School Of Mechanical & Manufacturing Engineering, NUST

Department of Mechanical Engineering

# CS-114 - Fundamental of Programing

## Lab AND Home Tasks 9

**Course Instructor:** Dr Jawad Khan

**Lab Instructor:** Muhammad Affan

**Student Name:** **Ahmed Adil Hussain**

**CMS ID:** **477537**

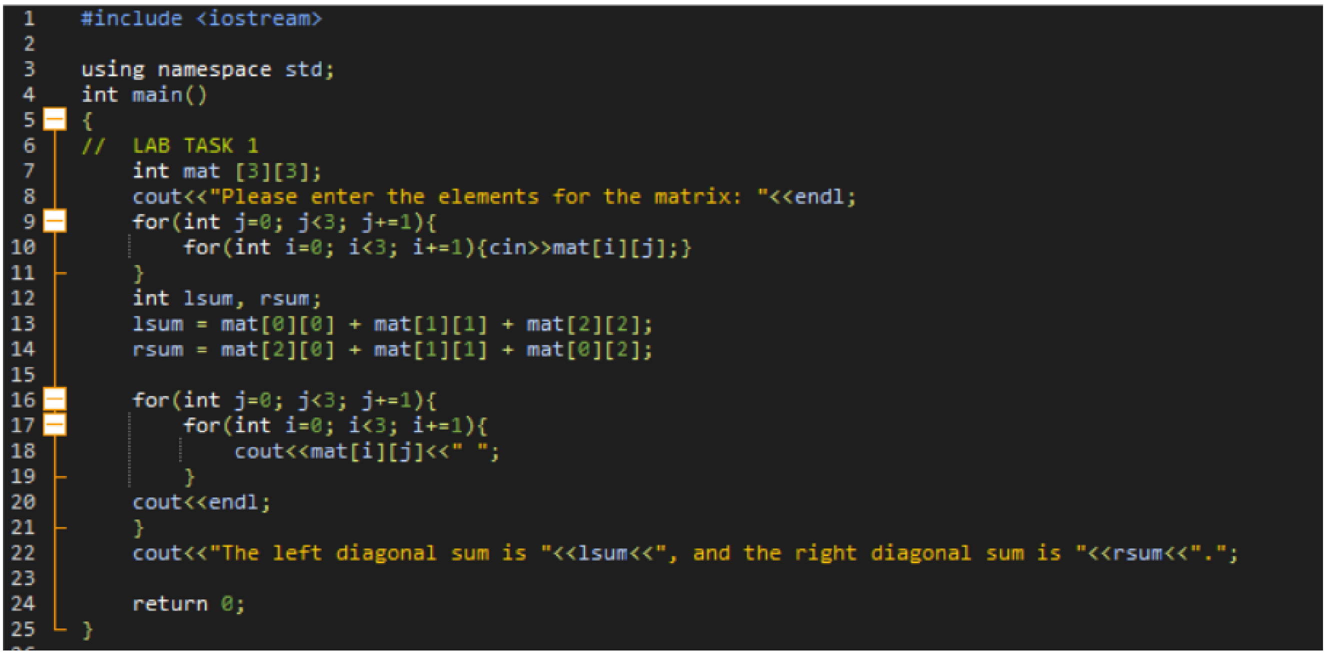
**DATE:**

**6 December 2023**

## LAB TASK 1:

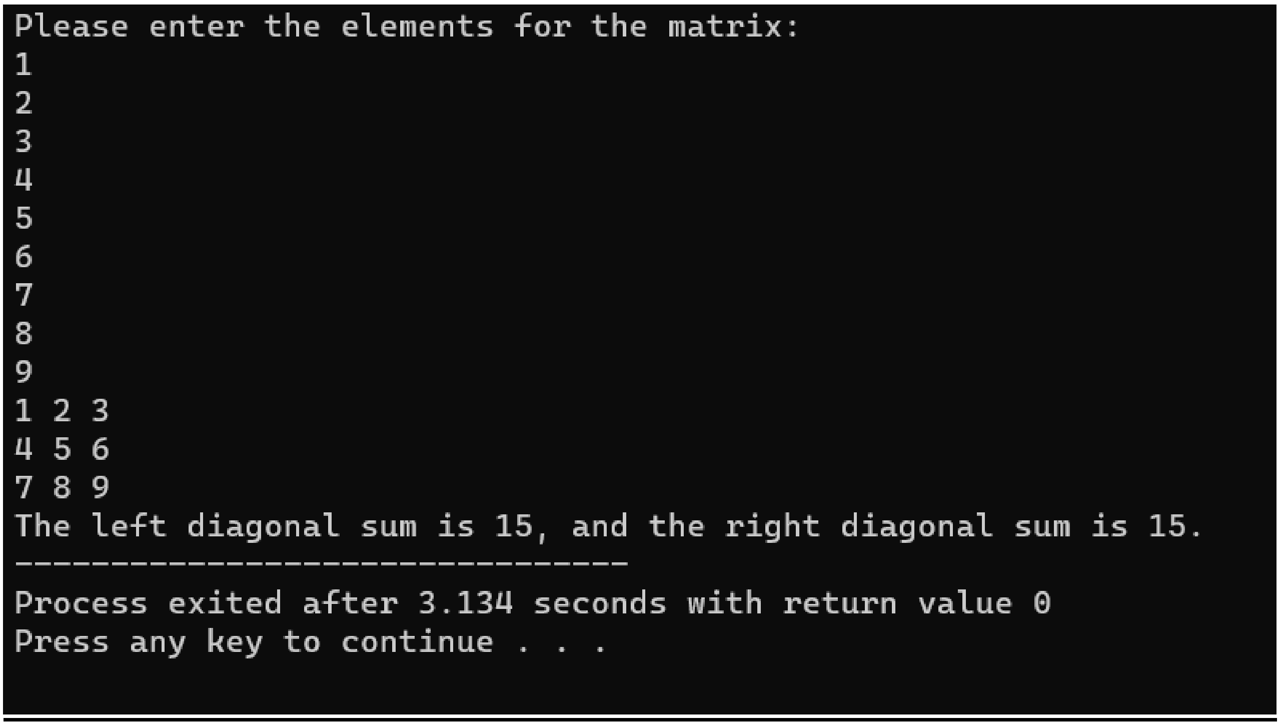
1. Make 2D Array in C++ and print left diagonal and right diagonal sum of a 3x3 matrix.

