

DAILY ONLINE ACTIVITIES SUMMARY

Date:	19-06-2020	Name:	Nayan. P. Joshi
Sem & Sec	8 th Sem A	USN:	4AL16CS058
Online Test Summary			
Subject	Big Data Analytics		
Max. Marks	30	Score	22
Certification Course Summary			
Course	Introduction to Information Security		
Certificate Provider	Great learning academy	Duration	6hrs
Coding Challenges			
Problem Statement: Write a C Program to rotate a Matrix by 90 Degree in Clockwise or Anticlockwise Direction.			
Status: Solved			
Uploaded the report in GitHub		yes	
If yes Repository name		nayan1998	
Uploaded the report in slack		yes	

Largest Tech Community | Hack... x +

techgig.com/challenge/result/module-2/bzd2SzZ4OXZjTIFscVNtWUVHeHV2Zz09

nayan19985july@gmail.com Logout

Test Completed!

You have successfully participated in CSE_BDA_9.

Rate this Test
Your Rating: ★★★★★ • Click to Rate

Results Analytics

✓ Module 2

Your Score **22** / 30

Type here to search

09:43 AM 19-06-2020

Introduction to Information Secu... x Tables - Bootstrap v4.5 x +

olympus.greatlearning.in/courses/11264

greatlearning Learning for Life Home Live Sessions Certificates My Courses

CONTENT ASSESSMENTS

Learning Material

Stanford and Computer Security

Intro to Stanford and Computer Security Field	✓
Computer Security - Its applications and its future	✓
Innovations in Cybersecurity - Quantum Computing	✓
What is the future of cryptography?	4m ✓

Introduction to Computer Security- Video Lessons

Introduction to Software Security Lesson 1	1h ✓
Introduction to Software Security Lesson 2	1h

Type here to search

11:35 AM 19-06-2020

Write a C Program to rotate a Matrix by 90 Degree in Clockwise or Anticlockwise Direction

```
#include <bits/stdc++.h>
```

```
#define N 3
```

```
using namespace std;
```

```
// Function to reverse rows of the matrix
```

```
void reverse_rows(int arr[N][N])
```

```
{
```

```
    int k;
```

```
    for (int i = 0; i < N; i++)
```

```
    {
```

```
        k = N-1;
```

```
        for (int j = 0; j < k; j++)
```

```
        {
```

```
            swap(arr[i][j], arr[i][k]);
```

```
            k--;
```

```
        }}}}
```

```
// Transpose of a matrix
```

```
void transpose1(int arr[N][N])
```

```
{
```

```
    for (int i = 0; i < N; i++)
```

```
        for (int j = i; j < N; j++)
```

```
            swap(arr[i][j], arr[j][i]);
```

```
}
```

```
// print the matrix
```

```
void print_matrix1(int mat[N][N])
```

```
{
```

```
    for (int i = 0; i < N; i++)
```

```
    {
```

```

        for (int j = 0; j < N; j++)
            printf("%d\t", mat[i][j]);

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

void reverse_column(int arr[N][N])
{
    int k;
    for (int i = 0; i < N; i++)
    {
        k = N-1;
        for (int j = 0; j < k; j++)
        {
            swap(arr[j][i], arr[k][i]);
            k--;
        }
    }
}

// Transpose of a matrix
void transpose2(int arr[N][N])
{
    for (int i = 0; i < N; i++)
        for (int j = i; j < N; j++)
            swap(arr[i][j], arr[j][i]);
}

void print_matrix2(int mat[N][N])
{
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
            printf("%d\t", mat[i][j]);

        printf("\n");
    }
}

```

```

    }
    printf("\n");
}

// Main function
int main()
{
    int mat[N][N] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
    printf("The matrix before rotation\n");
    print_matrix1(mat);
    transpose1(mat);
    printf("The matrix after rotation - Clockwise\n");
    reverse_rows(mat);
    print_matrix1(mat);
    printf("-----
\n");
    printf("The matrix before rotation\n");
    print_matrix2(mat);
    transpose2(mat);
    printf("The matrix after rotation - anticlockwise\n");
    reverse_column(mat);
    print_matrix2(mat);
    return 0;
}

```