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()
 4 🖯 {
         int sum1=0,sum2=0; // Declaring 2D array
int arr[3][3]; // Initialize 2D array using loop
cout<<"Enter elements of matrix: "<<endl;
                                                                                                                 D:\FOP\lab manuals\prac.exe
                                                                                                                Enter elements of matrix:
         for (int i=0:i<3:i++)
                                                                                                               Enter element[0][0]: 1
Enter element[0][1]: 2
10 T
           for (int j=0;j<3;j++)
                                                                                                                 nter element[0][2]:
                                                                                                               Enter element[1][0]: 4
Enter element[1][1]: 5
Enter element[1][2]: 6
12
             cout<<"Enter element["<<i<<"]["<<j<<"]: ";
cin>>arr[i][j];
14
15
16
                                                                                                                Enter element[2][0]: 7
         cout<<"The Matrix Form is: "<<endl; //Printing the element of 2D array
                                                                                                               Enter element[2][1]: 8
Enter element[2][2]: 9
17 |
18 |
19 |
          for(int i=0;i<3;i++)
                                                                                                                The Matrix Form is:
           for (int j=0;j<3;j++)
20 =
               cout<<arr[i][j]<<" ";
            if(i==j)
23 = 24 25 -
                                                                                                               The sum of left diagonal is = 15
The sum of right diagonal is = 15
              sum1+=arr[i][j];
26
27 🚍
            if(i+j==2)
                                                                                                                  rocess exited after 6.948 seconds with return value 0
              sum2+=arr[i][j];
28
29
31
           cout<<endl:
          cout<<"The sum of left diagonal is = "<<sum1<<end1;
cout<<"The sum of right diagonal is = "<<sum2;</pre>
33
```

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

44 45

46 47

48

return 0;

cout<<endl;

```
1 #include <iostream>
     using namespace std;
 3
      void matrices(int matrix1[3][3],int matrix2[3][3],int addmatrice[3][3])
 4
       //function declaration
 5 □ {
 6
              for(int i=0;i<3;i++)
 7 🖨
                                                                                                                  D:\FOP\lab manuals\lb9.exe
                    for(int j=0;j<3;j++)</pre>
 8
                                                                                                                  Elements of 1st Matrix are:
                                                                                                                 Elements of 1st Matrix ar

Enter element [0][0] = 1

Enter element [0][1] = 2

Enter element [0][2] = 3

Enter element [1][0] = 4

Enter element [1][1] = 5

Enter element [1][2] = 6

Enter element [2][0] = 7

Enter element [2][1] = 8

Enter element [2][2] = 9

Enter element [2][2] = 9
 9 🖨
10
                          addmatrice[i][j]=matrix1[i][j]+matrix2[i][j];
11
12
13 \ }
14
       int main()
15 □ {
16
              int matrix1[3][3];
17
              int matrix2[3][3];
                                                                                                                  Elements of 2nd Matrix are
18
              int addmatrice[3][3];
                                                                                                                  Enter element [0][0] = 1
Enter element [0][1] = 2
              cout<<"Elements of 1st Matrix are: "<<endl;</pre>
19
                                                                                                                 Enter element [0][1] = 2

Enter element [0][2] = 3

Enter element [1][0] = 4

Enter element [1][1] = 5

Enter element [1][2] = 6

Enter element [2][0] = 7

Enter element [2][1] = 8

Enter element [2][2] = 9

Sum of matrix is = 2

2 4 6
20
              for(int i=0;i<3;i++)</pre>
21 🖨
22
                    for(int j=0;j<3;j++)</pre>
23 🖨
24
                          cout<<"Enter element ["<<i<<"]["<<j<<"] = ";</pre>
25
                          cin>>matrix1[i][j];
26
27
28
              cout<<"Elements of 2nd Matrix are: "<<endl;
29
              for(int i=0;i<3;i++)
                                                                                                                  14 16 18
30日
31
                    for(int j=0;j<3;j++)
                                                                                                                  Process exited after 25.85 seconds with return value 0
32 🗀
33
                           cout<<"Enter element ["<<i<<"]["<<j<<"] = ";</pre>
34
                           cin>>matrix2[i][j];
35
36
37
              matrices(matrix1, matrix2, addmatrice); //calling the function
38
              cout<<"Sum of matrix is = "<<endl;</pre>
              for(int i=0;i<3;i++)
39
40 🖨
                    for(int j=0;j<3;j++)</pre>
41
42 🖨
43
                           cout<<addmatrice[i][j]<<" ";</pre>
```

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