## Code:



```cpp
#include <iostream>

using namespace std;
int main()
{
//  LAB TASK 1
    int mat [3][3];
    cout<<"Please enter the elements for the matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){cin>>mat[i][j];}
    }
    int lsum, rsum;
    lsum = mat[0][0] + mat[1][1] + mat[2][2];
    rsum = mat[2][0] + mat[1][1] + mat[0][2];

    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            cout<<mat[i][j]<<" ";
        }
    cout<<endl;
    }
    cout<<"The left diagonal sum is "<<lsum<<", and the right diagonal sum is "<<rsum<<".";

    return 0;
}
```

## Result:

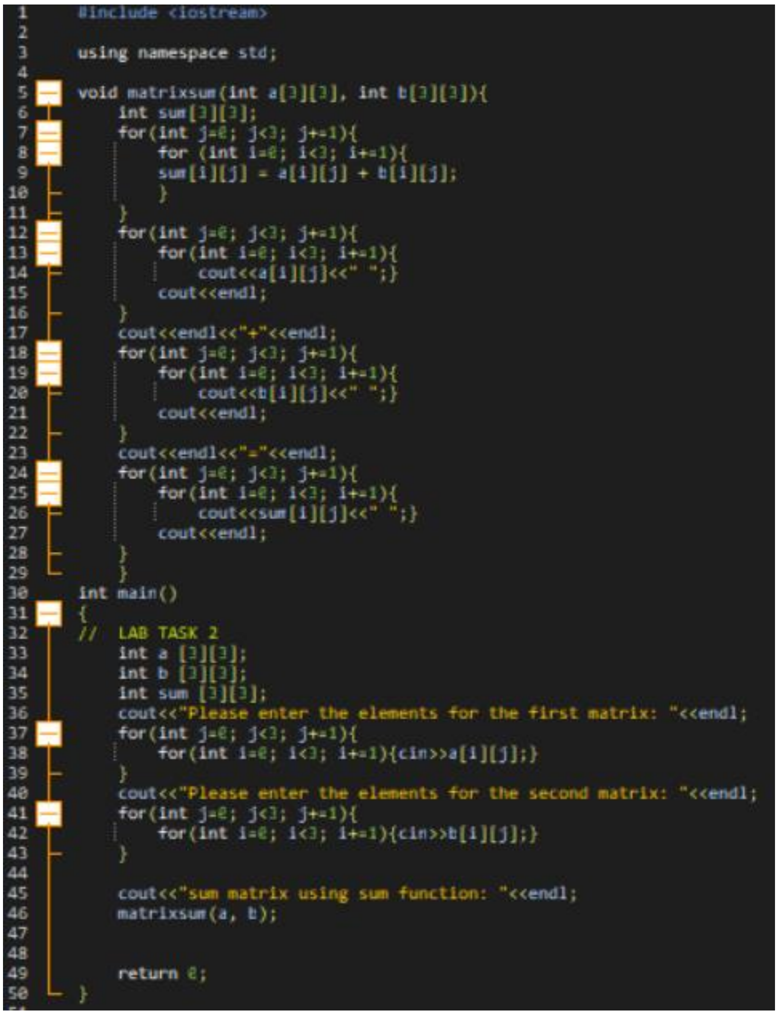


```
Please enter the elements for the matrix:
1
2
3
4
5
6
7
8
9
1 2 3
4 5 6
7 8 9
The left diagonal sum is 15, and the right diagonal sum is 15.
--------------------------------
Process exited after 3.134 seconds with return value 0
Press any key to continue . . .
```

