Practical-1

Aim: Create a Calculator in C++ by using all types of user defined functions. User can perform all types of basic arithmetic operations until he/she wants.

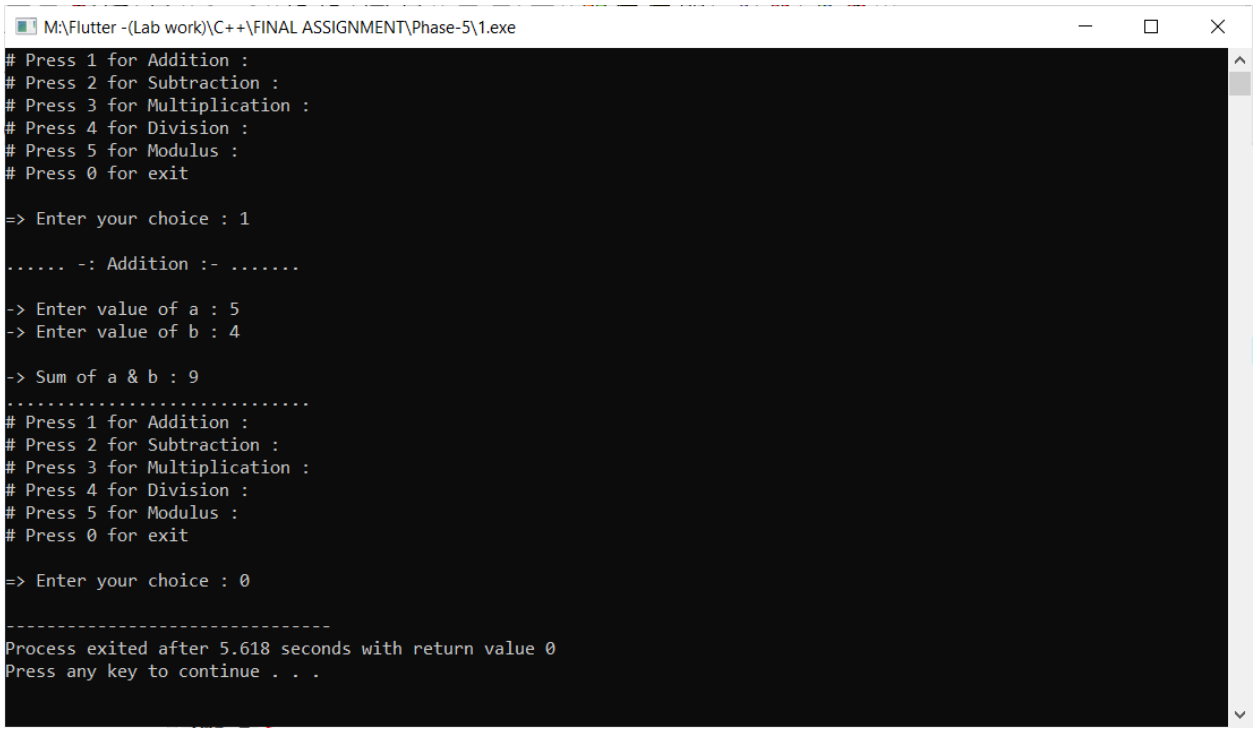
Program:

```
#include<iostream>
#include<stdio.h>
using namespace std;

void Addition (int n ,int m)
{
	cout <<endl<< "-> Sum of a & b : "<<n+m;
}
void Subtraction (int n ,int m)
{
	cout <<endl<< "-> Sub of a - b : "<<n-m;
}
void Multiplication (int n ,int m)
{
	cout <<endl<< "-> Mult of a * b : "<<n*m;
}
void Division (int n ,int m)
{
	cout <<endl<< "-> Div of a / b : "<<n/m;
}
void Modulus (int n ,int m)
{
	cout <<endl<< "-> Mod of a % b : "<< (n-(n/m)*m);
}

int main()
{

	int a,b,choice;
```

```
do
{
cout <<"# Press 1 for Addition : "<<endl;
cout <<"# Press 2 for Subtraction : "<<endl;
cout <<"# Press 3 for Multiplication : "<<endl;
cout <<"# Press 4 for Division : "<<endl;
cout <<"# Press 5 for Modulus : "<<endl;
cout <<"# Press 0 for exit"<<endl;

cout <<endl<< "=> Enter your choice : ";
cin >> choice;

switch(choice)
 {
                case 1 :
                                cout <<endl<<"..... -: Addition :- ..... "<<endl;
                                cout <<endl<< "-> Enter value of a : ";
                                cin >> a;
                                cout << "-> Enter value of b : ";
                                cin >> b;
                                Addition (a,b);
                                cout <<endl<<"..........................."<<endl;
                                break;
                case 2 :
                                cout <<endl<<"..... -: Subtraction :- ..... "<<endl;
                                cout <<endl<< "-> Enter value of a : ";
                                cin >> a;
                                cout << "-> Enter value of b : ";
                                cin >> b;
                                Subtraction (a,b);
                                cout <<endl<<"..........................."<<endl;
                                break;
                case 3 :
                                cout <<endl<<"..... -: Multiplication :- ..... "<<endl;
                                cout <<endl<< "-> Enter value of a : ";
                                cin >> a;
                                cout << "-> Enter value of b : ";
                                cin >> b;
                                Multiplication (a,b);
                                cout <<endl<<"..........................."<<endl;
```

```
                                        break;
                        case 4 :
                                        cout <<endl<<"..... -: Division :- ..... "<<endl;
                                        cout <<endl<< "-> Enter value of a : ";
                                        cin >> a;
                                        cout << "-> Enter value of b : ";
                                        cin >> b;
                                Division (a,b);
                                cout <<endl<<"..........................."<<endl;
                                        break;
                        case 5 :
                                        cout <<endl<<"..... -: Modulus :- ..... "<<endl;
                                        cout <<endl<< "-> Enter value of a : ";
                                        cin >> a;
                                        cout << "-> Enter value of b : ";
                                        cin >> b;
                                Modulus (a,b);
                                 cout <<endl<<"..........................."<<endl;
                                        break;
                        case 0 :
                                        break;

                        default :
                                cout <<endl<< "Invalid Choice.....";
                }
        }while(choice!=0);
        return 0;
}
```

