```
/*:DSA PRATICAL4:Tittle:- Write C/C++ program for storing matrix. Write functions for
a) Check whether given matrix is upper triangular or not
b) Compute summation of diagonal elements
c) Compute transpose of matrix
d) Add, subtract and multiply two matrices */

#include<iostream>
using namespace std;
class Matrix
{
private:
      int matrix[3][3], matrix1[3][3],matrixA[3][3],matrixB[3][3],matrixC[3][3],matrixD[3][3];
      int i,j,flag,sum,k;
public:
     void uppertri();   // FUNCTION FOR UPPERTRIANGULAR
     void sumdi();      //FUNCTION FOR SUM OF DIAGOANAL ELEMENT
     void transpose();   //FUNCTION FOR TRANSPOSE OF MATRIX
     void add();         // FUNCTION FOR ADDITION OF MATRIX
     void sub();         // FUNCTION FOR SUBTRACTION OF MATRIX
     void mult();        // FUNCTION FOR MULTIPLICATION OF MATRIX
};
void Matrix::uppertri()
{
     cout<<"\nEnter element of matrix:"<<endl;         // TAKING ELEMENT TO STORE IN THE
MATRIX
     for(i=0;i<3;i++)
     {
          for(j=0;j<3;j++)
          {
               cout<<"ELEMENT FOR POSITION:"<<i+1<<","<<j+1<<":";
               cin>>matrix[i][j];
          }
     }
     cout<<"MATRIX:"<<endl;
     for(i=0;i<3;i++)       //    MATRIX REPRESENTATION
     {
          for(j=0;j<3;j++)

               cout<<matrix[i][j]<<" ";
               cout<<endl;

     }
     flag=1;                    //CHECK WHETHER MATRIX IS UPPER TRIANGULAR OR NOT
     for(i=0;i<3;i++)
     {
          for(j=0;j<3;j++)
          {
```

```
                if(j<i && matrix[i][j]!=0)
                {
                    flag= 0;

                }


            }
        }
        if(flag==1)
        {

            cout<<" THIS IS UPPER TRIANGULAR MATRIX "<<endl;

        }
        else
        {
            cout<<"THIS IS NOT UPPER TRIANGULAR MATRIX"<<endl;
        }
}
void Matrix::sumdi()
{
        // SUM OF DIAGONAL ELEMENT
        sum=0;
        for(i=0;i<3;i++)
        {
            sum=sum+matrix[i][i];
        }
        cout<<"\n SUM OF DIAGONAL ELEMENT="<<sum;


}
void Matrix::transpose()
{
        // TRANSPOSE OF MATRIX
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                matrix1[i][j]= matrix[j][i];
            }
            cout<<"\n";
        }
        cout<<"TRANSPOSE OF MATRIX :\n";
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
```

```
            {
                cout<<matrix1[i][j]<<" ";
            }
            cout<<"\n";
        }
}
void Matrix::add()
{
  // ADDITION OF MATRIX
        cout<<"Enter Second MATRIX WHICH YOU WANT ADD PREVIOUS MATRIX:"<<endl;
        cout<<"\nEnter element of matrix:";          // TAKING ELEMENT TO STORE IN THE
MATRIX
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                cout<<"\nELEMENT FOR POSITION:"<<i+1<<","<<j+1<<":";
                cin>>matrixA[i][j];
            }
        }
        // REPRESENTATION OF SECOND    MATRIXA
        cout<<"MATRIX:"<<endl;
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                cout<<matrixA[i][j]<<" ";

            }
            cout<<"\n";
        }
  // ADITION OF TWO MATRIX ELEMENT STORE IN ANOTHER MATRIX

        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                matrixB[i][j]=matrix[i][j]+matrixA[i][j];
            }
        }
        // REPERSENT ADDTION OF TWO MATRIX
        cout<<"ADDITION OF TWO MATRIX:"<<endl;
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                cout<<matrixB[i][j]<<" ";
```

```
            }
            cout<<"\n";
        }

}
void Matrix::sub()
{
        // SUBTRACTION OF TWO MATRIX
         for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                matrixC[i][j]=matrix[i][j]-matrixA[i][j];
            }
        }
         // REPERSENT SUBTRACTION OF TWO MATRIX
        cout<<"SUBTRACTION OF TWO MATRIX:"<<endl;
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                cout<<matrixC[i][j]<<" ";

            }
            cout<<"\n";
        }
}
void Matrix::mult()
{
        // MULTIPLIACTION OF TWO MATRIX
         for(i=0;i<3;i++)
         {
              for(j=0;j<3;j++)
              {
                   sum=0;
                   for(k=0;k<3;k++)
                   {
                        sum += matrix[i][k]*matrixA[k][j];
                   }
                   matrixD[i][j]=sum;
              }
         }
         cout<<"MULTIPLICATION OF TWO MATRIX :\n";
         for(i=0;i<3;i++)
         {
              for(j=0;j<3;j++)
```

```
            {
                cout<<matrixD[i][j]<<" ";
            }
            cout<<"\n";
        }
}
int main()
{
    Matrix M;
    M.uppertri();
    M.sumdi();
    M.transpose();
    M.add();
    M.sub();
    M.mult();
return 0;

}
/*-----------------------------------------------------*-OUTPUT-*-----------------------------------------------------------------------
--
student@ubuntu:~$ g++ MATRIX2.cpp -o a
student@ubuntu:~$ ./a

Enter element of matrix:
ELEMENT FOR POSITION:1,1:1
ELEMENT FOR POSITION:1,2:2
ELEMENT FOR POSITION:1,3:3
ELEMENT FOR POSITION:2,1:0
ELEMENT FOR POSITION:2,2:5
ELEMENT FOR POSITION:2,3:6
ELEMENT FOR POSITION:3,1:0
ELEMENT FOR POSITION:3,2:0
ELEMENT FOR POSITION:3,3:8
MATRIX:
1 2 3
0 5 6
0 0 8
  THIS IS UPPER TRIANGULAR MATRIX

  SUM OF DIAGONAL ELEMENT=14


TRANSPOSE OF MATRIX :
1 0 0
2 5 0
3 6 8
Enter Second MATRIX WHICH YOU WANT ADD PREVIOUS MATRIX:
```

```
Enter element of matrix:
ELEMENT FOR POSITION:1,1:7

ELEMENT FOR POSITION:1,2:8

ELEMENT FOR POSITION:1,3:9

ELEMENT FOR POSITION:2,1:4

ELEMENT FOR POSITION:2,2:5

ELEMENT FOR POSITION:2,3:6

ELEMENT FOR POSITION:3,1:1

ELEMENT FOR POSITION:3,2:2

ELEMENT FOR POSITION:3,3:3
MATRIX:
7 8 9
4 5 6
1 2 3
ADDITION OF TWO MATRIX:
8 10 12
4 10 12
1 2 11
SUBTRACTION OF TWO MATRIX:
-6 -6 -6
-4 0 0
-1 -2 5
MULTIPLICATION OF TWO MATRIX :
18 24 30
26 37 48
8 16 24
------------------------------------------------------------------------------------------------------------------------
*/
```