## LAB TASK 2:

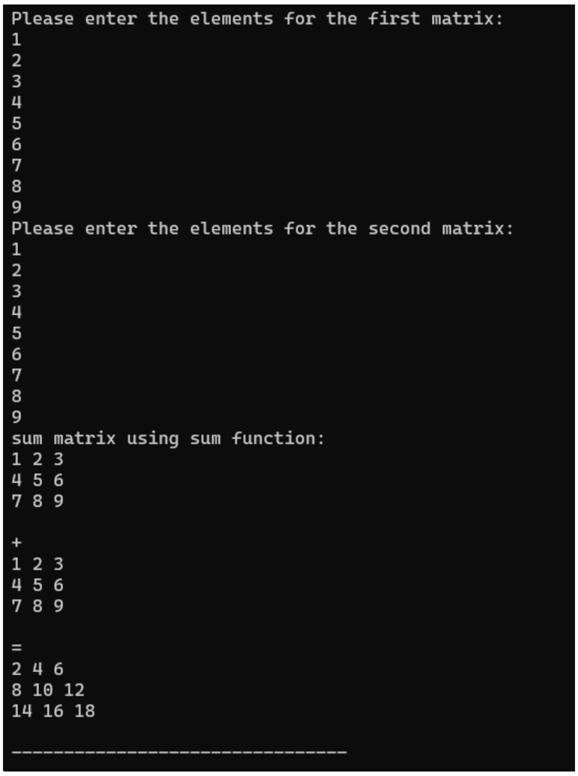
2. Write a function to add two 2D arrays of size 3x3.

**Code:**



```
#include <iostream>

using namespace std;

void matrixsum(int a[3][3], int b[3][3]){
    int sum[3][3];
    for(int j=0; j<3; j+=1){
        for (int i=0; i<3; i+=1){
        sum[i][j] = a[i][j] + b[i][j];
        }
    }
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            cout<<a[i][j]<<" ";}
        cout<<endl;
    }
    cout<<endl<<"+"<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            cout<<b[i][j]<<" ";}
        cout<<endl;
    }
    cout<<endl<<"="<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            cout<<sum[i][j]<<" ";}
        cout<<endl;
    }
    }
int main()
{
//  LAB TASK 2
    int a [3][3];
    int b [3][3];
    int sum [3][3];
    cout<<"Please enter the elements for the first matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){cin>>a[i][j];}
    }
    cout<<"Please enter the elements for the second matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){cin>>b[i][j];}
    }

    cout<<"sum matrix using sum function: "<<endl;
    matrixsum(a, b);


    return 0;
}
```

**Result:**

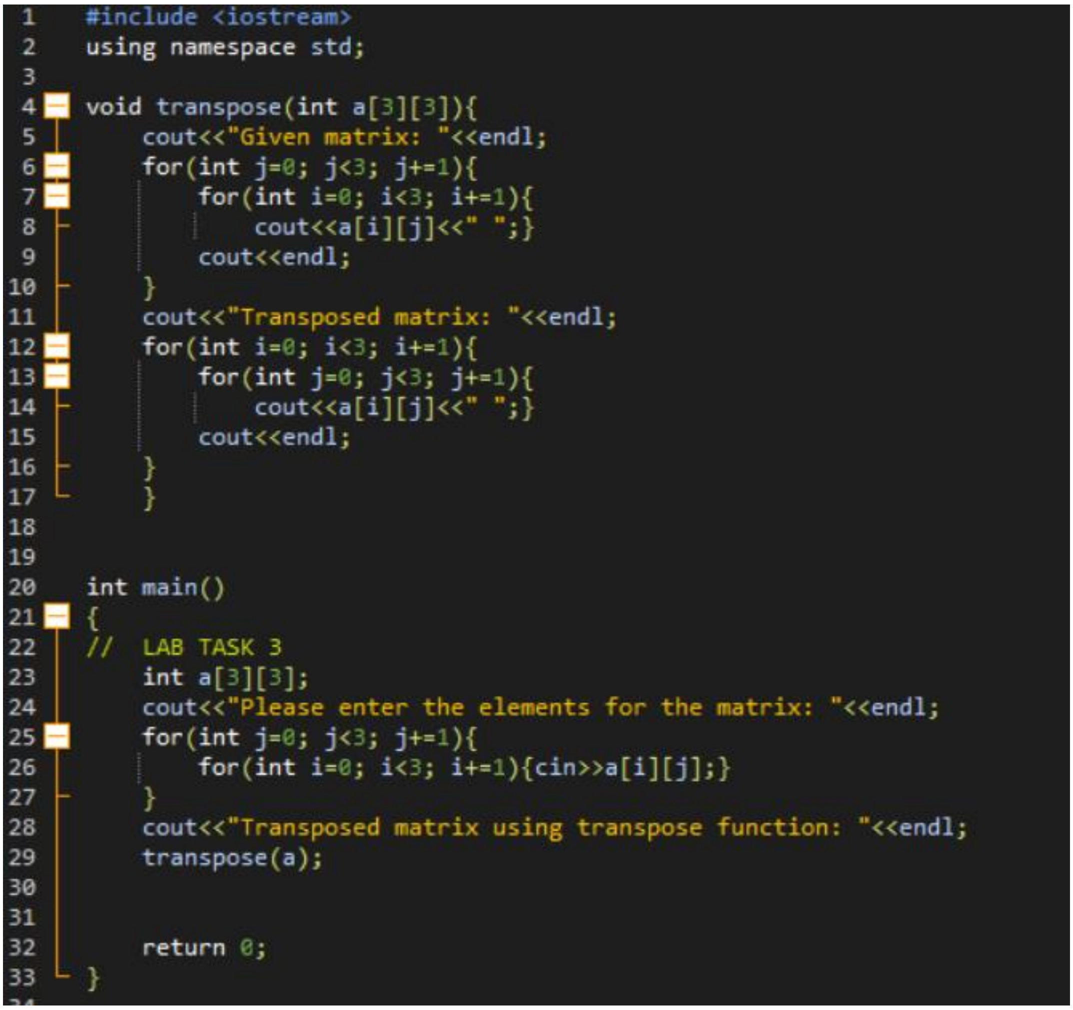


```
Please enter the elements for the first matrix:
1
2
3
4
5
6
7
8
9
Please enter the elements for the second matrix:
1
2
3
4
5
6
7
8
9
sum matrix using sum function:
1 2 3
4 5 6
7 8 9

+
1 2 3
4 5 6
7 8 9

=
2 4 6
8 10 12
14 16 18

--------------------------------
```