Output:


M:\Flutter -(Lab work)\C++\FINAL ASSIGNMENT\Phase-5\1.exe
Press 1 for Addition :
Press 2 for Subtraction :
Press 3 for Multiplication :
Press 4 for Division :
Press 5 for Modulus :
Press 0 for exit
=> Enter your choice : 1
..... -: Addition :- .....
-> Enter value of a : 5
-> Enter value of b : 4
-> Sum of a & b : 9
.....
Press 1 for Addition :
Press 2 for Subtraction :
Press 3 for Multiplication :
Press 4 for Division :
Press 5 for Modulus :
Press 0 for exit
=> Enter your choice : 0

Process exited after 5.618 seconds with return value 0
Press any key to continue . . .

Practical-2

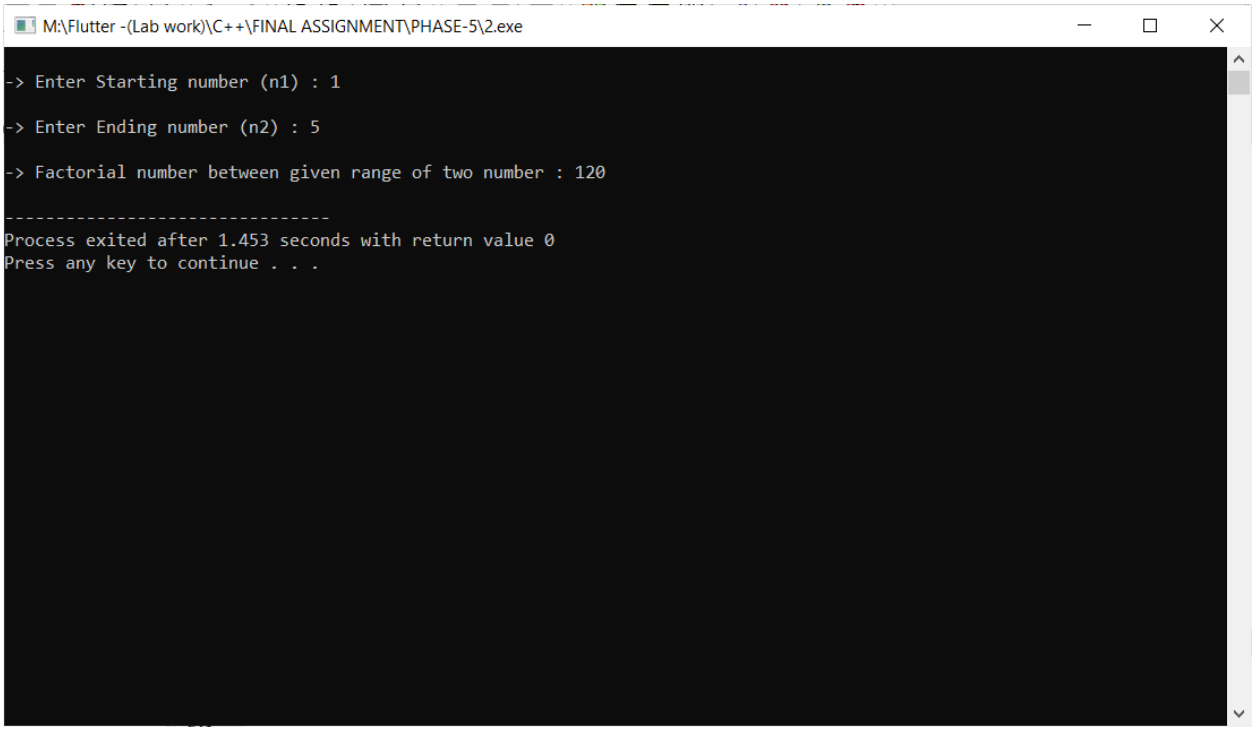
Aim: Develop a solution for Akshay by which he can retrieve factorial of all numbers between given range of two numbers using a C++ user defined function (UDF).

