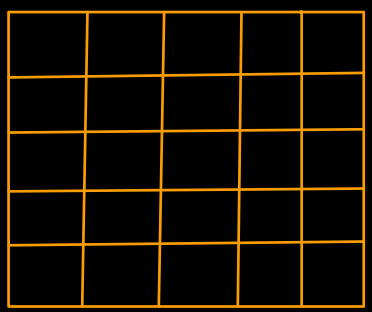
Chess
Sudoku
Snake & Ladder
Excel sheet
Locations
Calender
Images



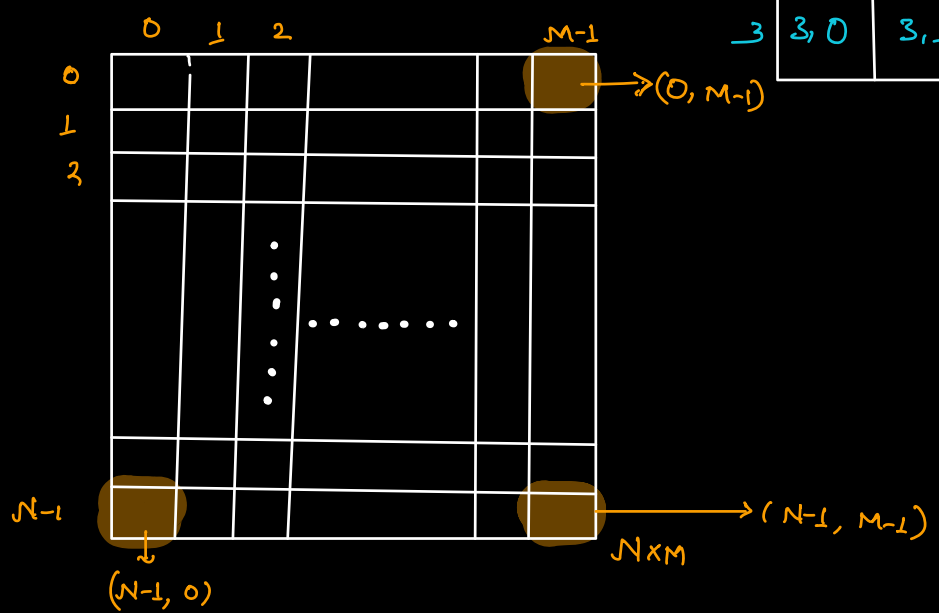
Rows & Col

```
int A[][] = new int [r][c];
```

r → No of rows
c → No of Col

```
int mat[][] = new int [4][3];
```

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



Q Given a matrix of size N x M. Print the sum of every row.

	0	1	2	3	
0	4	3	1	7	→ 15
1	6	2	3	4	→ 15
2	5	3	2	7	→ 17

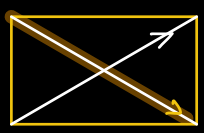
```
void printRowSum(int A[][]){
    int N = A.length; // Count of rows in A
    int M = A[0].length; // Count of cols in A
    int sum = 0;
    for(i=0; i<N; i++){
        sum = 0;
        for(j=0; j<M; j++){
            sum = sum + A[i][j];
        }
        Print(sum);
    }
}
```

TC : O(N M)

HW : Print the sum of all cols.

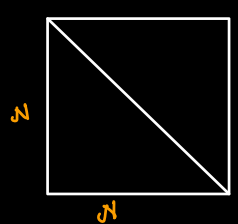
Q Given a square matrix of size N×N. Print the diagonal elements. (Left top → Bottom right)

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



```
for ( i=0; i<N; i++){
    for ( j=0; j<N; j++){
        if( i==j) {
            Print (A[i][j]);
        }
    }
}
```

TC : $O(N^2)$



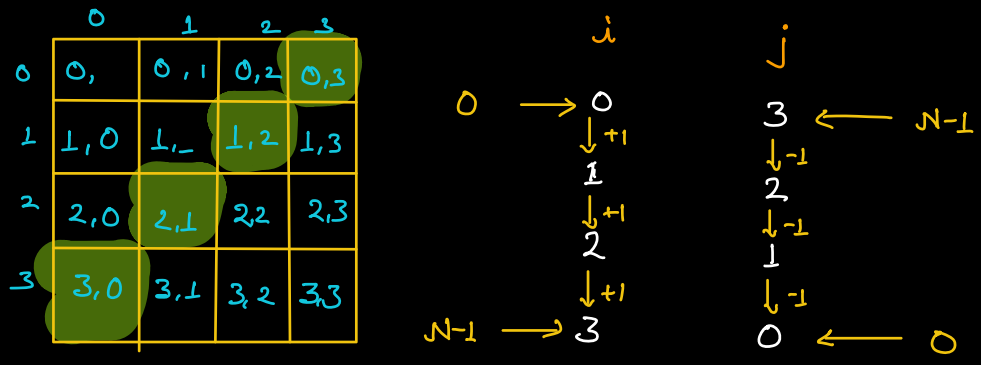
$N^2 + N^2 = d^2$

$d^2 = 2N^2$

$d = \sqrt{2}\,N$

```
for(i=0; i<N; i++){
    Print (A[i][i]);
}
```

→ TC:O(N)