**LAB TASK 3:**

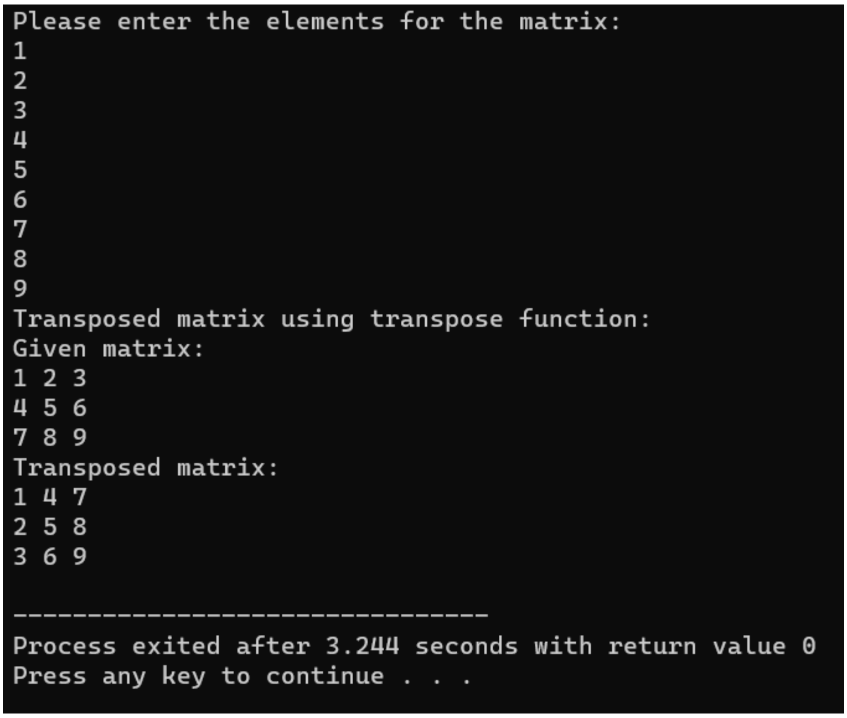
3. Using 2D arrays in C++, take transpose of a 3x3 matrix. Make a transpose function

**Code:**



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

void transpose(int a[3][3]){
    cout<<"Given matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            cout<<a[i][j]<<" ";}
        cout<<endl;
    }
    cout<<"Transposed matrix: "<<endl;
    for(int i=0; i<3; i+=1){
        for(int j=0; j<3; j+=1){
            cout<<a[i][j]<<" ";}
        cout<<endl;
    }
    }


int main()
{
//  LAB TASK 3
    int a[3][3];
    cout<<"Please enter the elements for the matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){cin>>a[i][j];}
    }
    cout<<"Transposed matrix using transpose function: "<<endl;
    transpose(a);


