15 - 09 - 22

# Experiment – 1

## Aim: Write a program to print the multiplication of two given matrix.

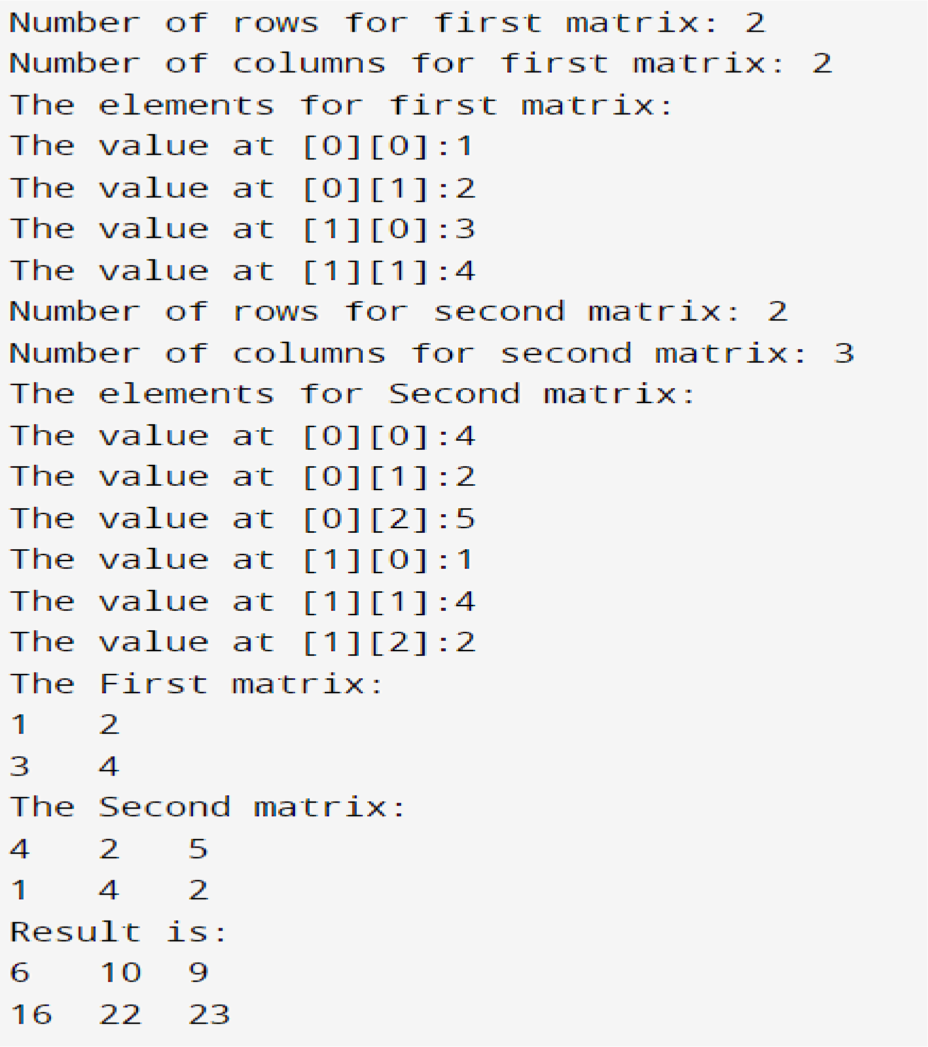
**Source Code:**

```
#include <iostream>
using namespace std;

class matrix{
  public:
  int a[10][10], b[10][10], c[10][10], r1, C1, r2, C2, i, j, k, sum;

  void input(){
    cout<<"Number of rows for first matrix: ";
    cin>>r1;
    cout<<"Number of columns for first matrix: ";
    cin>>C1;
    cout<<"The elements for first matrix: "<<endl;
    for(i=0; i<r1; i++){
      for(j=0; j<C1; j++){
        cout<<"The value at ["<<i<<"]["<<j<<"]:";
        cin>>a[i][j];
      }
    }
    cout<<"Number of rows for second matrix: ";
    cin>>r2;
    cout<<"Number of columns for second matrix: ";
    cin>>C2;
    cout<<"The elements for Second matrix: "<<endl;
    for(i=0; i<r2; i++){
        for(j=0; j<C2; j++){
            cout<<"The value at ["<<i<<"]["<<j<<"]:";
            cin>>b[i][j];
        }
    }
}

void display(){
    cout<<"The First matrix: "<<endl;
    for(i=0; i<r1; i++){
        for(j=0; j<C1; j++){
            cout<<a[i][j]<<"\t";
        }
        cout<<endl;
    }
    cout<<"The Second matrix: "<<endl;
    for(i=0; i<r2; i++){
        for(j=0; j<C2; j++){
            cout<<b[i][j]<<"\t";
        }
        cout<<endl;
    }
}

void multiply(){
    if(C1==r2) {
        for(i=0; i<r1; i++){
            for (j=0; j<C2; j++){
                sum = 0;
                for(k=0; k<r2; k++){
                    sum = sum + (a[i][k] * b[k][j]);
```

```
                        c[i][j] = sum;
                    }
                }
            }
        } else {
            cout<<"Multiplication not possible";
        }
    }

    void displayMultiplication(){
        cout<<"Result is: "<<endl;
        for(i=0; i<r1; i++){
            for(j=0; j<C2; j++){
                cout<<c[i][j]<<"\t";
            }
            cout<<endl;
        }
        cout<<endl;
    }
};

int main()
{
    matrix obj;
    obj.input();
    obj.display();
    obj.multiply();
    obj.displayMultiplication();
    return 0;
}
```