Program:

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

void factorial()
{

        int n1,n2,i,j,fact=1;

        cout <<endl<< "-> Enter Starting number (n1) : ";
        cin  >> n1;

        cout <<endl<< "-> Enter Ending number (n2) : ";
        cin  >> n2;

        cout <<endl<< "-> Factorial number between given range of two number : ";
        {
                for(i=n1; i<=n2 ;i++)
                {
                        fact = fact * i;
                }
                cout << fact <<endl;
         }

    }

   class A_Solution
   {
          public :
                void A_Data()
                  {
```

```
            factorial();
        }
};

int main()
{
    A_Solution a1;
    a1.A_Data();
    return 0;
}
```

Output:



Practical-3

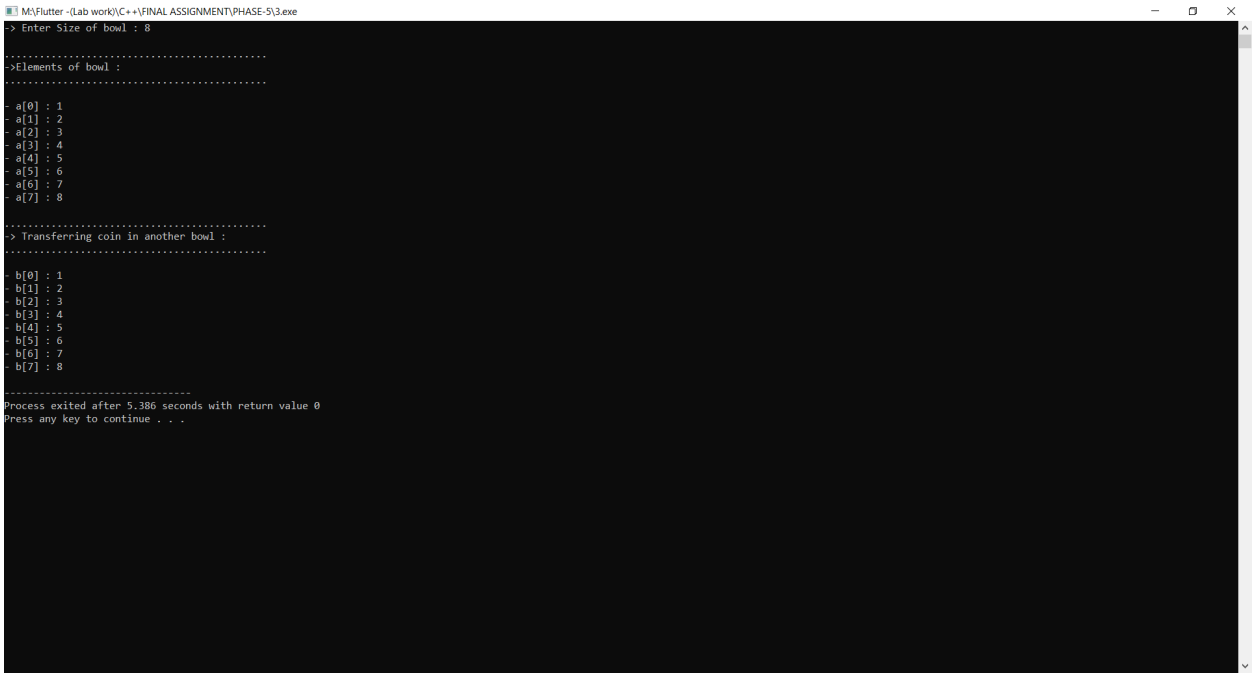
Aim: Kevin has two plain floors within different bowls containig one coin in each bowl. He bet his friend to transfer that coins in either bowls within 5 minutes.Help him by providing a C++ solution using UDF.

Program:

```
#include<iostream>
#include<string.h>
using namespace std;

void Bowl()
{
	int i,n;

	cout <<endl<<"-> Enter Size of bowl : ";
	cin  >> n;
	cout <<endl<<".....";
	cout <<endl<<"->Elements of bowl : "<<endl;
	cout <<"....."<<endl<<endl;
	int a[n];
	for(i=0;i<n;i++)
	{
		cout << "- a["<<i<<"] : ";
		cin  >> a[i];
	}
	cout <<endl<<".....";
	cout <<endl<<"-> Transferring coin in another bowl :  "<<endl;
	cout <<"....."<<endl<<endl;
	int b[n];
	for(i=0;i<n;i++)
	{
		b[i]=a[i];
		cout << "- b["<<i<<"] : "<<b[i] <<endl;
	}
}
```

```
class Kevin
{
	public :

		void K_Data()
		{
			Bowl();
		}
};

int main()
{
	Kevin k1;

	k1.K_Data();
	return 0;
}
```

Output:



Practical-4

Aim: Design a C++ UDF which producing cubes of all elements of provided array in form of another array. Then, find average value of that new array. Based on that average value decide that array's kinds :
If 22<=average<=35, then an array is "TIGHTER".
If 35<average<=50, then an array is "BALANCED".
If average>50, then an array is "TOXIC".
If average<22, then an array is "LOOSER".

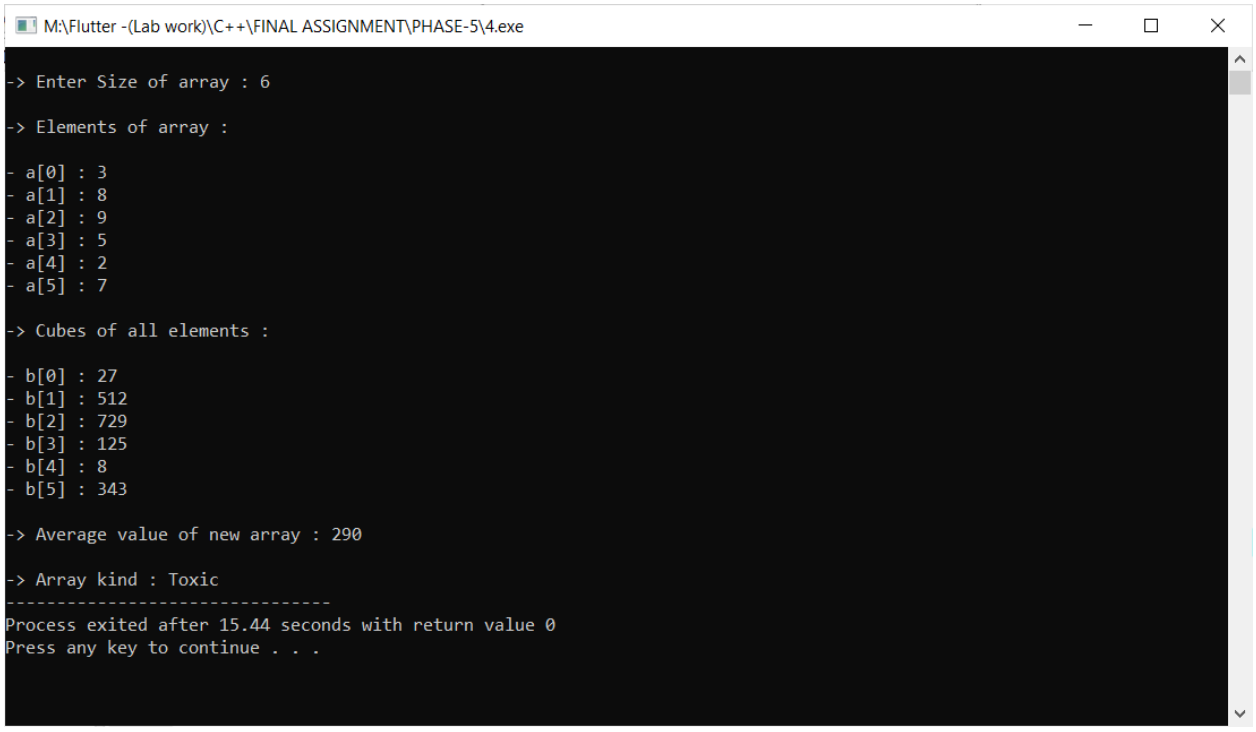
Program:

```
#include<iostream>
#include<string.h>
using namespace std;

void Array()
{
	int i,n,cube,avg=0,sum,k;

	cout <<endl<< "-> Enter Size of array : ";
	cin  >> n;
	cout <<endl<< "-> Elements of array : "<<endl<<endl;

	int a[n],b[n];
	for(i=0;i<n;i++)
	{
		cout << "- a["<<i<<"] : ";
		cin  >> a[i];
	}

	cout <<endl<<"-> Cubes of all elements : "<<endl<<endl;
	for(i=0;i<n;i++)
	{
		cube = a[i]*a[i]*a[i];
		avg  =  avg+cube;
```

```
            b[i] = cube;
        }
        k = avg/n;

        for(i=0;i<n;i++)
        {
            cout <<"- b["<<i<<"] : ";
            cout << b[i] <<endl;
        }
        cout <<endl<<"-> Average value of new array : "<<k <<endl<<endl;

        cout <<"-> Array kind : ";

        if(k>22 && k<=35)
        {
            cout << "Tighter";
        }
        else if(k>35 && k<=50)
        {
            cout << "Balanced";
        }
        else if(k>50)
        {
            cout <<"Toxic";
        }
        else
        {
            cout <<"Looser";
        }
}

class Sample
{
        public :
            void S_Data()
            {
                Array();
            }
};
```

```
int main()
{

	Sample s1;
	s1.S_Data();
	return 0;
}
```

Output:



Practical-5

Aim: A scientist wants to create a sientific calculator which only contains functionalities like:
maximum number from 3 number
square of a given number
square root of a given number
components of a given number
Design a C++ system to help this scientist by using UDFs.

Program:

```
#include<iostream>
#include<string.h>
#include<math.h>
using namespace std;

void S_Calculator()
{
        int i,n,sqr;

        cout <<endl<<"-> Enter any value : ";
        cin  >> n;

        cout <<endl<<"---------------------------------------"<<endl;
        cout <<"=> Find Maximum number from 3 number : ";
        cout <<endl<<"---------------------------------------"<<endl;
        cout <<endl<<"-> Elements of Array : "<<endl<<endl;
        int a[n];
        for(i=0;i<n;i++)
        {
                cout << "- a["<<i<<"] : ";
                cin  >> a[i];
        }
```

```
	for(i=0;i<n;i++)
	{
		if(a[0]<a[i])
		{
			a[0]=a[i];
		}
	}
	cout <<endl<< "-> Maximum number : "<<a[0];

	cout <<endl<<endl<<"----------------------------------------"<<endl;
	cout <<"=> Find square of a given number  : ";
	cout <<endl<<"----------------------------------------"<<endl;
	cout <<endl<<"-> Elements of Array : "<<endl<<endl;
	for(i=0;i<n;i++)
	{
		cout << "- a["<<i<<"] : ";
		cin  >> a[i];
	}
	cout <<endl<<"-> Square of a given number : "<<endl;
	for(i=0;i<n;i++)
	{
		sqr = a[i]*a[i];
		cout <<endl<<"- "<<sqr;
	}

	cout <<endl<<endl<<"----------------------------------------"<<endl;
	cout <<"=> Find square root of a given number : ";
	cout <<endl<<"----------------------------------------"<<endl;
	cout <<endl<<"-> Elements of Array : "<<endl<<endl;
	for(i=0;i<n;i++)
	{
		cout << "- a["<<i<<"] : ";
		cin  >> a[i];
	}
	cout <<endl<<"-> Square root of given number : "<<endl;
	int b[n];
	for(i=0;i<n;i++)
	{
		b[i]=sqrt((a[i]));
		cout <<endl<<"- "<<b[i];
	}
}

class Calculator
```

```
{
        public :

                void C_Data()
                {
                        S_Calculator();
                }
};