    return 0;
}
```

**Result:**

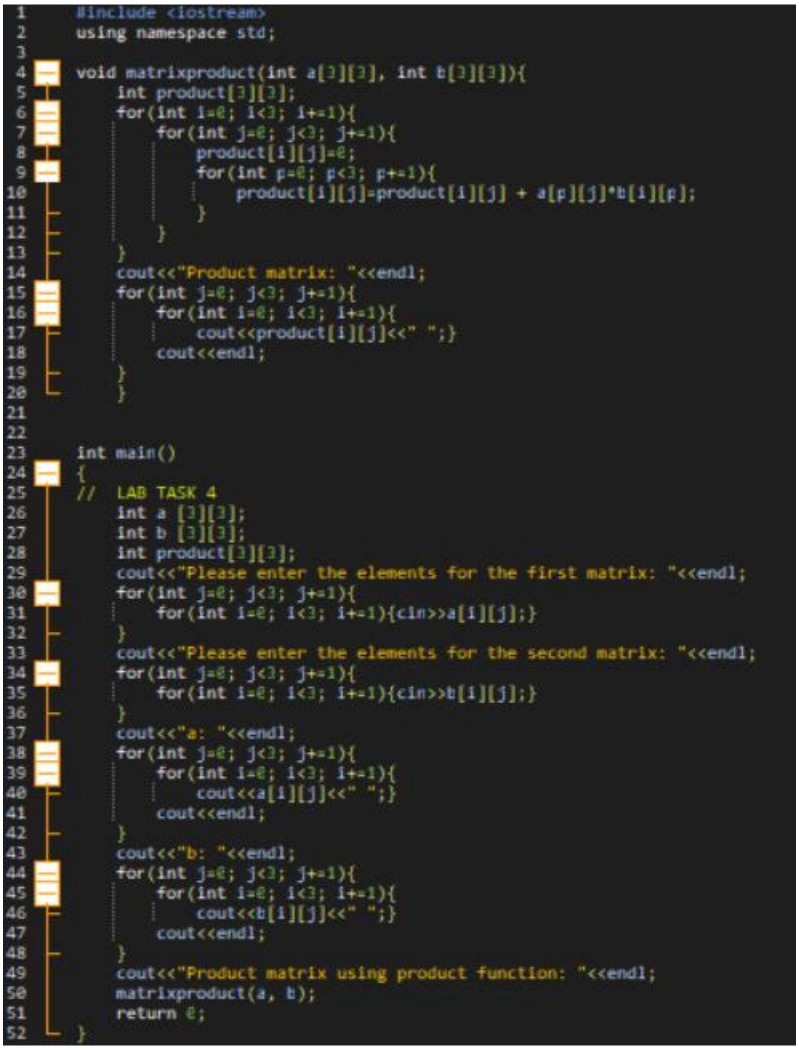


```
Please enter the elements for the matrix:
1
2
3
4
5
6
7
8
9
Transposed matrix using transpose function:
Given matrix:
1 2 3
4 5 6
7 8 9
Transposed matrix:
1 4 7
2 5 8
3 6 9

--------------------------------
Process exited after 3.244 seconds with return value 0
Press any key to continue . . .
```

## LAB TASK 4:

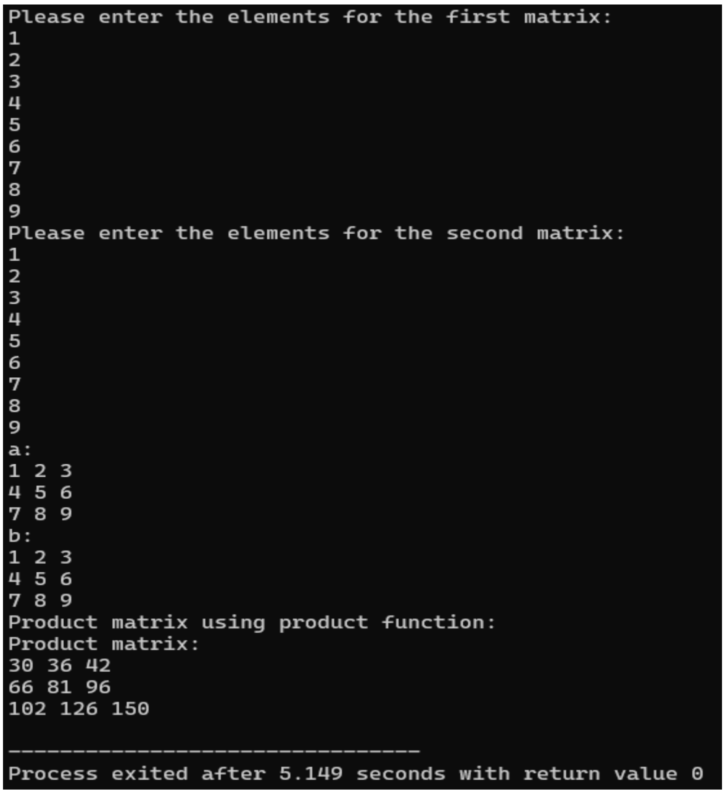
4. Using 2D arrays in C++, implement 3x3 matrix multiplication. Make a function.

## Code:



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

void matrixproduct(int a[3][3], int b[3][3]){
    int product[3][3];
    for(int i=0; i<3; i+=1){
        for(int j=0; j<3; j+=1){
            product[i][j]=0;
            for(int p=0; p<3; p+=1){
                product[i][j]=product[i][j] + a[p][j]*b[i][p];
            }
        }
    }
    cout<<"Product matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            cout<<product[i][j]<<" ";}
        cout<<endl;
    }
    }


int main()
{
//  LAB TASK 4
    int a [3][3];
    int b [3][3];
    int product[3][3];
    cout<<"Please enter the elements for the first matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){cin>>a[i][j];}
    }
    cout<<"Please enter the elements for the second matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){cin>>b[i][j];}
    }
    cout<<"a: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            cout<<a[i][j]<<" ";}
        cout<<endl;
    }
    cout<<"b: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            cout<<b[i][j]<<" ";}
        cout<<endl;
    }
    cout<<"Product matrix using product function: "<<endl;
    matrixproduct(a, b);
    return 0;
}
```

