



Data Structure and Algorithm (MCA 271)

Lab Practical –

BY

Himanshu Heda (24225013)

SUBMITTED TO

Prof. Vandna Kansal

SCHOOL OF SCIENCES

2024-2025

Program Description:

Code of the program

Output: - Paste the o/p of the program.

```
// 1. Transpose of a matrix

#include <stdio.h>
void main()
{
    int a[3][3], transposed[3][3], i, j;
    printf("Enter matrix elements : ");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    printf("Matrix : \n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d\t", a[i][j]);
        }
        printf("\n");
    }
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            transposed[j][i] = a[i][j]; // Swap indices
        }
    }

    // Display the transposed matrix
    printf("Transposed Matrix : \n");
```

```
for (i = 0; i < 3; i++)
{
    for (j = 0; j < 3; j++)
    {
        printf("%d\t", transposed[i][j]);
    }
    printf("\n");
}
```

Output : --

```
PS D:\2MCA\DSA> .\array_transpose.exe
```

```
Enter matrix elements : 1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
Matrix :
```

```
1      2      3
```

```
4      5      6
```

```
7      8      9
```

```
Transposed Matrix :
```

```
1      4      7
```

```
2      5      8
```

```
3      6      9
```

```
PS D:\2MCA\DSA> █
```

```
// 2. Multiplication of two matrices.

#include <stdio.h>
void main()
{
    int a[3][3], b[3][3], c[3][3], i, j, k;
    printf("Enter matrix A elements : ");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    printf("Enter matrix B elements : ");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            scanf("%d", &b[i][j]);
        }
    }
    printf("Matrix A : \n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d\t", a[i][j]);
        }
        printf("\n");
    }
    printf("Matrix B : \n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d\t", b[i][j]);
        }
        printf("\n");
    }
}
```

```
// Matrix multiplication
printf("Matrix A * B : \n");
for (i = 0; i < 3; i++) {
    for (j = 0; j < 3; j++) {
        c[i][j] = 0; // Initialize the element
        for (k = 0; k < 3; k++) {
            c[i][j] += a[i][k] * b[k][j]; // Dot product
        }
        printf("%d\t", c[i][j]);
    }
    printf("\n");
}
```

Output : --

```
PS D:\2MCA\DSA> .\matrix_mul.exe
```

```
Enter matrix A elements : 1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
Enter matrix B elements : 9
```

```
8
```

```
7
```

```
6
```

```
5
```

```
4
```

```
3
```

```
2
```

```
1
```

```
Matrix A :
```

```
1      2      3
```

```
4      5      6
```

```
7      8      9
```

```
Matrix B :
```

```
9      8      7
```

```
6      5      4
```

```
3      2      1
```

```
Matrix A * B :
```

```
30      24      18
```

```
84      69      54
```

```
138     114     90
```

```
PS D:\2MCA\DSA> █
```