15 - 09 - 22

**Output:**



```
Number of rows for first matrix: 2
Number of columns for first matrix: 2
The elements for first matrix:
The value at [0][0]:1
The value at [0][1]:2
The value at [1][0]:3
The value at [1][1]:4
Number of rows for second matrix: 2
Number of columns for second matrix: 3
The elements for Second matrix:
The value at [0][0]:4
The value at [0][1]:2
The value at [0][2]:5
The value at [1][0]:1
The value at [1][1]:4
The value at [1][2]:2
The First matrix:
1   2
3   4
The Second matrix:
4   2   5
1   4   2
Result is:
6   10  9
16  22  23
```

22 - 09 -22

# Experiment – 2

Aim: Write a function to raise a number m to power n. The function takes a double value for m and integer value for n. Use default value for n to make the function to calculate squares

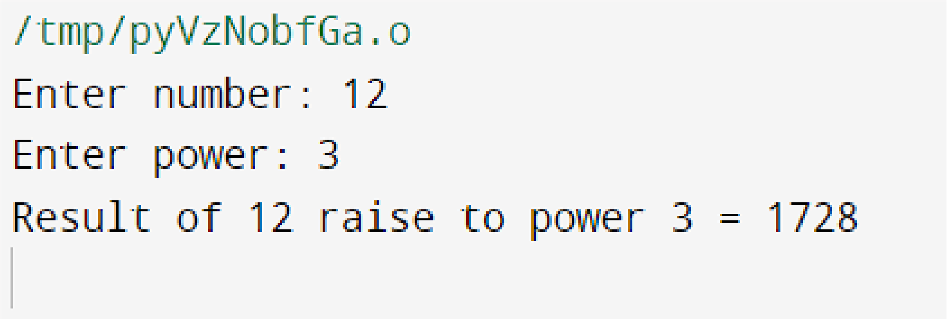
**Source Code:**

```
#include<iostream>
using namespace std;

class Power{
   public:
   double m;
   int n;
   double ans=1;

   void input(){
      cout<<"Enter number: ";
      cin>>m;
      cout<<"Enter power: ";
      cin>>n;
   }

   void Pow(){
      for(int i=0; i<n; i++){
         ans = ans * m;
      }
   }
```

22 - 09 -22

```
    void display(){
        cout<<"Result of "<<m<<" raise to power "<<n<<" = "<<ans<<endl;
    }
};

int main(){
    Power obj;
    obj.input();
    obj.Pow();
    obj.display();

    return 0;
}
```

**OUTPUT:**



```
/tmp/pyVzNobfGa.o
Enter number: 12
Enter power: 3
Result of 12 raise to power 3 = 1728
```

22 - 09 -22

}

## Experiment – 3

Aim- Calculate area of circle, rectangle and triangle by method overloading.