int main()
{
        Calculator c1;
        c1.C_Data();
        return 0;
}
```

Output:

Practical-6

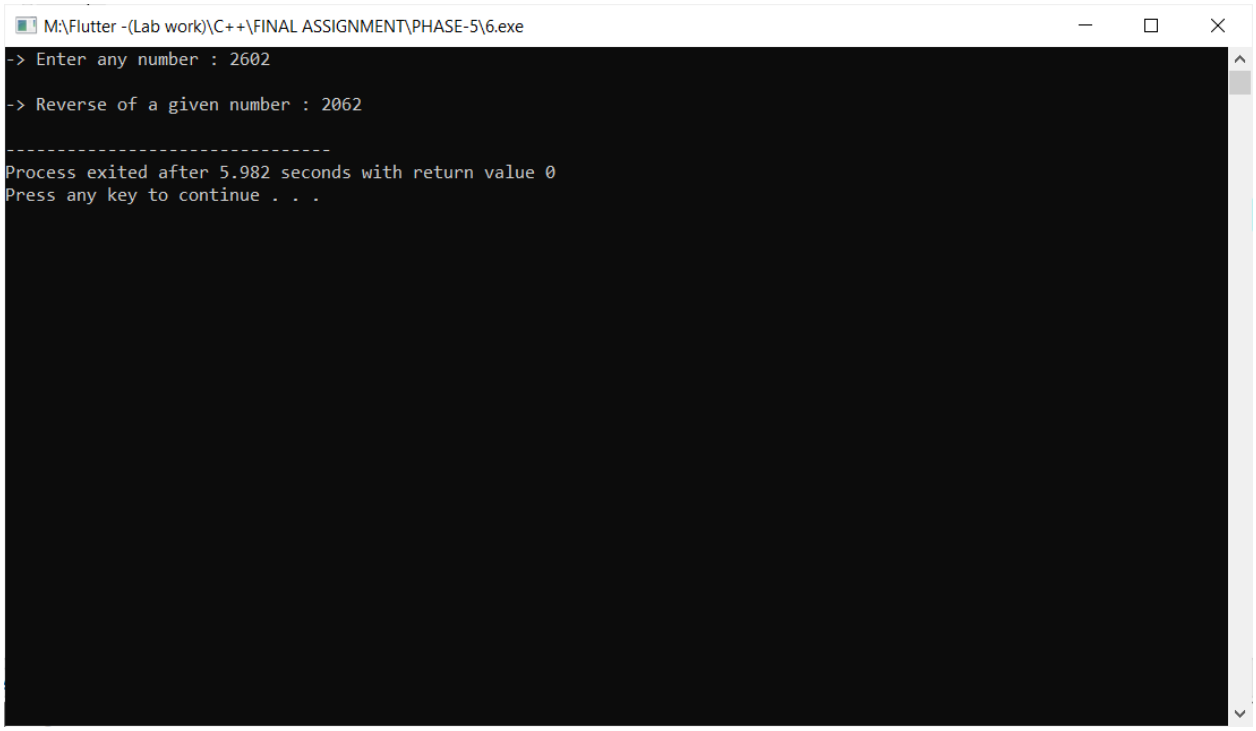
Aim: A Reality show on TV organizes "Fastest-fingers Fast"round for entering in a Game. In this round participant has to find reverse of a given number as soon as possible to win this round. Design a C++ UDF for that.

Program:

```
#include<iostream>
#include<string.h>
using namespace std;

void Reality_show()
{
        int n,rem,rev=0;

        cout << "-> Enter any number : ";
        cin  >>n;

        while (n!=0)
        {
                rem = n%10;
                rev = rev*10+rem;
                n   = n/10;
        }
        cout <<endl<<"-> Reverse of a given number : "<<rev<<endl;
}

class Round_win
{
        public :
                void G_Data()
                {
                        Reality_show();
                }
};
```