```
#include <iostream>
      using namespace std;
                                                                                   D:\FOP\lab manuals\prac2.exe
                                                                                                                                                                                void transpose(int arr[3][3]) //declaration of function
                                                                                  Enter elements of matrix
                                                                                  Enter element[0][0]: 1
           cout<<"Transpose is: "<<endl;
               for(int i=0;i<3;i++)
                                                                                  Enter element[0][1]: 2
                                                                                 Enter element[0][2]: 3
Enter element[1][0]: 4
               for(int j=0;j<3;j++)
                   cout<<arr[j][i]<<" ";
                                                                                  Enter element[1][1]: 5
11 -
12
13 -
14
                                                                                 Enter element[1][2]: 6
Enter element[2][0]: 7
               cout << endl:
                                                                                  Enter element[2][1]: 8
int main() {

int arr[3][3];

cout<<"Enter elements of matrix "<<endl;
                                                                                  Enter element[2][2]: 9
                                                                                  Matrix will be
18 |
19 |
20 |
21 |
           for(int i=0;i<3;i++)
                                                                                 1 2 3
4 5 6
               for(int j=0;j<3;j++)
                                                                                    8 9
22
                   cout<<"Enter element["<<i<<"]["<<j<<"]: ";
                                                                                  Transpose is:
23
                   cin>>arr[i][j];
                                                                                    4 7
24
                                                                                     5 8
25
                                                                                     6 9
26
27
           cout<<endl;
           cout<<"Matrix will be "<<endl;
27
28
29 = 30
31 = 32
           for(int i=0;i<3;i++)
                                                                                  Process exited after 5.162 seconds with return value 0
               for(int j=0;j<3;j++)</pre>
                                                                                  Press any key to continue \dots
                   cout<<arr[i][j]<<" ";
33
34
               cout << end1:
35
           transpose(arr); //calling the function
37
```

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

```
1
     #include <iostream>
     using namespace std;
 3 □ void multiplymatrices(int matrix1[3][3], int matrix2[3][3], int result[3][3]) {
 4 = for(int i=0;i<3;i++) {
 5 for(int j=0;j<3;j++) {
 6
    result[i][j]=0;
 7
     }
 8
     }
 9
10 ☐ for (int i=0;i<3;i++) {
11 \square for (int j=0;j<3; j++) {
12 = for(int k=0; k<3; k++) {
     result[i][j] =result[i][j]+ matrix1[i][k] * matrix2[k][j];
13
14
15
16
      }
   L }
17
18 ☐ int main() {
19
     int matrix1[3][3];
20
     int matrix2[3][3];
21
     int result[3][3];
     cout << "Enter elements of first matrix "<<endl;</pre>
22
23 for (int i=0;i<3;i++){
24 ☐ for (int j=0;j<3;j++) {
     cout<<"Enter elements["<<i<<"]["<<j<<"] : ";</pre>
25
    cin>>matrix1[i][j];
```

```
27
28
    }
29
    cout<<"Enter elements of second matrix "<<endl;</pre>
30 ☐ for (int i=0;i<3;i++) {
31 ☐ for (int j=0;j<3;j++) {
32
     cout<<"Enter elements["<<i<<"]["<<j<<"] : ";
33
     cin >> matrix2[i][j];
34
   - }
35
   - }
36
     multiplymatrices(matrix1, matrix2, result);//Calling the function
37
    // Display the result
   cout << "Multiplication of matrices: "<<endl;</pre>
39 ☐ for (int i=0;i<3;i++) {
40 \square for (int j=0;j<3;j++) {
41
     cout<<result[i][j]<< " ";
42
43
    cout<<endl;
44
    }
45
    return 0;
46
   └ }
```

5. Print the multiplication table of 15 using recursion.