## Result:

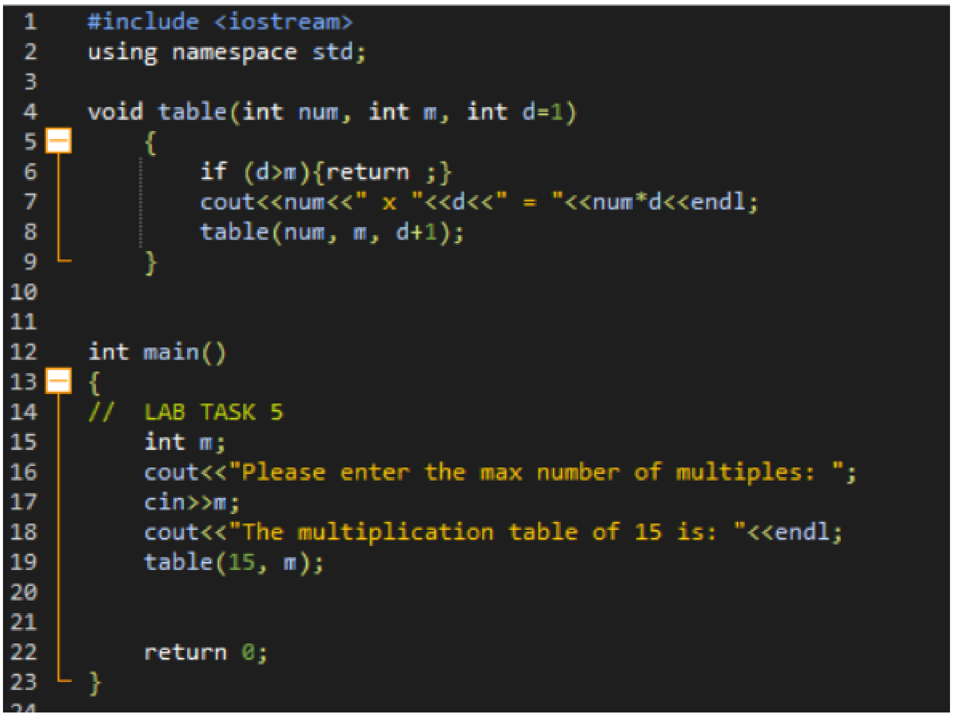


```
Please enter the elements for the first matrix:
1
2
3
4
5
6
7
8
9
Please enter the elements for the second matrix:
1
2
3
4
5
6
7
8
9
a:
1 2 3
4 5 6
7 8 9
b:
1 2 3
4 5 6
7 8 9
Product matrix using product function:
Product matrix:
30 36 42
66 81 96
102 126 150

--------------------------------
Process exited after 5.149 seconds with return value 0
```

**LAB TASK 5:**

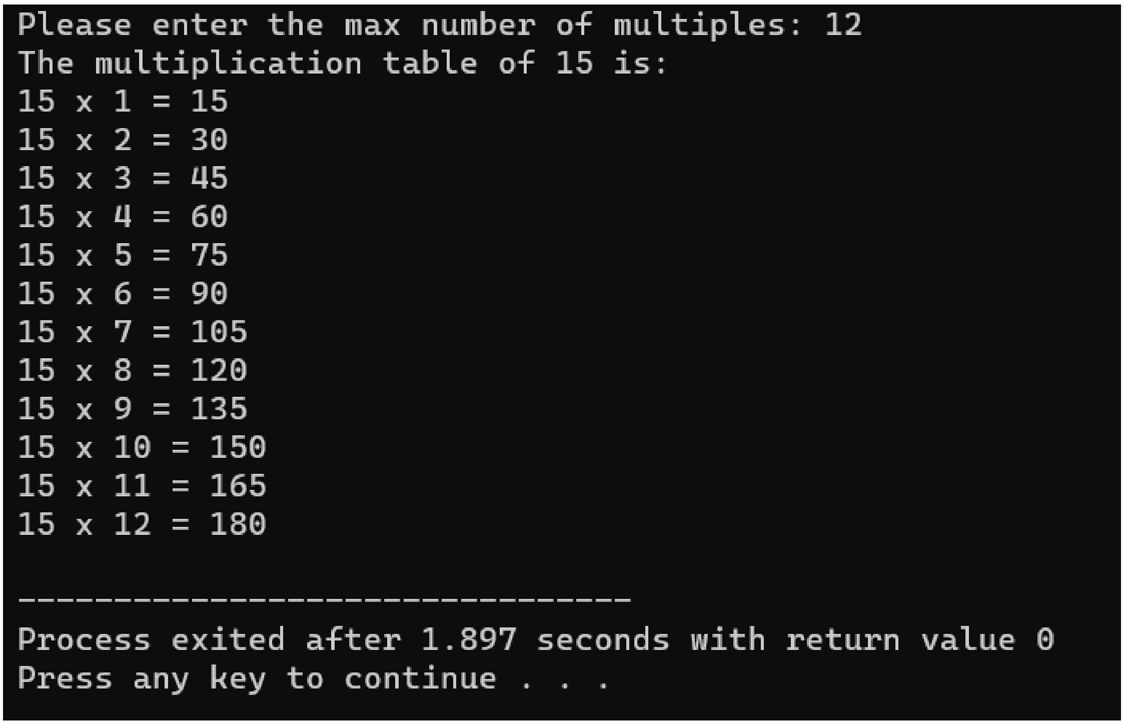
5. Print the multiplication table of 15 using recursion.

