### Rotate by 90 degree

**Company Tags**

[**Paytm**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Paytm) [**Zoho**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Zoho) [**Morgan Stanley**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Morgan%20Stanley) [**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**Samsung**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Samsung) [**Snapdeal**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Snapdeal) [**D-E-Shaw**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=D-E-Shaw)

Statement….

Given asquare **matrix[][]** of size **N x N**. The task is to rotate it by**90 degrees in an anti-clockwise** direction without using any extra space.

**Example 1:**

**Input:**

N = 3

matrix[][] = [[1 2 3],

  [4 5 6],

  [7 8 9]]

**Output:**

3 6 9

2 5 8

1 4 7

JAVA CODE..

//{ Driver Code Starts

//Initial Template for Java

import java.util.\*;

import java.io.\*;

import java.lang.\*;

class DriverClass

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while (t-- > 0)

{

int n = sc.nextInt();

int[][] arr = new int[n][n];

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

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

arr[i][j] = sc.nextInt();

CodingMaxima g = new CodingMaxima ();

g.rotate(arr);

printMatrix (arr);

}

}

static void printMatrix(int arr[][])

{

for (int i = 0; i < arr.length; i++) {

for (int j = 0; j < arr[0].length; j++)

System.out.print(arr[i][j] + " ");

System.out.println("");

}

}

}

// } Driver Code Ends

//User function Template for Java

class CodingMaxima

{

static void rotate(int matrix[][])

{

int n=matrix.length;

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

for (int j = i + 1; j < n; j++) {

int temp = matrix[i][j];

matrix[i][j] = matrix[j][i];

matrix[j][i] = temp;

}

}

for(int i=0;i<n/2;i++){

for(int j=0;j<n;j++){

int temp=matrix[i][j];

matrix[i][j]=matrix[n-1-i][j];

matrix[n-1-i][j]=temp;

}

}

}

}