Problem Statement :- Write C/C++ program for storing matrix. Write functions for

- Check whether given matrix is upper triangular or not
- Compute summation of diagonal elementsCompute transpose of matrix
- Add, subtract and multiply two matrices
- Determines the location of a saddle point if one exists (An m x n matrix is said to have a saddle point if some entry a[i][j] is the smallest value in row i and the largest value in j.)

```
#include<iostream>
#include <bits/stdc++.h>
const int MAX = 100;
using namespace std;
class Matrix
{
   int
a1[10][10],a2[10][10],c[10][10],sum,a[10][10],s[10][10],m[10][10],i,j,k,n,d;
   public:
   void getdata();
   void display();
   void operate();
   void trans();
   void diagonal();
   void sumall();
   bool spoint();
};
void Matrix::getdata()
{
   cout<<"enter the size of square martix(n*n) : ";
   cin>>n;
   cout<<"enter the elements of matrix 1 : \n";
   for(i=0;i<n;i++)
   {
      for(j=0;j<n;j++)
      {
         cin>>a1[i][j];
```

```
    }
  }
  cout<<"enter the elements of matrix 2 : \n";
  for(i=0;i<n;i++)
  {
    for(j=0;j<n;j++)
    {
      cin>>a2[i][j];
    }
  }
}
void Matrix::operate()
{
  for(i=0;i<n;i++)
  {
    for(j=0;j<n;j++)
    {
      a[i][j]=a1[i][j]+a2[i][j];
    }
  }
  for(i=0;i<n;i++)
  {
    for(j=0;j<n;j++)
    {
      s[i][j]=a1[i][j]-a2[i][j];
    }
  }
  for(i=0;i<n;i++)
  {
    for(j=0;j<n;j++)
    {
      m[i][j]=0;
    }
  }
  for(i=0;i<n;i++)
  {
```

```
      for(j=0;j<n;j++)
      {
         for(k=0;k<n;k++)
         {
            m[i][j]=m[i][j]+a1[i][k]*a2[k][j];
         }
      }
   }
}
void Matrix::trans()
{
   for(i=0;i<n;i++)
   {
      for(j=0;j<n;j++)
      {
         c[i][j]=a1[j][i];
      }
   }
}
void Matrix::diagonal()
{
   d=0;
   for(i=0;i<n;i++)
   {
      for(j=0;j<n;j++)
      {
         if(i==j)
         {
            d=d+a1[i][j];
         }
      }
   }
}
void Matrix::sumall()
{
   sum=0;
```

```
   for(i=0;i<n;i++)
   {
      for(j=0;j<n;j++)
      {
         sum=sum+a1[j][i];
      }
   }
}
void Matrix::display()
{
   cout<<"\nadditon of matrix 1 and 2 is\n";
   for(i=0;i<n;i++)
   {
      for(j=0;j<n;j++)
      {
         cout<<a[i][j]<<"\t";
         cout<<"\n";
      }
      cout<<"\nsubtration of matrix 1 and 2 is\n";
      for(i=0;i<n;i++)
      {
         for(j=0;j<n;j++)
         {
            cout<<s[i][j]<<"\t";
            cout<<"\n";
         }
         cout<<"\nmultiplication of matrix 1 and 2 is\n";
         for(i=0;i<n;i++)
         {
            for(j=0;j<n;j++)
            {
               cout<<m[i][j]<<"\t";
               cout<<"\n";
            }
            cout<<"\ntranspose of matrix 1\n";
            for(i=0;i<n;i++)
```

```
            {
               for(j=0;j<n;j++)
               cout<<c[i][j]<<"\t";
               cout<<"\n";
            }
            cout<<"\nsum of diagonal elements of matrix 1\n"<<d;
            cout<<"\nsum of all elements of matrix 1\n";
            cout<<sum;
         }
      }
   }
}
bool Matrix:: spoint()
{
   for (int i = 0; i < n; i++)
   {
      int min_row = a[i][0], col_ind = 0;
      for (int j = 1; j < n; j++)
      {
         if (min_row > a[i][j])
         {
            min_row = a[i][j];
            col_ind = j;
         }
      }
      int k;
      for (k = 0; k < n; k++)
      {
         if (min_row < a[k][col_ind])
         {
            break;
         }
         if (k == n)
         {
            cout <<"Value of Saddle Point : "<< min_row;
            return true;
```

```
        }
      }
    }
    return false;
}
int main()
{
    Matrix b;
    b.getdata();
    b.operate();
    b.trans();
    b.diagonal();
    b.sumall();
    b.spoint();
    b.display();
    return 0;
}
```

Output :-