

Lab Manual 9

Submitted by: Ali Rehman Qureshi

Class: ME-15(C)

ID: 459203

Lab Task

Task 1:

Make 2D Array in C++ and print left diagonal and right diagonal sum of a 3x3 matrix.

```
#include <iostream>
using namespace std;
int main()
{
    int mat[3][3];
    int sum = 0;
    for (int i = 0; i < 3; i++)
    {
        cout << "Enter data for row " << i + 1 << endl;
        for (int j = 0; j < 3; j++)
        {
            cin >> mat[i][j];
        }
    }
    cout << "The resultant matrix\n";
    for (int m = 0; m < 3; m++)
    {
        for (int n = 0; n < 3; n++)
        {
            cout << mat[m][n] << " ";
        }
        cout << endl;
    }
    for (int k = 0; k < 3; k++)
    {
        sum += mat[k][k];
    }
    cout << "The sum of the right diagonal is " << sum << endl;
    sum = 0;
    for (int j = 2; j >= 0; j--)
    {
        sum += mat[2-j][j];
    }
    cout << "The sum of the left diagonal is " << sum;
    return 0;
}
```

```
Microsoft Visual Studio Debug Console
Enter data for row 1
7
8
9
Enter data for row 2
9
11
14
Enter data for row 3
15
7
14
The resultant matrix
7 8 9
9 11 14
15 7 14
The sum of the right diagonal is 32
The sum of the left diagonal is 35
```

Task 2:

Write a function to add two 2D arrays of size 3x3.

```
#include <iostream>
using namespace std;
int result(int a[3][3], int b[3][3])
{
    int sum[3][3];
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            sum[i][j] = a[i][j] + b[i][j];
        }
    }
    cout << "The resultant matrix after addition is: " << endl;
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            cout << sum[i][j]<<" ";
        }
        cout << endl;
    }
    return 0;
}
int main()
{
    int num = 0;
    int mat1[3][3];
    int mat2[3][3];
    cout << "Enter the data for matrix number one\n";
    for (int i = 0; i < 3; i++)
    {
        cout << "Enter data for row " << i + 1 << endl;
        for (int j = 0; j < 3; j++)
        {
            cin >> mat1[i][j];
        }
    }
}
```

```

    }
}
cout << "The resultant matrix\n";
for (int m = 0; m < 3; m++)
{
    for (int n = 0; n < 3; n++)
    {
        cout << mat1[m][n]<<" ";
    }
    cout << endl;
}
cout << "Enter the data for matrix number two\n";
for (int i = 0; i < 3; i++)
{
    cout << "Enter data for row " << i + 1 << endl;
    for (int j = 0; j < 3; j++)
    {
        cin >> mat2[i][j];
    }
}
cout << "The resultant matrix\n";
for (int m = 0; m < 3; m++)
{
    for (int n = 0; n < 3; n++)
    {
        cout << mat2[m][n] << " ";
    }
    cout << endl;
}
result(mat1, mat2);
return 0;
}

```

```
Microsoft Visual Studio Debug Console
Enter the data for matrix number one
Enter data for row 1
1
2
3
Enter data for row 2
4
5
6
Enter data for row 3
7
10
8
The resultant matrix
1 2 3
4 5 6
7 10 8
Enter the data for matrix number two
Enter data for row 1
7
8
9
Enter data for row 2
11
9
4
Enter data for row 3
4
11
11
The resultant matrix
7 8 9
11 9 4
4 11 11
The resultant matrix after addition is:
8 10 12
15 14 10
11 21 19
```

Task 3:

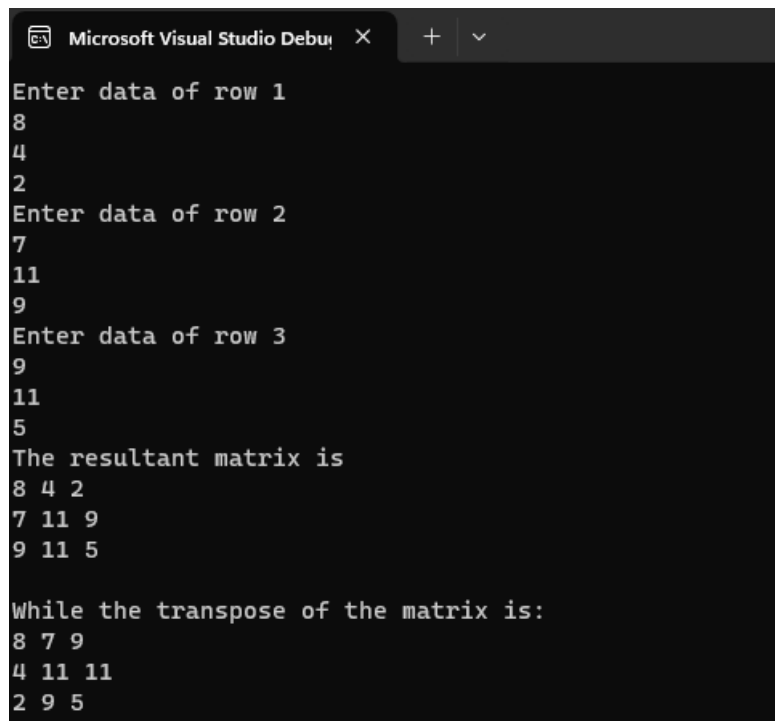
Using 2D arrays in C++, take transpose of a 3x3 matrix. Make a transpose function.

```
#include <iostream>
using namespace std;
int transpose(int a[3][3])
{
    int result[3][3];
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            result[j][i] = a[i][j];
        }
    }
    cout << "While the transpose of the matrix is: " << endl;
    for (int i = 0; i < 3; i++)
```

```

        {
            for (int j = 0; j < 3; j++)
            {
                cout << result[i][j]<<" ";
            }
            cout << "\n";
        }
        return 0;
    }
    int main()
    {
        int mat[3][3];
        for (int i = 0; i < 3; i++)
        {
            cout << "Enter data of row " << i + 1 << endl;
            for (int j = 0; j < 3; j++)
            {
                cin >> mat[i][j];
            }
        }
        cout << "The resultant matrix is " << endl;
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                cout<< mat[i][j]<<" ";
            }
            cout << endl;
        }
        cout << endl;
        transpose(mat);
        return 0;
    }
}

```



```

Microsoft Visual Studio Debug Console
Enter data of row 1
8
4
2
Enter data of row 2
7
11
9
Enter data of row 3
9
11
5
The resultant matrix is
8 4 2
7 11 9
9 11 5

While the transpose of the matrix is:
8 7 9
4 11 11
2 9 5

```

Task 4:

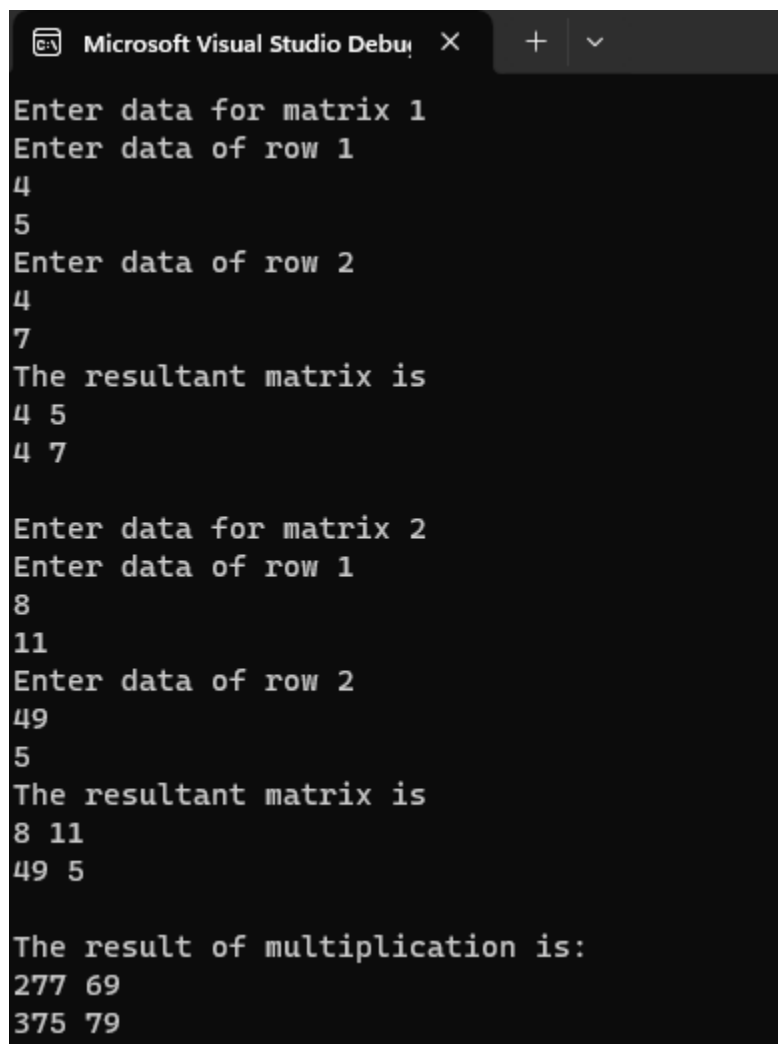
Using 2D arrays in C++, implement 3x3 matrix multiplication. Make a function.

```
#include <iostream>
using namespace std;
int multiply(int a[2][2], int b[2][2])
{
    int result[2][2] = { 0 };
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            for (int k = 0; k < 2; k++)
            {
                result[i][j] += a[i][k] * b[k][j];
            }
        }
    }
    cout << "The result of multiplication is: " << endl;
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            cout << result[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
int main()
{
    int mat1[2][2];
    cout << "Enter data for matrix 1\n";
    for (int i = 0; i < 2; i++)
    {
        cout << "Enter data of row " << i + 1 << endl;
        for (int j = 0; j < 2; j++)
        {
            cin >> mat1[i][j];
        }
    }
    cout << "The resultant matrix is " << endl;
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            cout << mat1[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl;
    int mat2[2][2];
    cout << "Enter data for matrix 2\n";
    for (int i = 0; i < 2; i++)
    {
        cout << "Enter data of row " << i + 1 << endl;
        for (int j = 0; j < 2; j++)
        {
```

```

        cin >> mat2[i][j];
    }
}
cout << "The resultant matrix is " << endl;
for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 2; j++)
    {
        cout << mat2[i][j] << " ";
    }
    cout << endl;
}
cout << endl;
multiply(mat1, mat2);
return 0;
}