**Code:**



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

void table(int num, int m, int d=1)
    {
        if (d>m){return ;}
        cout<<num<<" x "<<d<<" = "<<num*d<<endl;
        table(num, m, d+1);
    }


int main()
{
//  LAB TASK 5
    int m;
    cout<<"Please enter the max number of multiples: ";
    cin>>m;
    cout<<"The multiplication table of 15 is: "<<endl;
    table(15, m);


    return 0;
}
```

**Result:**

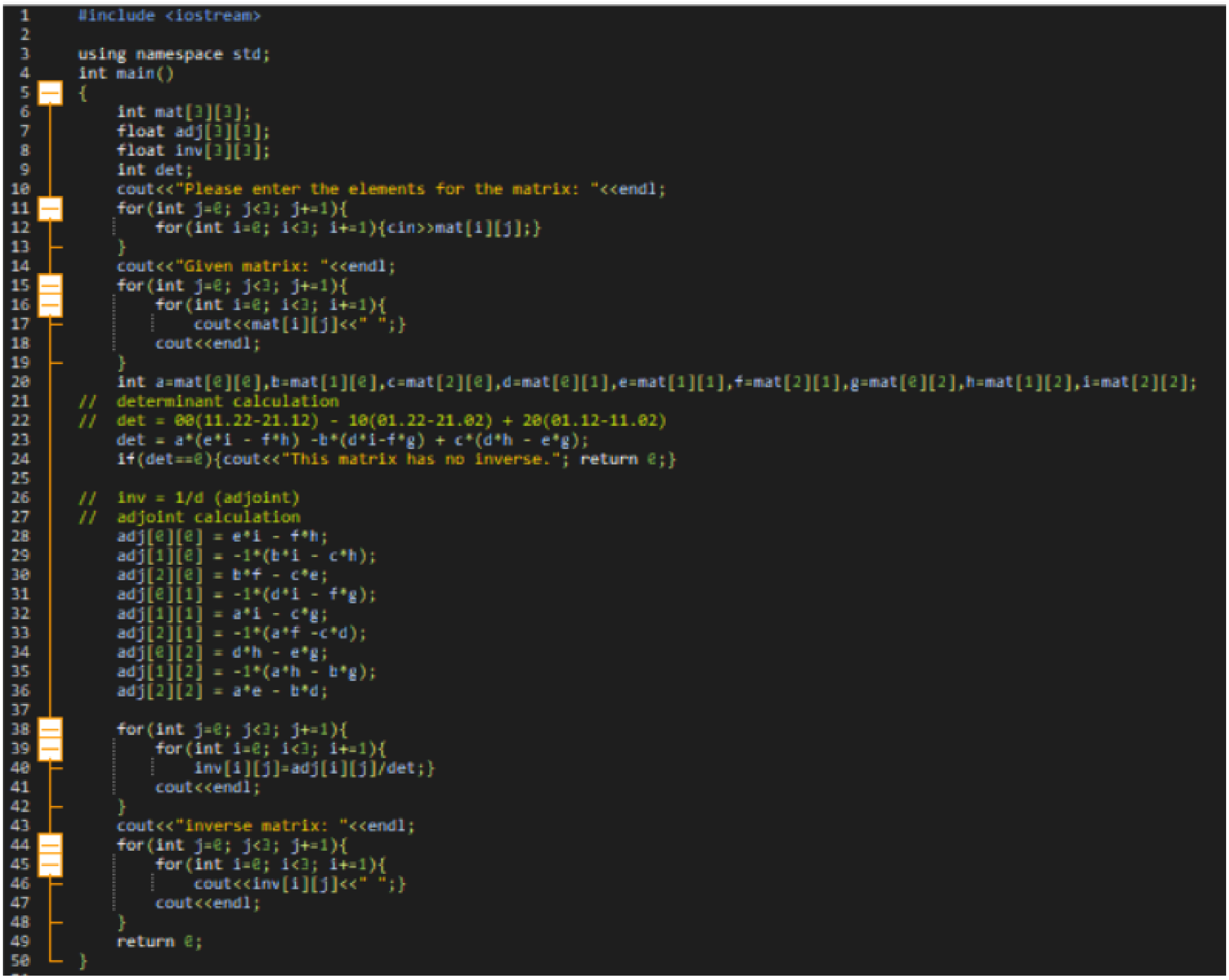


```
Please enter the max number of multiples: 12
The multiplication table of 15 is:
15 x 1 = 15
15 x 2 = 30
15 x 3 = 45
15 x 4 = 60
15 x 5 = 75
15 x 6 = 90
15 x 7 = 105
15 x 8 = 120
15 x 9 = 135
15 x 10 = 150
15 x 11 = 165
15 x 12 = 180