```
int main()
{
	Round_win r1;
	r1.G_Data();
	return 0;
}
```

Output:



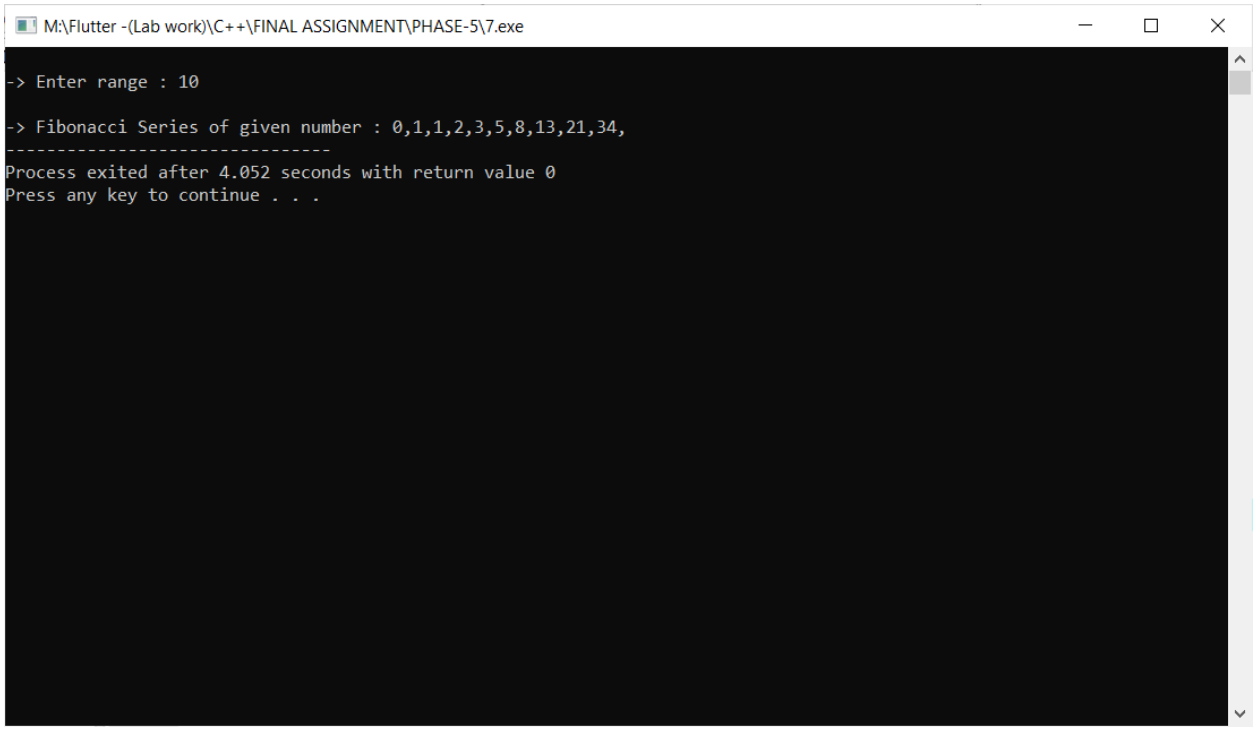
Practical-7

Aim: Ajay has to find Fibonacci Series upto given number to successfully pass in Math's examination. Help him by designing a UDF in C++.

Program:

```
#include<iostream>
#include<string.h>
using namespace std;

void Fibonacci_series(int n)
{
        int i,t1=0,t2=1,s;
        for(i=0 ;i<n ;i++)
        {

                cout  << s  <<",";
                t1=t2;
                t2=s;
                s= t1+t2;
        }
}

int main()
{
        int i,n;

        cout << "-> Enter range : ";
        cin  >> n;

        cout <<endl << "-> Fibonacci Series of given number : ";

