NATIONAL UNIVERSITY OF SCIENCE & TECNOLOGY

SCHOOL OF MECHANICAL AND MANUFACTURING ENGINEERING

[FUNDAMENTALS OF PROGRAMMING –(LAB)]

MANUAL # 9

SEMESTER # 01

CLASS: - ME-15 [SEC A]

NAME: TALHA MATEEN

ROLL NO.: 454713

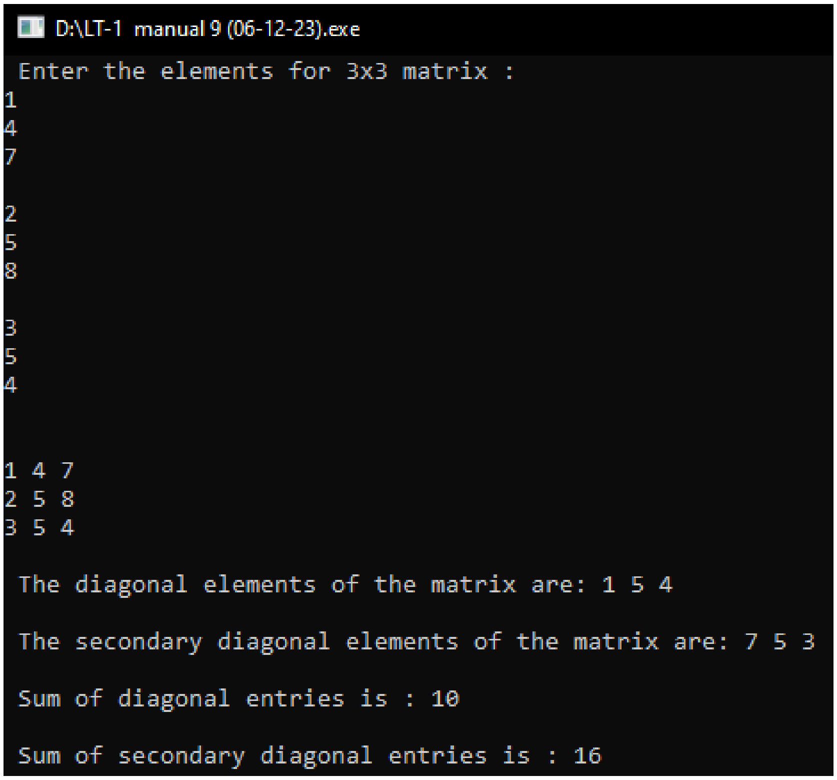
SUBMITTED TO: M. AFFAN TARIQ

LAB TASK

QUESTION NO.1:

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

#include <bits/stdc++.h>
using namespace std;

int main()
{
int sum_dia=0 , sum_sec ;
//sum_dia = sum of diagonal entries :: sum_sec = sum of secondary diagonal entries.

int a[3][3];          // taking a 3x3 matrix.
cout<<" Enter the elements for 3x3 matrix :\n";
for(int i=0; i<3;i++){
  for(int j=0; j<3 ;j++){
   cin>>a[i][j];     //getting entries by user.
  }
  cout<<endl;
}
//displaying matrix in order
for(int i=0; i<3;i++){
  cout<<endl;

  for(int j=0; j<3 ;j++){
    cout<<a[i][j]<<" ";
}}
 cout<<endl<<endl;
             //printing diagonal elements
  cout << " The diagonal elements of the matrix are: ";
  for(int i = 0; i < 3; i++) {
    cout << a[i][i] << " ";
  }
  cout << endl<<endl;
              //printing secondary diagonal elements
  cout << " The secondary diagonal elements of the matrix are: ";
  for(int i = 0; i < 3; i++) {
    cout << a[i][2-i] << " ";
  }
  cout <<endl<< endl;

