

Assignment No.:

1

Title:

To Matrix Operations

Subject:

Data Structures Laboratory

Class:

SY13 (BTech) C.S.E.

Roll No.:

2215055 (MITV21BTIT0010)

Assessment (Marks):

Signature and Date of

Assessment:

MIT SCHOOL OF ENGINEERING

Rajbaug, Loni-Kalbhor, Pune



Experiment No. 1

- * Experiment Title :— Implement C++ program to perform matrix operations.

* Objectives :—

- a) To understand the use of a 2D Array.
- b) To understand matrix operations.

* Problem Statement :— Write a program in C++ to perform matrix operations like add_matrix(), sub_matrix(), multi_matrix() using appropriate data structure. Also write how to determine time complexity and space complexity of your program.

* Outcomes :—

- 1) Understanding the use of 2-dimensional Array.
- 2) Understanding the matrix operations.
→ To add or subtract matrices, there must be of identical order and for multiplication, the number of columns in the first matrix equals the number of rows in the second matrix.

* Addition of 2 Matrices:
 → Consider the two matrices A & B of order 2×2 . Then the sum is given by :-

$$\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} + \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} = \begin{bmatrix} a_1+a_2 & b_1+b_2 \\ c_1+c_2 & d_1+d_2 \end{bmatrix}$$

* Algorithm (Matrix Addition)

- 1) Start
- 2) Define 2 matrices and their respective row and column numbers.
- 3) Read no of rows and columns (size of matrix) to perform matrix addition.
- 4) Read two matrices A & B as per specified size.
- 5) Use 2 for loops, outer for loop is used to read row elements & inner for loop to read column elements.
- 6) Perform addition of 2 elements of matrix A & B.
- 7) Finally display the result for addition of given matrices.
- 8) Stop.

MIT SCHOOL OF ENGINEERING

Rajbaug, Lonu-Kalbhor, Pune



MIT-ADT
UNIVERSITY
PUNE, INDIA
A step towards world class education

* Multiplication of 2 Matrices :-

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} * \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{pmatrix}$$

* Algorithm (Matrix Multiplication)

- 1) Start.
- 2) Define three matrices A, B, C and read their respective row and column numbers.
- 3) We check if the matrix can be multiplied or not, if n is not equal to matrix can't be multiplied and an error message is generated.
- 4) Read matrices A and B.
- 5) First, start a loop which goes up to giving row elements of A, Secondly, inside it again starts a loop which goes up to giving row elements of B. At last, we define a loop which goes up to giving column element of B.
- 6) Then, store their corresponding multiplication by $\text{sum} = \text{sum} + A[i][k] * B[k][j]$ which gets updated each time and acts as the mathematical formula of multiplication used for matrix.
- 7) sum is assigned into $C[i][j]$ and likewise stores the multiplication result of matrix A and B.
- 8) Finally, display the result for multiplication of given matrices.
- 9) Stop.

MIT SCHOOL OF ENGINEERING

Rajbaug, Lonikalbhor, Pune



- * Conclusion - Thus, we implemented matrix operations using 2D array.

* Questions

- 1) What is 2D array?

- It is similar to 1D array which can be viewed as a grid (or table) with rows and columns.

2) Define the terms time complexity & space complexity?

- Time complexity is the time taken by the algorithm to execute each set of instructions.

- Space complexity is usually referred to as the amount of memory consumed by the algorithm.

/*
ASSIGNMENT NO :- 1

problem statement

Write a program in C++ to perform matrix operations like add_matrix(), sub_matrix(), multi_matrix() using appropriate data structure. Also write how to determine time complexity and space complexity of your program.

Author: Anurag Haldey.

Date: 18/08/2022.

*/

```
#include <iostream>
using namespace std;

class matrix
{
public:
    int a[20][20], b[20][20], sum[20][20], subs[20][20], mult[20][20], i, j, o, r, c, k;
    char mat;

    void input();
    void display();
    void add();
    void sub();
    void mul();
    void transpose();

};

void matrix::input()
{
    cout<<"Enter rows and column number here ";
    cin>>r>>c;
    cout<<"enter "<<r*c<<" elements"<<"in the matrix\n";
    cout<<"Enter the elements in First matrix"<<endl;
```

```

        for(i=0;i<r;i++){
            for(j=0;j<c;j++){
                cin>>a[i][j];
            }
        }

        cout<<"Enter the elements in Second matrix"<<endl;
        for(i=0;i<r;i++){
            for(j=0;j<c;j++){
                cin>>b[i][j];
            }
        }

        void matrix::display()
        {
            cout<<"First matrix: "<<endl;
            for(i=0;i<r;i++){
                for(j=0;j<c;j++){
                    cout<<a[i][j]<<\t';
                }
                cout<<endl;
            }

            cout<<"Second matrix: "<<endl;
            for(i=0;i<r;i++){
                for(j=0;j<c;j++){
                    cout<<b[i][j]<<\t';
                }
                cout<<endl;
            }
        }

        void matrix::add()
        {
            cout<<"Addition of matrix"<<endl;
            for(i=0;i<2;i++){
                for(j=0;j<2;j++){
                    sum[i][j]=a[i][j]+b[i][j];
                }
            }

            cout<<sum[i][j]<<\t';
        }
    }
}

```

```

    }

    cout<<endl;
}

}

void matrix::sub()
{
    cout<<" Subtraction of matrix"<<endl;
    for(i=0;i<2;i++){
        for(j=0;j<2;j++){
            sub[i][j]=a[i][j]-b[i][j];
            cout<<sub[i][j]<<'t';
        }
        cout<<endl;
    }
}

void matrix::mul()
{
    cout<<" Multiplication of matrix"<<endl;
    for(i=0;i<2;i++){
        for(j=0;j<2;j++){
            mult[i][j]=0;
            for(o=0;o<2;o++){
                mult[i][j]+=a[i][o]*b[o][j];
            }
            cout<<mult[i][j]<<'t';
        }
        cout<<endl;
    }
}

void matrix::transpose()
{
    cout<<"\n choose your matrix to transpose operation(a /b): ";
    cin>>mat;
    if(mat=='a')

```

```
{  
    for(i=0;i<r;i++)  
    {  
        for(j=0;j<c;j++)  
        {  
            cout<<"\t"<<a[j][i];  
        }  
        cout<<"\n";  
    }  
  
}  
else{  
    for(i=0;i<r;i++)  
    {  
        for(j=0;j<c;j++)  
        {  
            cout<<"\t"<<b[j][i];  
        }  
        cout<<"\n";  
    }  
}  
}  
  
int main()  
{  
    matrix m;  
    m.input();  
    m.display();  
    m.add();  
    m.sub();  
    m.mul();  
    m.transpose();  
    return 0;  
}
```

Output:

Enter rows and column number here

2 2

enter 4 elements in the matrix

Enter the elements in First

matrix 1

2

3

4

Enter the elements in Second matrix

3

4

5

6

First matrix:

1 2

3 4

Second matrix:

3 4

5 6

Addition of matrix

4 6

8 10

Subtraction of matrix

-2 -2

-2 -2

Multiplication of matrix

13 16

29 36

choose your matrix to transpose operation(a /b):

b 3 5