**Source Code:**

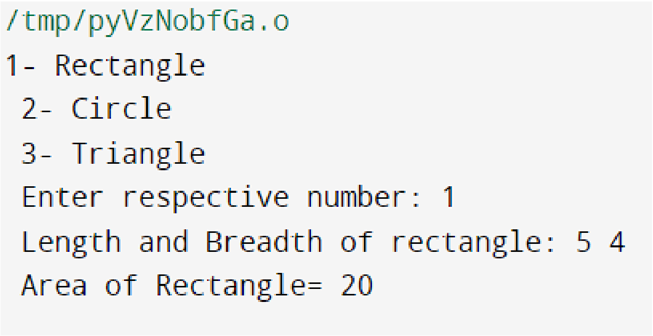
```
#include <iostream>

#include <math.h>

using namespace std;


class areaCalc{

   public:

   void area(float r){

   cout << "\n Area of Circle= " << 3.14 * r * r;

   }


   void area(int l, int b){

   cout << "\n Area of Rectangle= " << l * b ;

   }


   void area(float b, floath){

   int ar

   ar = 0.5*b*h;

   cout << "\n Area of Triangle= " << ar ;

   }

};
```

22 - 09 -22

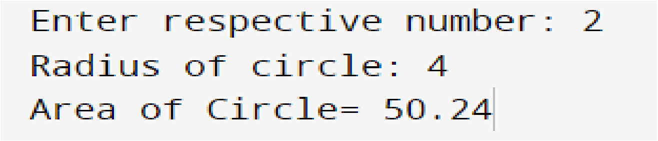
```
int main(){
    areaCalc obj;


    int r, l, b, a, c;
    cout << " 1- Rectangle\n 2- Circle\n 3- Triangle";
    cout << "\n Enter respective number: ";
    cin >> c;


    switch (c)
    {
    case 1:
        cout << "\n Length and Breadth of rectangle: ";
        cin >> l >> b;
        obj.area(l, b);
        break;
    case 2:
        cout << "\n Radius of circle: ";
        cin >> r;
        obj.area(r);
        break;
    case 3:
        cout << "\n Two sides of triangle: ";
        cin >> a >> b;
        obj.area(a, b);
        break;
    default:
        cout << "invalid choice";
    }
}
```

22 - 09 -22

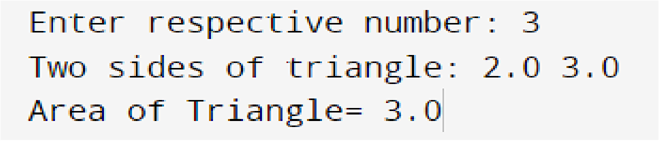
**OUTPUT:**



```
/tmp/pyVzNobfGa.o
1- Rectangle
 2- Circle
 3- Triangle
 Enter respective number: 1
 Length and Breadth of rectangle: 5 4
 Area of Rectangle= 20
```



```
 Enter respective number: 2
 Radius of circle: 4
 Area of Circle= 50.24
```



```
 Enter respective number: 3
 Two sides of triangle: 2.0 3.0
 Area of Triangle= 3.0
```

22 - 09 -22

# Experiment – 4

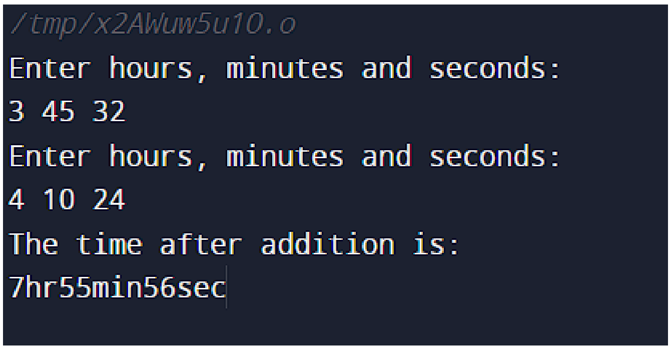
Aim: Create a class time with members hours, minutes and seconds. Take input, add two time objects and pass objects as argument to function and display result.