  for(int i = 0; i < 3; i++) {
  sum_dia += a[i][i];          // using loop for sum of diagonal entries
  }
```

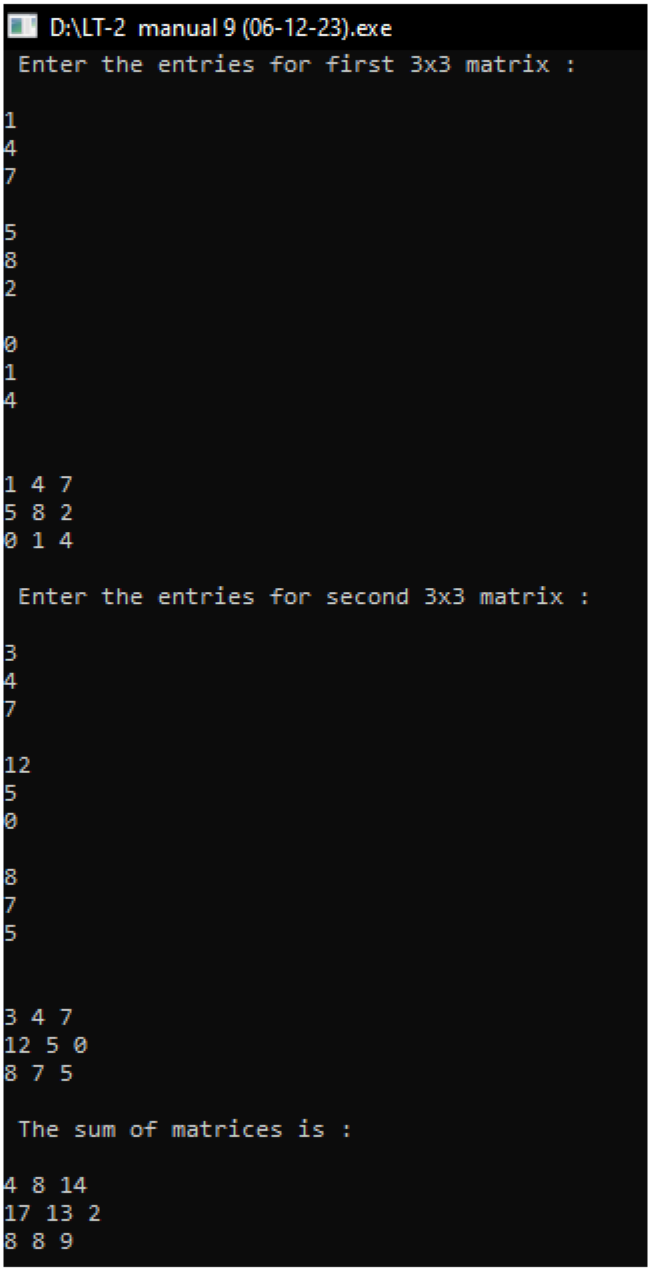
```
   cout<<" Sum of diagonal entries is : "<<sum_dia<<endl;
   cout << endl;

  // using loop for sum of secondary diagonal entries
   for(int i = 0; i < 3; i++) {
   sum_sec += a[i][2-i];
   }
   cout<<" Sum of secondary diagonal entries is : "<<sum_sec;

   return 0;   //end program
}
```



```
 Enter the elements for 3x3 matrix :
1
4
7

2
5
8

3
5
4



1 4 7
2 5 8
3 5 4

 The diagonal elements of the matrix are: 1 5 4

 The secondary diagonal elements of the matrix are: 7 5 3

 Sum of diagonal entries is : 10

 Sum of secondary diagonal entries is : 16
```

QUESTION NO.2:

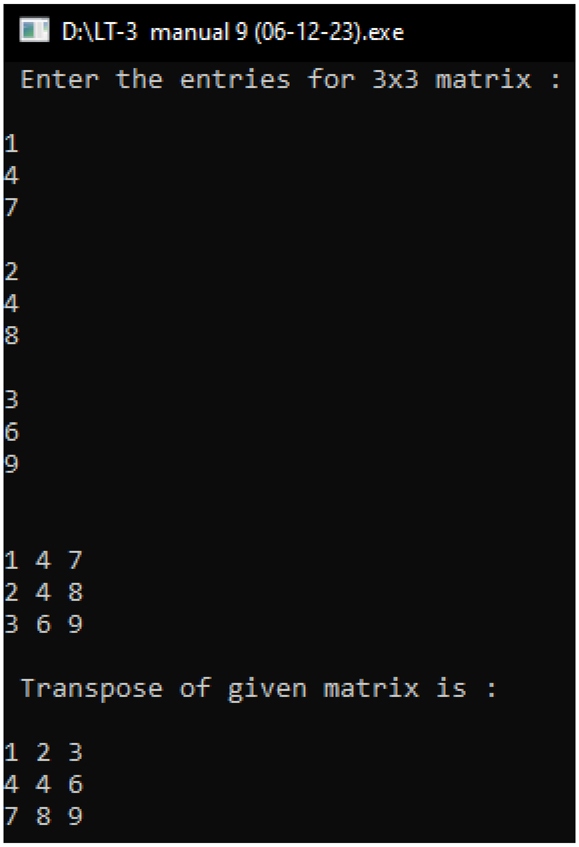
```
//Write a function to add two 2D arrays of size 3x3

