1. Write a C++ program to create a class called MATRIX using a two-dimensional array of integers. Implement the following by overloading the operator == which checks the compatibility of two matrices to be added and subtracted. Perform the following by overloading + and - operators. Display the result by overloading the operator <<.

```
#include<iostream>
using namespace std;
class matrix
{
private:long m[5][5];
int row;int col;
public:void  getdata();
int operator ==(matrix);
matrix operator+(matrix);
matrix operator-(matrix);
friend ostream & operator << (ostream &,matrix &);
};
/* function to check whether the order of matrix are same or not */
int matrix::operator==(matrix cm)
{
if(row==cm.row && col==cm.col)
{
return 1;
}
return 0;
}
/* function to read data for matrix*/
void matrix::getdata()
{
cout<<"enter the number of rows\n";
cin>>row;
cout<<"enter the number of columns\n";
cin>>col;
cout<<"enter the elements of the matrix\n";
```

```
for(int i=0;i<row;i++)
{
for(int j=0;j<col;j++)
{
cin>>m[i][j];
}
}
}
/* function to add two matrix */
matrix matrix::operator+(matrix am)
{
matrix temp;
for(int i=0;i<row;i++)
{
for(int j=0;j<col;j++)
{
temp.m[i][j]=m[i][j]+am.m[i][j];
}
temp.row=row;
temp.col=col;
}
return temp;
}
/* function to subtract two matrix */
matrix matrix::operator-(matrix sm)
{
matrix temp;
for(int i=0;i<row;i++)
{
for(int j=0;j<col;j++)
{
temp.m[i][j]=m[i][j]-sm.m[i][j];
}
temp.row=row;
temp.col=col;
```

```
}
return temp;
}
/* function to display the contents of the matrix */
ostream & operator <<(ostream &fout,matrix &d)
{
for(int i=0;i<d.col;i++)
{
for(int  j=0;j<d.col;j++)
{
fout<<d.m[i][j];
cout<<" ";
}
cout<<endl;
}
return fout;
}
/* main function */
int main()
{
matrix m1,m2,m3,m4;


 m1.getdata();
 m2.getdata();
 if(m1==m2)
 {
 m3=m1+m2;
 m4=m1-m2;
 cout<<"Addition of matrices\n";
 cout<<"the result is\n";
 cout<<m3;
 cout<<"subtraction of matrices\n";
 cout<<"The result is \n";
 cout<<m4;
 }
```

```
else
{
cout<<"order of the input matrices is not identical\n";
}

}
```

OUTPUT