```
D:\FOP\lab manuals\prc5.exe
31
     #include<iostream>
                                                           15 X 1 = 15
     using namespace std;
                                                           15 X 2 = 30
 3
     int table(int n,int x) //function declaration
                                                           15 X 3 = 45
 4 □ {
                                                           15 X 4 = 60
 5 🖨
          if(x<=10){
                                                           15 X 5 = 75
          cout<<n<<" X "<<x<<" = "<<n*x<<endl;
 6
                                                           15 X 6 = 90
                                                           15 X 7 = 105
 7
          table(n,x+1);
                                                           15 X 8 = 120
 8
                                                           15 X 9 = 135
 9
     }
                                                           15 X 10 = 150
10
     int main()
11 □ {
                                                           Process exited after 8.601 seconds with return value 0
12
          int n=15;
                                                           Press any key to continue . . .
          table(n,1); //calling the function
13
14
        return 0;
15 L }
```

#### **HOMETASK:**

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

```
1
     #include<iostream>
 2
      using namespace std:
 3
      double Determinant(int mat[3][3])
 4 □ {
 5
     return mat[0][0]*(mat[1][1]*mat[2][2]-mat[1][2]*mat[2][1])-
 6
      mat[0][1]*(mat[1][0]*mat[2][2]-mat[1][2]*mat[2][0])+mat[0][2]*(mat[1][0]*mat[2][1]-
 7
      mat[1][1]*mat[2][0]);
 8 L
 9 □ void Adjoint(int mat[3][3], int adj[3][3]) {
10 pfor (int i=0;i<3;i++) {
11 p for (int j=0;j<3;j++) {
    adj[i][j]=(mat[(j+1)%3][(i+1)%3]*mat[(j+2)%3][(i+2)%3]) -
13
    (mat[(j+1)%3][(i+2)%3]*mat[(j+2)%3][(i+1)%3]);
14
15 <del>}</del> }
17 □ void Inverse(int mat[3][3], double inv[3][3]) {
18 double det = Determinant(mat);
19 ☐ if (det==0) {
20
    cout<<"Inverse does not exist"<<endl;
21
    return;
22
23
    int adj[3][3];
24 Adjoint(mat,adj);
25 for (int i=0;i<3;i++) {
26 for (int j=0;j<3;j++) {
    inv[i][j]=adj[i][j]/det;
27
28
    - }
                                                                   D:\FOP\lab manuals\lb9hmt1.exe
29
30 E
                                                                   Enter elements of matrix:
                                                                  Enter element[0][0] : 4
Enter element[0][1] : 2
Enter element[0][2] : 6
31 ☐ int main() {
32 | int mat[3][3];
                                                                  Enter element[1][0]
33 cout << "Enter elements of matrix:"<<endl;</p>
                                                                  Enter element[1][1]: 6
Enter element[1][2]: 9
Enter element[2][0]: 5
Enter element[2][1]: 3
34 pfor (int i=0;i<3;i++) {
35 ☐ for (int j=0;j<3;j++) {
     cout<<"Enter element["<<i<<"]["<<j<<"] : ";
37
     cin>>mat[i][j];
                                                                   Enter element[2][2] : 1
38
                                                                   Inverse of the matrix:
39
                                                                   0.190909 -0.145455 0.163636
40
    double inv[3][3];
                                                                   -0.372727 0.236364 0.109091
    Inverse(mat, inv);
                                                                   0.163636 0.0181818 -0.145455
41
42 if (Determinant(mat) != 0) {
43 cout << "Inverse of the matrix:"<<endl;
44 for (int i=0;i<3;i++) {
                                                                   Process exited after 23.49 seconds with return value 0
                                                                   Press any key to continue . . .
45 ☐ for (int j=0;j<3;j++) {
46 | cout<<inv[i][j] << " ";
47
48
    cout<<endl;
49
     }
50
51
    return 0;
```