#include <bits/stdc++.h>

using namespace std;

int main()
{
int a[3][3];              //taking a 3x3 matrix
cout<<" Enter the entries for first 3x3 matrix : \n";
cout<<endl;

for(int i=0; i<3;i++){
    for(int j=0; j<3 ;j++){
        cin>>a[i][j];      // input from user
    }
    cout<<endl;
}

//displaying matrix in order
for(int i=0; i<3;i++){
    cout<<endl;

    for(int j=0; j<3 ;j++){
       cout<<a[i][j]<<" ";
}}
  cout<<endl<<endl;

  //taking another 3x3 matrix.
  int b[3][3];
  cout<<" Enter the entries for second 3x3 matrix : \n";
  cout<<endl;

for(int i=0; i<3;i++){
    for(int j=0; j<3 ;j++){
        cin>>b[i][j];      // input from user
    }
    cout<<endl;
}
//displaying matrix in order.
for(int i=0; i<3;i++){
    cout<<endl;

    for(int j=0; j<3 ;j++){
       cout<<b[i][j]<<" ";
}}
cout<<endl<<endl;

//adding the matrices.
int sum =0 ;
cout<<" The sum of matrices is :"<<endl;
for(int i=0 ; i<3 ; i ++){
    cout<<endl;

    //uing loop to display the sum
    for( int j = 0; j< 3 ; j ++){
        sum= a[i][j]+b[i][j];
cout<<sum<<" ";

}}
return 0;   //end program
}
```



```
 Enter the entries for first 3x3 matrix :

1
4
7

5
8
2

0
1
4


1 4 7
5 8 2
0 1 4

 Enter the entries for second 3x3 matrix :

3
4
7

12
5
0

8
7
5


3 4 7
12 5 0
8 7 5

 The sum of matrices is :

4 8 14
17 13 2
8 8 9
```

QUESTION NO.3:

```
//Using 2D arrays in C++, take transpose of a 3x3 matrix

#include <bits/stdc++.h>

using namespace std;

int main()
{
int a[3][3];              //taking a 3x3 matrix
cout<<" Enter the entries for 3x3 matrix : \n";
cout<<endl;

for(int i=0; i<3;i++){
   for(int j=0; j<3 ;j++){
      cin>>a[i][j];      // input from user
   }
   cout<<endl;
}

//displaying matrix in order
for(int i=0; i<3;i++){      //rows
   cout<<endl;

   for(int j=0; j<3 ;j++){   //columns
     cout<<a[i][j]<<" ";
}}
 cout<<endl<<endl;

// for transpose swaping the rows with column of matrix.
cout<<" Transpose of given matrix is : "<<endl;

 //now j represents number of  rows
   for(int j=0; j<3 ;j++){
      cout<<endl;
 //now i represents number of columns
      for(int i=0; i<3;i++){
         cout<<a[i][j]<<" ";
   } }

