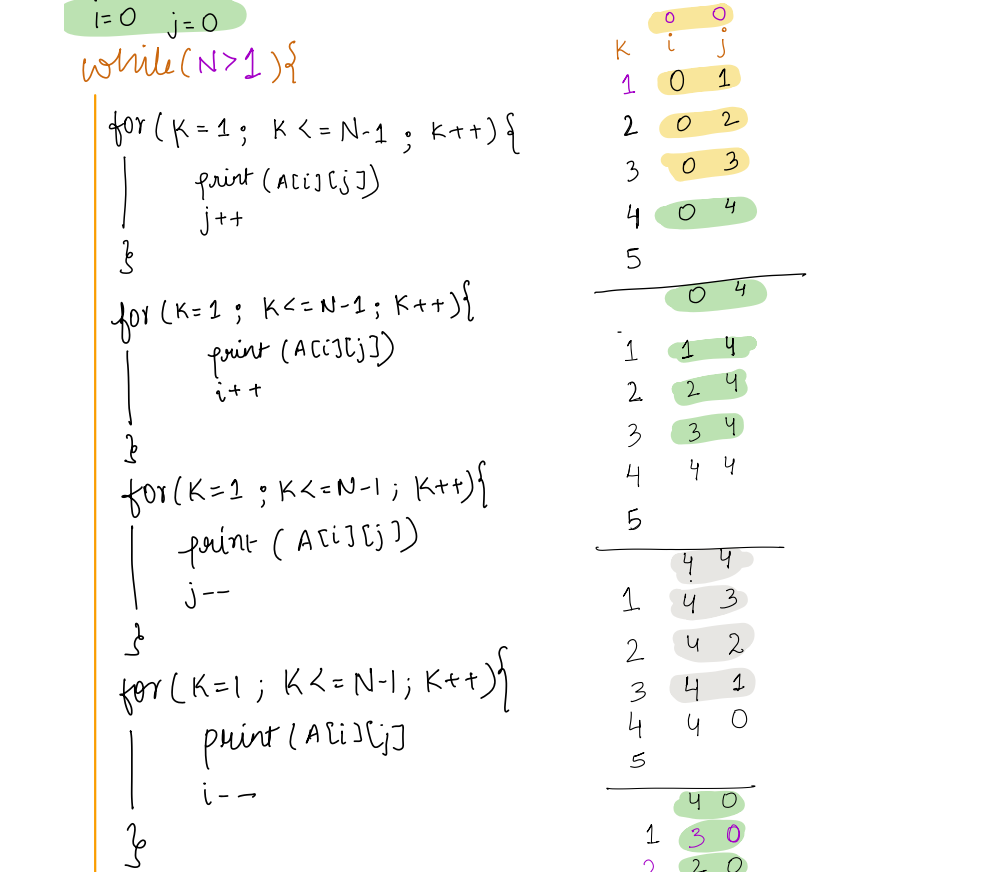
Q3: Transponse Matrix:

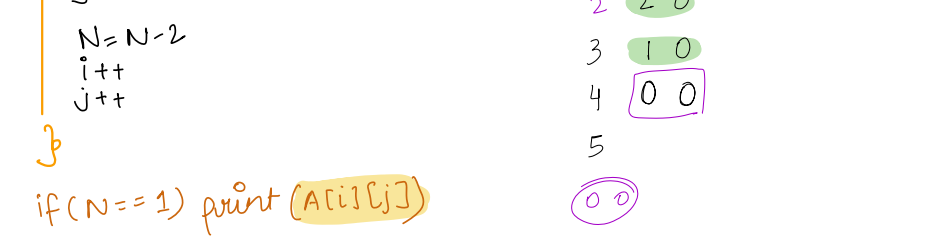




Q2: Print matrix boundaries







TC : O(N\*N)

SC : O(1)

2D Prefix Sum