```



The screenshot shows the Microsoft Visual Studio Debug Console with a dark background. The text is white and displays the output of a C++ program. The program prompts the user to enter data for two 2x2 matrices. For Matrix 1, the inputs are 4, 5, 4, and 7. For Matrix 2, the inputs are 8, 11, 49, and 5. The program then displays the resultant matrix for each input set and the final result of multiplying the two matrices.

```

Microsoft Visual Studio Debug Console
Enter data for matrix 1
Enter data of row 1
4
5
Enter data of row 2
4
7
The resultant matrix is
4 5
4 7

Enter data for matrix 2
Enter data of row 1
8
11
Enter data of row 2
49
5
The resultant matrix is
8 11
49 5

The result of multiplication is:
277 69
375 79

```

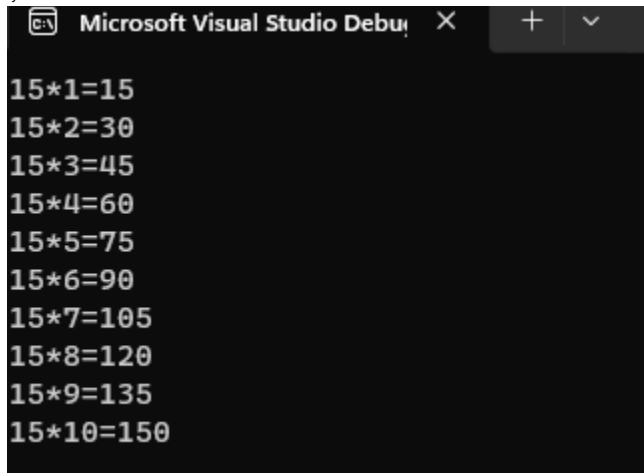
Task 5:

Print the multiplication table of 15 using recursion.

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

int table(int num, int i)
{
    if (i <= 10)
    {
        cout << num << "*" << i << "=" << num * i << endl;
        return table(num, i + 1);
    }
    else
    {
        return 1;
    }
}

int main()
{
    table(15, 1);
    return 0;
}
```

The screenshot shows the Microsoft Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio Debug" with a close button (X) and window control buttons (+ and v). The console output displays the multiplication table for 15, with each row showing the number 15 multiplied by integers from 1 to 10. The output is as follows:

```
15*1=15
15*2=30
15*3=45
15*4=60
15*5=75
15*6=90
15*7=105
15*8=120
15*9=135
15*10=150
```

Home Task

Task 1:

Write a C++ program to take inverse of a 3x3 matrix using its determinant and adjoint.

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

int Determinant(int a[2][2])
{
    return (a[0][0] * a[1][1]) - (a[0][1] * a[1][0]);
}

int Adjoint(int b[2][2])
{

```

```

    int temp = 0;
    temp = b[0][0];
    b[0][0] = b[1][1];
    b[1][1] = temp;
    temp = -b[0][1];
    b[0][1] = -b[1][0];
    b[1][0] = temp;
    return 0;
}
int Inverse(int a[2][2])
{
    cout << "The inverse is :" << endl;
    cout << 1 << "/" << Determinant(a) << " multiplied by the matrix :" << endl;
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
int InputMatrix(int b[2][2])
{
    for (int i = 0; i < 2; i++)
    {
        cout << "Enter data for row " << i + 1 << endl;
        for (int j = 0; j < 2; j++)
        {
            cin >> b[i][j];
        }
    }
    return 0;
}
int main()
{
    int mat[2][2];
    InputMatrix(mat);
    cout << "The input matrix :" << endl;
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            cout << mat[i][j] << " ";
        }
        cout << endl;
    }
    Determinant(mat);
    Adjoint(mat);
    Inverse(mat);
    return 0;
}

```



Microsoft Visual Studio Debug Console



Enter data for row 1

7

8

Enter data for row 2

4

2

The input matrix :

7 8

4 2

The inverse is :

1/-18 multiplied by the matrix :

2 -4

-8 7