**Source Code:**

```
#include <iostream>
using namespace std;

class Time
{
   int hours, minutes, seconds;

public:
   void setTime();
   void addTime(Time, Time);
};

void Time :: setTime()
{
   cout << "Enter hours, minutes and seconds: " << endl;
   cin >> hours >> minutes >> seconds;
}

void Time :: addTime(Time t1, Time t2)
{
   int hr, m, s;
   s = t1.seconds + t2.seconds;
   m = t1.minutes + t2.minutes + s / 60;
   hr = t1.hours + t2.hours + m / 60;
   s %= 60;
   m %= 60;
   cout << "The time after addition is: " << endl
```

```
        << hr << ":" << m << ":" << s;
}

int main()
{
    Time t1, t2, t3;
    t1.setTime();
    t2.setTime();
    t3.addTime(t1, t2);
    return 0;

    }
```

**OUTPUT:**



```
/tmp/x2AWuw5u1O.o
Enter hours, minutes and seconds:
3 45 32
Enter hours, minutes and seconds:
4 10 24
The time after addition is:
7hr55min56sec
```

29 - 09 - 22

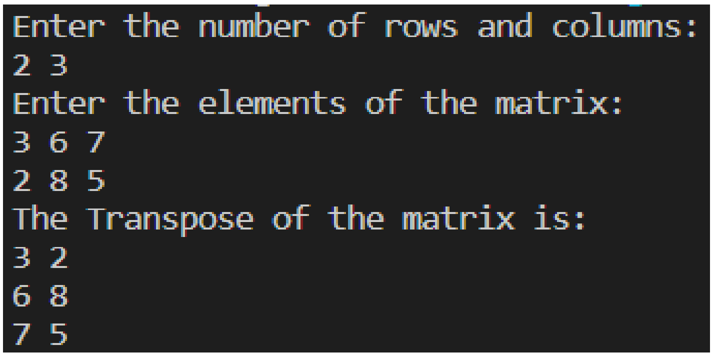
## Experiment-5

## Aim: Write a program to print the transpose of a given matrix.

**Source Code:**

```
#include<iostream>
using namespace std;

class matrix{
  int a[10][10],b[10][10];
  int row,col;
  public:
    void InputMatrix();
    void TransposeMatrix();
    void Display();
};
  void matrix :: InputMatrix(){
  cout<<"Enter the number of rows and columns: "<<endl;
  cin>>row>>col;
  cout<<"Enter the elements of the matrix: "<<endl;
  for (int i=0; i<row; i++){
    for (int j=0; j<col; j++){
      cin>>a[i][j];
    } }
}
  void matrix :: TransposeMatrix(){
  cout<<"The Transpose of the matrix is: "<<endl;
  for (int i=0; i<row; i++){
    for (int j=0; j<col; j++){
    b[j][i] = a[i][j];
    } }
    }


void matrix :: Display(){
    for (int i=0; i<col; i++){
        for (int j=0; j<row; j++){
            cout<<b[i][j]<<" ";
        }
        cout<<endl;
    }
}


int main(){
    matrix x;
    x.InputMatrix();
    x.TransposeMatrix();
    x.Display();
    return 0;
    }
```

**OUTPUT:**



```
Enter the number of rows and columns:
2 3
Enter the elements of the matrix:
3 6 7
2 8 5
The Transpose of the matrix is:
3 2
6 8
7 5
```