return 0;   //end program
}
```

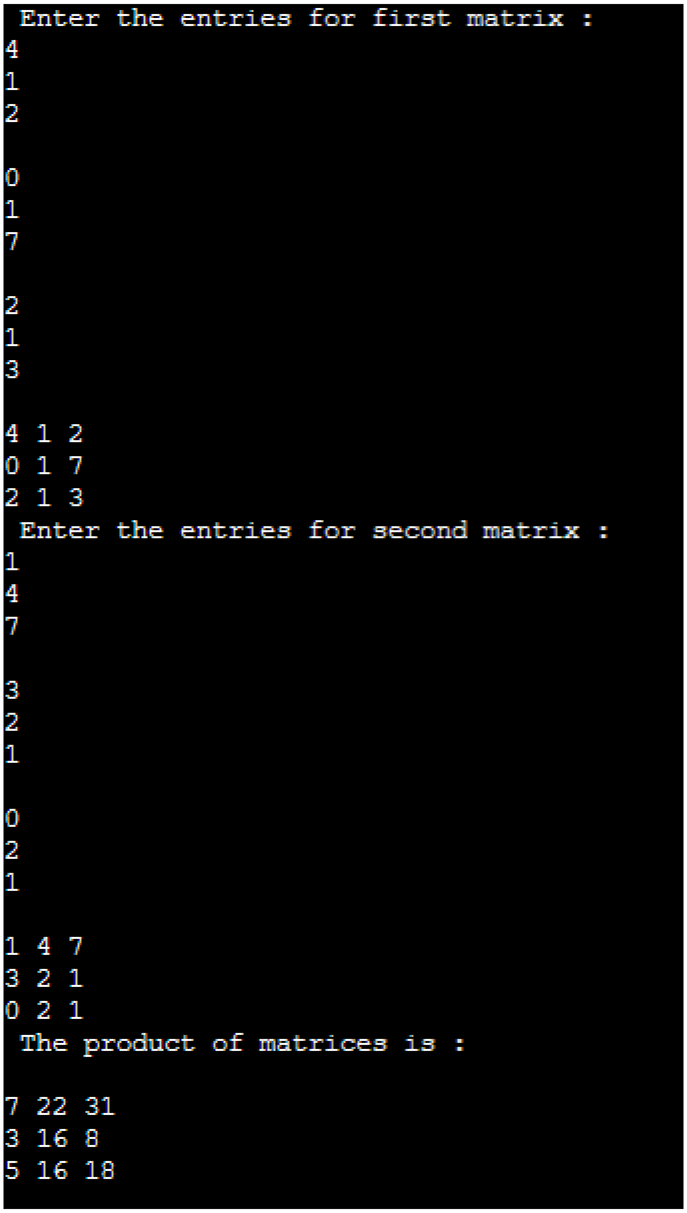


```
 Enter the entries for 3x3 matrix :

1
4
7

2
4
8

3
6
9


1 4 7
2 4 8
3 6 9

 Transpose of given matrix is :

1 2 3
4 4 6
7 8 9
```

```
}
```

QUESTION NO.4:

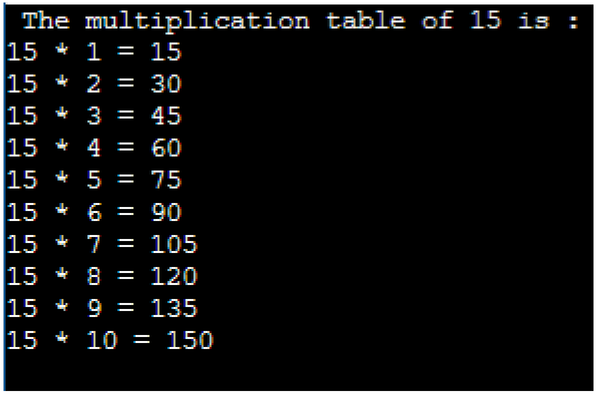
```
//Using 2D arrays in C++, implement 3x3 matrix multiplication.

#include <bits/stdc++.h>

using namespace std;

int main()
{
int m1[3][3];        // taking a 3x3 matrix
cout<<" Enter the entries for first matrix : "<<endl;

for(int i = 0; i < 3 ; i ++){

    for(int j = 0 ; j < 3 ; j ++){
        cin>>m1[i][j];    //input from user.
    }
    cout<<endl;
}
// displaying out first 3x3 matrix .
for(int i = 0; i < 3 ; i ++){

    for(int j = 0 ; j < 3 ; j ++){
        cout<<m1[i][j]<<" ";
    }
    cout<<endl;
}

int m2[3][3];       // taking another 3x3 matrix.
cout<<" Enter the entries for second matrix : "<<endl;

for(int i = 0; i < 3 ; i ++){

    for(int j = 0 ; j < 3 ; j ++){
        cin>>m2[i][j];      //input from user.
    }
    cout<<endl;
}
//printing out second matrix of order 3x3
for(int i = 0; i < 3 ; i ++){

    for(int j = 0 ; j < 3 ; j ++){
        cout<<m2[i][j]<<" ";
    }
    cout<<endl;
}
//here we have resulting matrix i.e multiplication of given matrices.
int p[3][3];

//using loops for multiplying and arranging output matrix.
for(int i = 0; i < 3 ; i ++){

   for(int j = 0 ; j < 3 ; j ++){

      for(int k = 0; k < 3 ; k ++){
   p[i][j]+=m1[i][k]*m2[k][j];     // multiplying matrices.
}
} }
// printing the new matrix.
cout<<" The product of matrices is : "<<endl;
for(int i = 0 ; i < 3 ; i ++){
   cout <<endl;
   for(int j = 0 ; j < 3 ; j ++){
      cout<<p[i][j]<<" ";
   }
}
   return 0; // end program.
}
```