--------------------------------
Process exited after 1.897 seconds with return value 0
Press any key to continue . . .
```

## HOME TASK:

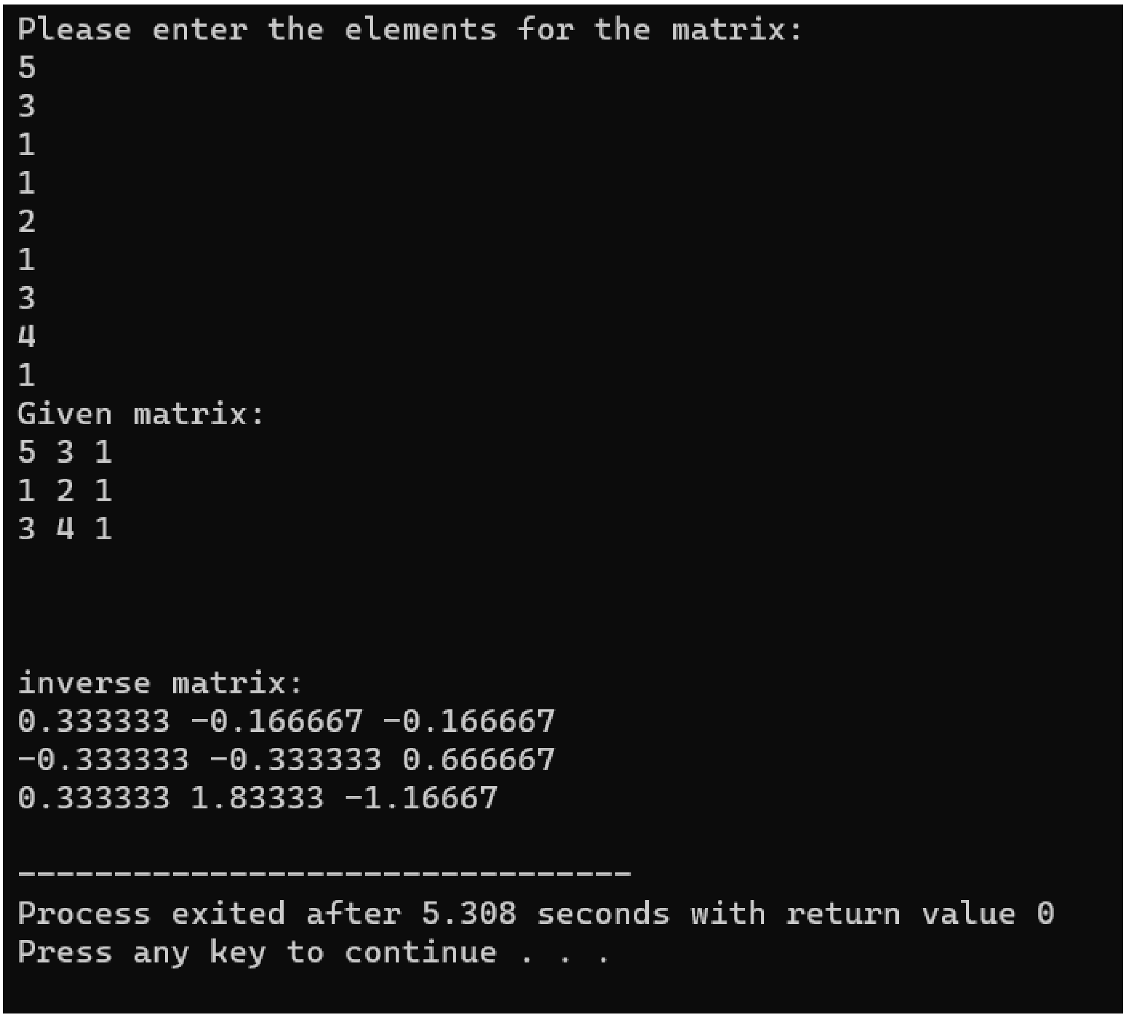
Write a C++ program to take inverse of a 3x3 matrix using its determinant and adjoint..

## Code: (result next page)



```cpp
#include <iostream>

using namespace std;
int main()
{
    int mat[3][3];
    float adj[3][3];
    float inv[3][3];
    int det;
    cout<<"Please enter the elements for the matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){cin>>mat[i][j];}
    }
    cout<<"Given matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            cout<<mat[i][j]<<" ";}
        cout<<endl;
    }
    int a=mat[0][0],b=mat[1][0],c=mat[2][0],d=mat[0][1],e=mat[1][1],f=mat[2][1],g=mat[0][2],h=mat[1][2],i=mat[2][2];
//  determinant calculation
//  det = 00(11.22-21.12) - 10(01.22-21.02) + 20(01.12-11.02)
    det = a*(e*i - f*h) -b*(d*i-f*g) + c*(d*h - e*g);
    if(det==0){cout<<"This matrix has no inverse."; return 0;}

//  inv = 1/d (adjoint)
//  adjoint calculation
    adj[0][0] = e*i - f*h;
    adj[1][0] = -1*(b*i - c*h);
    adj[2][0] = b*f - c*e;
    adj[0][1] = -1*(d*i - f*g);
    adj[1][1] = a*i - c*g;
    adj[2][1] = -1*(a*f -c*d);
    adj[0][2] = d*h - e*g;
    adj[1][2] = -1*(a*h - b*g);
    adj[2][2] = a*e - b*d;

    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            inv[i][j]=adj[i][j]/det;}
        cout<<endl;
    }
    cout<<"inverse matrix: "<<endl;
    for(int j=0; j<3; j+=1){
        for(int i=0; i<3; i+=1){
            cout<<inv[i][j]<<" ";}
        cout<<endl;
    }
    return 0;
}
```

**Result:**



```
Please enter the elements for the matrix:
5
3
1
1
2
1
3
4
1
Given matrix:
5 3 1
1 2 1
3 4 1



inverse matrix:
0.333333 -0.166667 -0.166667
-0.333333 -0.333333 0.666667
0.333333 1.83333 -1.16667

--------------------------------
Process exited after 5.308 seconds with return value 0
Press any key to continue . . .
```