29 - 09 - 22

## Experiment -6

## Aim: Write a program to print the Addition of a given matrix.

**Source Code:**

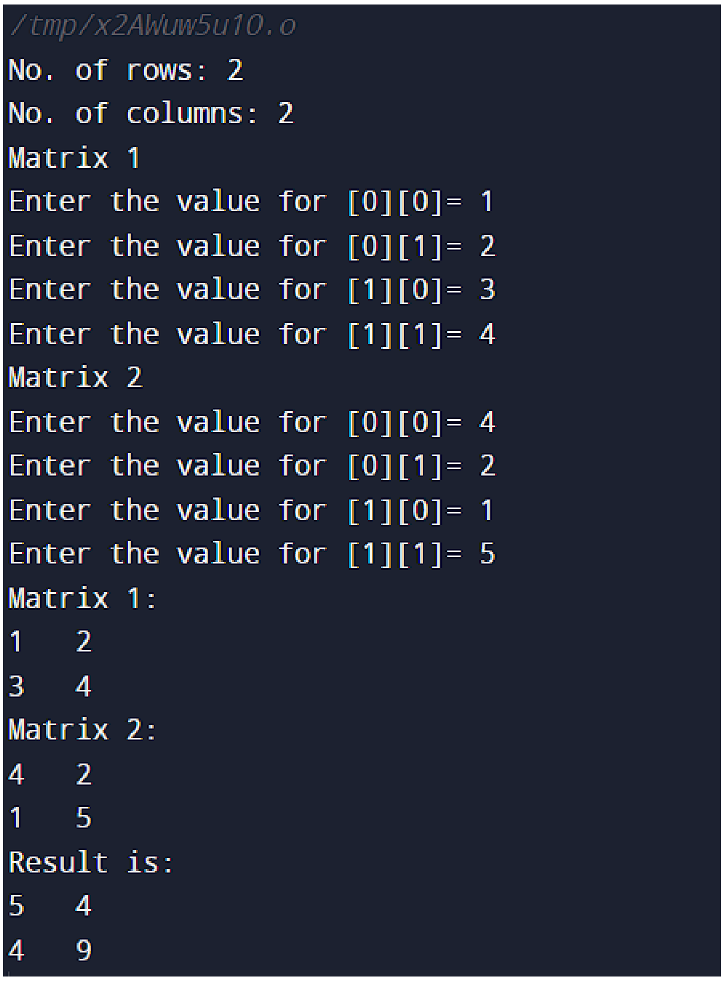
```
#include <iostream>
using namespace std;

class matrix{
  public:
  int a[10][10], b[10][10], R[10][10], r, c;

  void input(){
    cout<<"No. of rows: ";
    cin>>r;
    cout<<"No. of columns: ";
    cin>>c;

    cout<<"Matrix 1"<<endl;
    for(int i=0; i<r; i++){
      for(int j=0; j<c; j++){
        cout<<"Enter the value for ["<<i<<"]["<<j<<"]= ";
        cin>>a[i][j];
      }
    }

    cout<<"Matrix 2"<<endl;
    for(int i=0; i<r; i++){
      for(int j=0; j<c; j++){
        cout<<"Enter the value for ["<<i<<"]["<<j<<"]= ";
        cin>>b[i][j];
      }
    }
  }

  void display(){
    cout<<"Matrix 1:"<<endl;
```

```
        for(int i=0; i<r; i++){
            for(int j=0; j<c; j++){
                cout<<a[i][j]<<"\t";
            }
            cout<<endl;
        }

        cout<<"Matrix 2:"<<endl;
        for(int i=0; i<r; i++){
            for(int j=0; j<c; j++){
                cout<<b[i][j]<<"\t";
            }
            cout<<endl;
        }
    }

    void addition(){
        for(int i=0; i<r; i++){
            for(int j=0; j<c; j++){
                R[i][j] = a[i][j] + b[i][j];
            }
        }
    }

    void result(){
        cout<<"Result is:"<<endl;
        for(int i=0; i<r; i++){
            for(int j=0; j<c; j++){
                cout<<R[i][j]<<"\t";
            }
            cout<<endl;
        }
    }
};

int main()
{
    matrix obj;
```

29 - 09 - 22

```
    obj.input();
    obj.display();
    obj.addition();
    obj.result();

    return 0;
}
```

**OUTPUT:**



```
/tmp/x2AWuw5u1O.o
No. of rows: 2
No. of columns: 2
Matrix 1
Enter the value for [0][0]= 1
Enter the value for [0][1]= 2
Enter the value for [1][0]= 3
Enter the value for [1][1]= 4
Matrix 2
Enter the value for [0][0]= 4
Enter the value for [0][1]= 2
Enter the value for [1][0]= 1
Enter the value for [1][1]= 5
Matrix 1:
1   2
3   4
Matrix 2:
4   2
1   5
Result is:
5   4
4   9
```

06 - 10 - 22

# Experiment-7

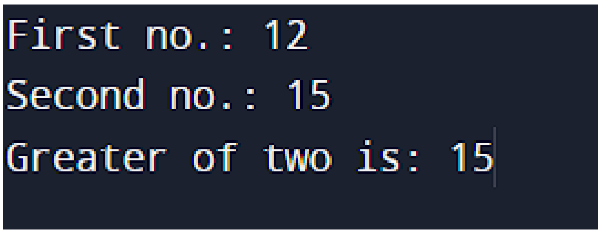
Aim - Write a program to find the greatest of two given numbers in two different classes using friend function.