```
 Enter the entries for first matrix :
4
1
2

0
1
7

2
1
3

4 1 2
0 1 7
2 1 3
 Enter the entries for second matrix :
1
4
7

3
2
1

0
2
1

1 4 7
3 2 1
0 2 1
 The product of matrices is :

7 22 31
3 16 8
5 16 18
```

QUESTION NO.5:

```
#include <bits/stdc++.h>
using namespace std;

void mul_table(int n, int i) {
   if (i > 10) return;
   cout << n << " * " << i << " = " << n * i << endl;
   return mul_table(n, i + 1);
}

int main() {

   cout<<"The multiplication table of 15 is : "<<endl;
   int n = 15;
   mul_table(n, 1);
   return 0;
}
```

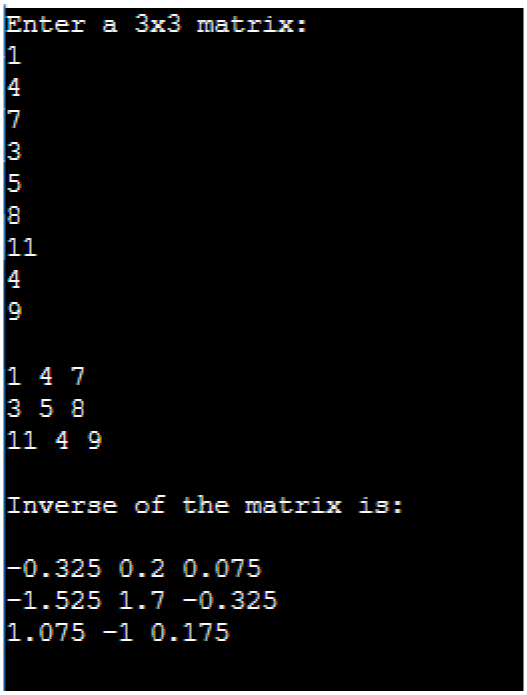


```
 The multiplication table of 15 is :
15 * 1 = 15
15 * 2 = 30
15 * 3 = 45
15 * 4 = 60
15 * 5 = 75
15 * 6 = 90
15 * 7 = 105
15 * 8 = 120
15 * 9 = 135
15 * 10 = 150
```

HOME TASK

QUESTION NO.1:

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    double a[3][3], adj[3][3], inv[3][3], det;
    int i, j;

    cout << "Enter a 3x3 matrix:" << endl;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            cin >> a[i][j];
        }
    }

    adj[0][0] = a[1][1] * a[2][2] - a[2][1] * a[1][2];
    adj[0][1] = a[0][2] * a[2][1] - a[0][1] * a[2][2];
    adj[0][2] = a[0][1] * a[1][2] - a[0][2] * a[1][1];
    adj[1][0] = a[1][2] * a[2][0] - a[1][0] * a[2][2];
    adj[1][1] = a[0][0] * a[2][2] - a[0][2] * a[2][0];
    adj[1][2] = a[1][0] * a[0][2] - a[0][0] * a[1][2];
    adj[2][0] = a[1][0] * a[2][1] - a[2][0] * a[1][1];
    adj[2][1] = a[2][0] * a[0][1] - a[0][0] * a[2][1];
    adj[2][2] = a[0][0] * a[1][1] - a[1][0] * a[0][1];

    det = a[0][0] * adj[0][0] + a[0][1] * adj[0][1] + a[0][2] * adj[0][2];

//displaying matrix in order
for(int i=0; i<3;i++){
    cout<<endl;

    for(int j=0; j<3 ;j++){
       cout<<a[i][j]<<" ";
}}
  cout<<endl<<endl;

    if (det == 0) {
        cout << "Inverse of the matrix does not exist." << endl;
        return 0;
    }

    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            inv[i][j] = adj[i][j] / det;

        }
    }

    cout << "Inverse of the matrix is:\n" << endl;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            cout << inv[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}
```



```
Enter a 3x3 matrix:
1
4
7
3
5
8
11
4
9

1 4 7
3 5 8
11 4 9

Inverse of the matrix is:

-0.325 0.2 0.075
-1.525 1.7 -0.325
1.075 -1 0.175
```