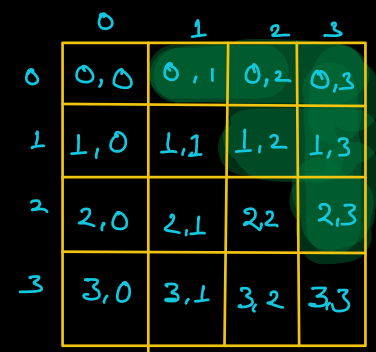


TC : O(N)

Variant

Q Given a matrix (N x N). Print all element above the diagonal. (top left → bottom right).

Row	Col
0	1, 2, 3.., N-1
1	2, 3, 4 ... N-1
2	3, 4, 5 --- N-1
3	4, 5, 6 --- N-1

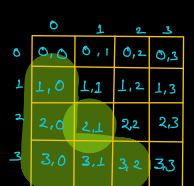



```
for ( i=0; i<N; i++){
    for (j= i+1; j<N ; j++){
        Print (A[i][j]).
    }
}
```

iterations = $\frac{N(N-1)}{2}$

TC : $O(N^2)$

HW: Print the lower triangle elements.



PayTM
ServiceNow

Q Given a sq matrix (N x N). Convert it to its transpose. (Without any extra space → O(1) SC)

A:

1	2	3
4	5	6
7	8	9

→ A^T:

1	4	7
2	5	8
3	6	9

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

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

i	j
0	0
0	1
0	2
1	0

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

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

i, j ⟷ j, i

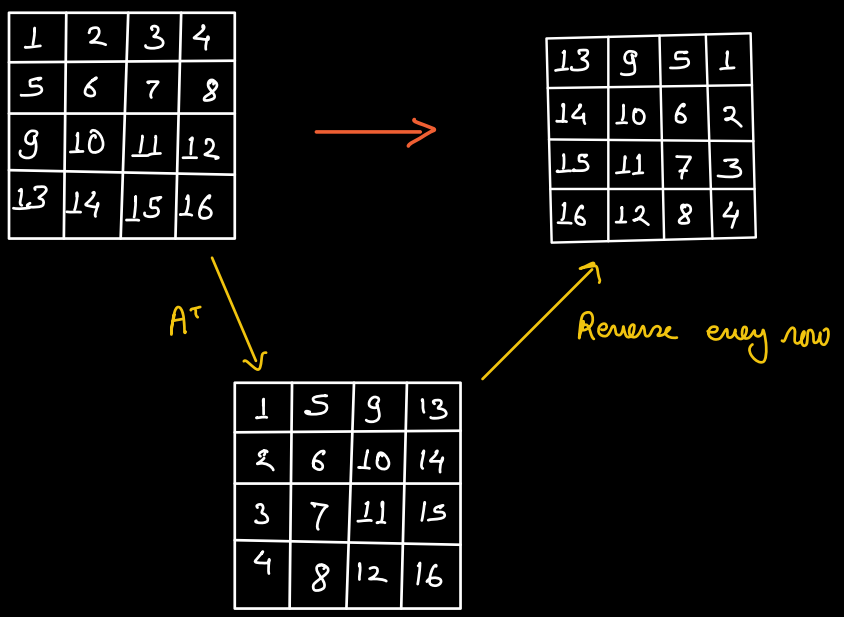
```
for ( i=0; i<N; i++){
    for (j= i+1; j<N ; j++){
        // Swap A[i][j] & A[j][i]
        temp = A[i][j];
        A[i][j] = A[j][i];
        A[j][i] = temp;
    }
}
```

TC : $O(N^2)$

- Pen + Paper (dry run)
- ide

Amazon
Adobe
MS
VMware

Q Given a sq matrix.
Rotate the matrix by 90° in clockwise direction &
Without using any extra space (SC: O(1)).



Steps

- Convert the matrix to its transpose.
- Reverse every row of the transpose.

	TC	SC
	$O(N^2)$	$O(1)$
+	$O(N^2)$	$O(1)$
	$O(N^2)$,	$O(1)$