**Source Code:**

```
#include <iostream>
using namespace std;

class Second;
class First{
  private:
  int x;

  public:
  void input(){
    cout<<"First no.: ";
    cin>>x;
  }
  friend void check(First, Second);
};

class Second{
  private:
  int y;

  public:
  void input(){
    cout<<"Second no.: ";
```

```
        cin>>y;
    }

    friend void check(First, Second);
};

void check(First n1, Second n2){
    if(n1.x>n2.y){
        cout<<"Greater of two is: "<<n1.x;
    } else {
        cout<<"Greater of two is: "<<n2.y;
    }
}

int main()
{
    First num1;
    Second num2;
    num1.input();
    num2.input();
    check(num1, num2);
    return 0;
}
```

**OUTPUT:**



```
First no.: 12
Second no.: 15
Greater of two is: 15
```

06 - 10 - 22

## Experiment-8

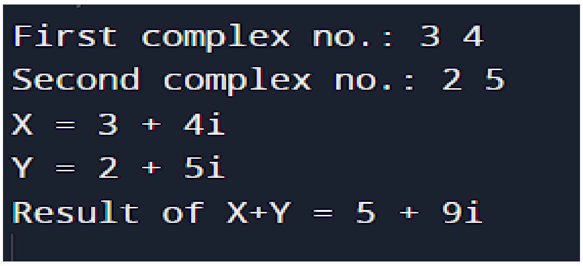
Aim : Write a program to perform addition of two complex numbers using constructor overloading. The first constructor which takes no argument is used to create objects which are not initialized, second which takes one argument is used to initialize real and imag parts to equal values and third which takes two argument is used to initialized real and imag to two different values.

**Source Code:**

```
#include <iostream>
using namespace std;

class Complex{
  public:
  int real;
  int img;
  Complex(){
     real = 0;
     img = 0;
  }

  Complex(int r, int i){
     real = r;
     img = i;
  }

  void sum(Complex C1, Complex C2){
     Complex C3;
C3.real = C1.real + C2.real;
     C3.img = C1.img + C2.img;
```

```
    cout<<"Result of X+Y = "<<C3.real<<" + "<<C3.img<<"i"<<endl;
  }
 };
int main()
{
   int real, img;

   cout<<"First complex no.: ";
   cin>>real>>img;
   Complex C1(real,img);

   cout<<"Second complex no.: ";
   cin>>real>>img;
   Complex C2(real,img);

   cout<<"X = "<<C1.real<<" + "<<C1.img<<"i"<<endl;
   cout<<"Y = "<<C2.real<<" + "<<C2.img<<"i"<<endl;

   Complex C3;

   C3.sum(C1, C2);

   return 0;
}
```

**OUTPUT:**



```
First complex no.: 3 4
Second complex no.: 2 5
X = 3 + 4i
Y = 2 + 5i
Result of X+Y = 5 + 9i
```

20 - 10 - 22

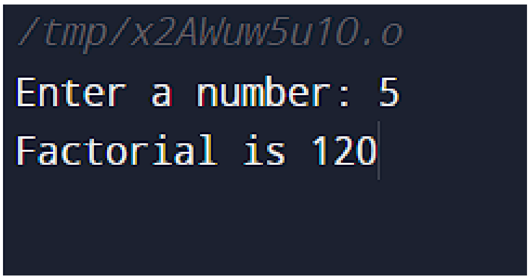
## Experiment-9

Aim - Write a program to generate a Fibonacci series using copy constructor.

