GeeksforGeeks
A computer science portal for geeks

Custom Search

Suggest a Topic

Login

Write an Article

# Inplace rotate square matrix by 90 degrees | Set 1

Given an square matrix, turn it by 90 degrees in anti-clockwise direction without using any extra space.

**Examples :**

```
Input
 1  2  3
 4  5  6
 7  8  9

Output:
 3  6  9
 2  5  8
 1  4  7

Input:
 1  2  3  4
 5  6  7  8
 9 10 11 12
13 14 15 16

Output:
 4  8 12 16
 3  7 11 15
 2  6 10 14
 1  5  9 13
```

**Recommended: Please solve it on "*PRACTICE*" first, before moving on to the solution.**

An approach that requires extra space is already discussed here.

**How to do without extra space?**

Below are some important observations.

*First row of source –> First column of destination, elements filled in opposite order*

Second row of source –> Second column of destination, elements filled in opposite order

so … on

Last row of source –> Last column of destination, elements filled in opposite order.

An N x N matrix will have floor(N/2) square cycles. For example, a 4 X 4 matrix will have 2 cycles. The first cycle is formed by its 1st row, last column, last row and 1st column. The second cycle is formed by 2nd row, second-last column, second-last row and 2nd column.

The idea is for each square cycle, we swap the elements involved with the corresponding cell in the matrix in anti-clockwise direction i.e. from top to left, left to bottom, bottom to right and from right to top one at a time. We use nothing but a temporary variable to achieve this.

Below steps demonstrate the idea

```
First Cycle (Involves Red Elements)
 1  2  3 4
 5  6  7  8
 9 10 11 12
 13 14 15 16


Moving first group of four elements (First
elements of 1st row, last row, 1st column
and last column) of first cycle in counter
clockwise.
 4  2  3 16
 5  6  7 8
 9 10 11 12
 1 14  15 13

Moving next group of four elements of
first cycle in counter clockwise
 4  8  3 16
 5  6  7  15
 2  10 11 12
 1  14  9 13

Moving final group of four elements of
first cycle in counter clockwise
 4  8 12 16
 3  6  7 15
 2 10 11 14
 1  5  9 13


Second Cycle (Involves Blue Elements)
 4  8 12 16
 3  6 7  15
 2  10 11 14
 1  5  9 13

Fixing second cycle
 4  8 12 16
 3  7 11 15
 2  6 10 14
 1  5  9 13
```

Below is the implementation of above idea.

## C++

```
// C++ program to rotate a matrix by 90 degrees
#include <bits/stdc++.h>
#define N 4
using namespace std;

void displayMatrix(int mat[N][N]);

// An Inplace function to rotate a N x N matrix
// by 90 degrees in anti-clockwise direction
void rotateMatrix(int mat[][N])
{
    // Consider all squares one by one
    for (int x = 0; x < N / 2; x++)
    {
        // Consider elements in group of 4 in
        // current square
        for (int y = x; y < N-x-1; y++)
        {
            // store current cell in temp variable
            int temp = mat[x][y];

            // move values from right to top
            mat[x][y] = mat[y][N-1-x];

            // move values from bottom to right
            mat[y][N-1-x] = mat[N-1-x][N-1-y];

            // move values from left to bottom
            mat[N-1-x][N-1-y] = mat[N-1-y][x];

            // assign temp to left
            mat[N-1-y][x] = temp;
        }
    }
}

// Function to print the matrix
void displayMatrix(int mat[N][N])
{
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
            printf("%2d ", mat[i][j]);

        printf("\n");
    }
    printf("\n");
}


/* Driver program to test above functions */
int main()
{
    // Test Case 1
    int mat[N][N] =
    {
        {1, 2, 3, 4},
        {5, 6, 7, 8},
        {9, 10, 11, 12},
        {13, 14, 15, 16}
    };
```

```
            System.out.print("\n");
        }
        System.out.print("\n");
    }

    /* Driver program to test above functions */
    public static void main (String[] args)
    {
        int N = 4;

        // Test Case 1
        int mat[][] =
        {
            {1, 2, 3, 4},
            {5, 6, 7, 8},
            {9, 10, 11, 12},
            {13, 14, 15, 16}
        };


        // Tese Case 2
        /* int mat[][] = {
                            {1, 2, 3},
                            {4, 5, 6},
                            {7, 8, 9}
                        };
         */

        // Tese Case 3
        /*int mat[][] = {
                        {1, 2},
                        {4, 5}
                    };*/

        // displayMatrix(mat);

        rotateMatrix(N,mat);

        // Print rotated matrix
        displayMatrix(N,mat);
    }
}

// This code is contributed by Prakriti Gupta
```

Run on IDE Copy Code

## Python3

```
# Python3 program to rotate a matrix by 90 degrees
N = 4

# An Inplace function to rotate
# N x N matrix by 90 degrees in
# anti-clockwise direction
def rotateMatrix(mat):

    # Consider all squares one by one
    for x in range(0, int(N/2)):

        # Consider elements in group
        # of 4 in current square
        for y in range(x, N-x-1):

            # store current cell in temp variable
            temp = mat[x][y]

            # move values from right to top
            mat[x][y] = mat[y][N-1-x]
            # move values from bottom to right
            mat[y][N-1-x] = mat[N-1-x][N-1-y]

            # move values from left to bottom
            mat[N-1-x][N-1-y] = mat[N-1-y][x]

            # assign temp to left
            mat[N-1-y][x] = temp


# Function to pr the matrix
def displayMatrix( mat ):

    for i in range(0, N):

        for j in range(0, N):

            print (mat[i][j], end = ' ')
        print ("")




# Driver Code
mat = [[0 for x in range(N)] for y in range(N)]

# Test case 1
mat = [ [1, 2, 3, 4 ],
        [5, 6, 7, 8 ],
        [9, 10, 11, 12 ],
        [13, 14, 15, 16 ] ]

'''
# Test case 2
mat = [ [1, 2, 3 ],
        [4, 5, 6 ],
        [7, 8, 9 ] ]

# Test case 3
mat = [ [1, 2 ],
        [4, 5 ] ]

'''

rotateMatrix(mat)

# Print rotated matrix
displayMatrix(mat)


# This code is contributed by saloni1297
```

Run on IDE Copy Code

## C#

```
// C# program to rotate a
// matrix by 90 degrees
using System;

class GFG
{
    // An Inplace function to
    // rotate a N x N matrix
    // by 90 degrees in anti-
    // clockwise direction
    static void rotateMatrix(int N,
                             int [,]mat)
    {
        // Consider all
        // squares one by one
```