**Source Code:**

```
#include <iostream>
using namespace std;

class factorial{
  public:
  int num;

 factorial();
//default constructor

 factorial(int num){
      int ans=1;
      if(num==0 || num==1){
         cout<<"Factorial is "<<ans<<endl;
      } else{
         for(int i=num; i>1; i--){
            ans = ans * i;
         }
         cout<<"Factorial is "<<ans;
      }
   }
};

int main(){
   int n;
   cout<<"Enter a number: ";
   cin>>n;
   factorial obj(n);
   return 0;
}
```

**OUTPUT:**



```
/tmp/x2AWuw5u1O.o
Enter a number: 5
Factorial is 120
```

20 - 10 - 22

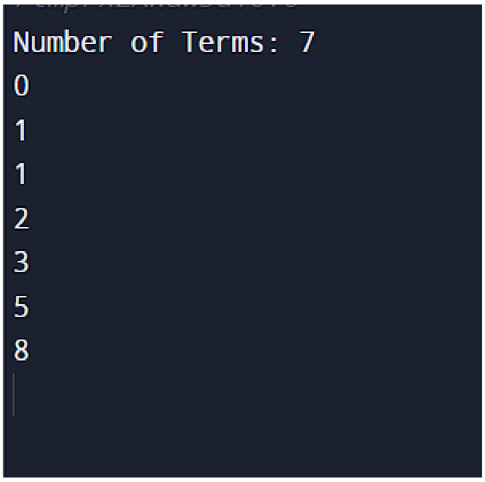
# Experiment-10

## Aim : Write a program to generate Fibonacci Series using Copy Constructor.

**Source Code:**

```
#include  <iostream>
using namespace std;

class fibonacci_series

{
   private:
   int f1,  f2, f ;
   public:
   fibonacci_series ( ){
      f1=0;
      f2=1;
      cout<<0<<endl<<1<<endl;
      f = f1 + f2;

   }

   fibonacci_series ( fibonacci_series & ptr ){
      f1 = ptr.f1;
      f2 = ptr.f2;
      f = ptr.f;
   }

   void increment ( ) {
        f1 = f2  ;
        f2 = f ;
        f =f1 + f2 ;
    }

    void show ( ){
        cout << f <<endl;
    }
};

int main (void){
    int x;
    cout<<"Number of Terms: ";
    cin>>x;
    fibonacci_series number;
    for ( int i = 0 ; i <= x-3 ; i++ ){
        number. show ( ) ;
        number. increment ( );
    }
}
```

**OUTPUT:**



```
Number of Terms: 7
0
1
1
2
3
5
8
```

03-11-22

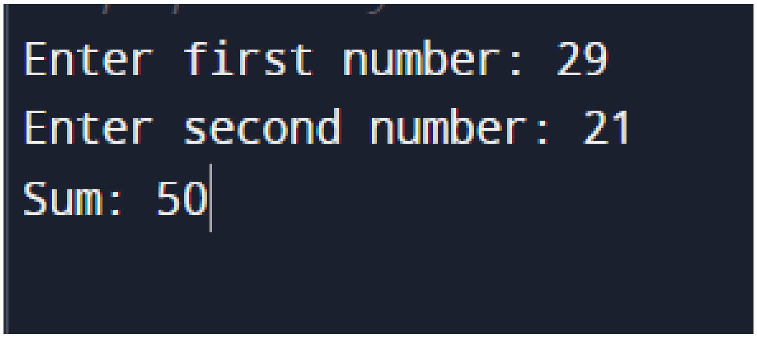
# Experiment-11

Aim: Write a program, sum of 2 numbers declared in class and the display the number and sum using friend class.

**Source Code:**

```
#include<iostream>

using namespace std;


class B;

class A{

   private:

      int x;

   public:

      int val(){

         cout<<"Enter first number";

         cin>>x;

         return 0;

      }

      friend int add(A,B);

};

class B{

   private:

      int y;

   public:
    int val(){
      cout<<"Enter second number";
      cin>>y;
      return 0;
    }
    friend int add(A,B);
};
int add(A obj1 , B obj2){
  return(obj1.x + obj2.y);
}
int main(){
  A obj1;
  B obj2;
  obj1.val();
  obj2.val();
  cout<<"Sum: "<<add(obj1 , obj2);
  return 0;
}
```

**OUTPUT:**



```
Enter first number: 29
Enter second number: 21
Sum: 50
```