        Fibonacci_series(n);
        return 0;
}
```

Output:


M:\Flutter -(Lab work)\C++\FINAL ASSIGNMENT\PHASE-5\7.exe
-> Enter range : 10
-> Fibonacci Series of given number : 0,1,1,2,3,5,8,13,21,34,
Process exited after 4.052 seconds with return value 0
Press any key to continue . . .

Practical-8

Aim: Design a C++ UDF which converts given seconds into time in format of HH:MM:SS. Also create another UDF which converts given time into total seconds. End user have choice to perform either operations whenever he/she wants.

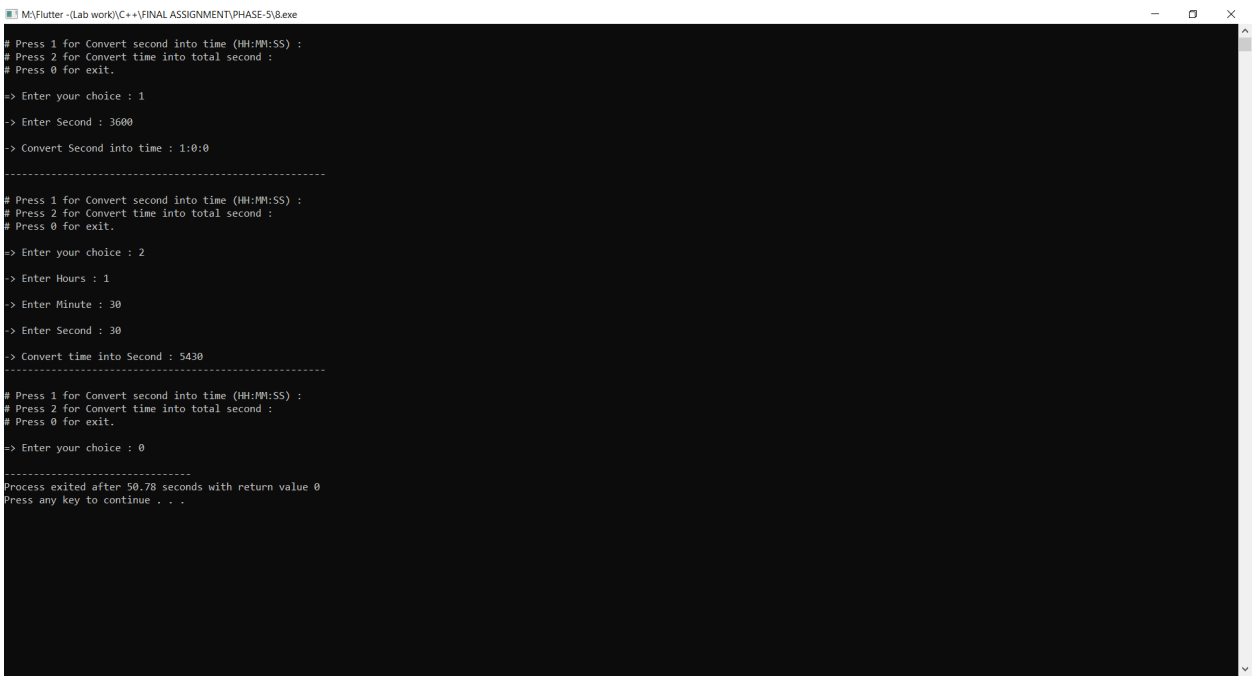
Program:

```
#include<iostream>
#include<string.h>
using namespace std;

void P_Choice()
{

        int hr,min,second,sec,total_sec;
        int choice,a,b;

        while(choice!=0)
        {

        cout <<endl<< "# Press 1 for Convert second into time (HH:MM:SS) : "<<endl;
        cout << "# Press 2 for Convert time into total second : "<<endl;
        cout << "# Press 0 for exit."<<endl;

        cout <<endl<<"=> Enter your choice : ";
        cin  >> choice;

        switch(choice)
        {
                case 1 :
                        cout <<endl<<"-> Enter Second : ";
                        cin  >> second;

                         hr= (second/60)/60;
```

```
                    min=(second/60)%60;
                    sec=second%60;

                    cout <<endl<< "-> Convert Second into time : "<<hr <<":" <<min <<":"
                    <<sec <<endl;
                    cout<<endl<<"-----------------------------------------------------"
                        <<endl;
                     break;

                case 2 :
                        cout <<endl<<"-> Enter Hours : ";
                        cin  >> hr;
                        cout <<endl<<"-> Enter Minute : ";
                        cin  >> min;
                        cout <<endl<<"-> Enter Second : ";
                         cin  >> sec;

                         total_sec = (hr*60*60 )+(min *60 )+ sec;
                         cout <<endl<< "-> Convert time into Second : "<<total_sec ;
                         cout<<endl<<"-----------------------------------------------------"
                             <<endl;
                         break;

                 case 0 :
                           break;

                 default :
                          cout <<endl<<"Invalid Input ......";
        }
        }
}

class Choice
{
        public :
                void C_Data()
                {
                        P_Choice();
                }
};
```

```
int main()
{
	Choice c1;
	c1.C_Data();
	return 0;
}
```

Output:



Practical-9

Aim: A Supreme Court wants a system which automatically figure out difference of two given time whether it is in seconds or any other format. Develop a solution in C++ using UDF.

Program:

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

void List()
{
                        cout<<"[1] Seconds     "<<endl;
                        cout<<"[2] Hours       "<<endl;
                        cout<<"[3] Minutes   "<<endl;
                        cout<<"[4] Time        "<<endl;
                        cout<<"[0] Exit        "<<endl<<endl;
}

void Seconds()
{
        int s1, s2;

        cout<<endl<<"=> Enter First Seconds: ";
        cin>>s1;
        cout<<endl<<"=> Enter Last Seconds: ";
        cin>>s2;

        if(s1<s2)
        {
                cout<<endl<<"-----------------------------------------------------"<<endl;
                cout<<"- Difference of two given time is: "<<s2-s1<<endl;
                cout<<"-----------------------------------------------------"<<endl;
        }
        else if(s1>s2)
```

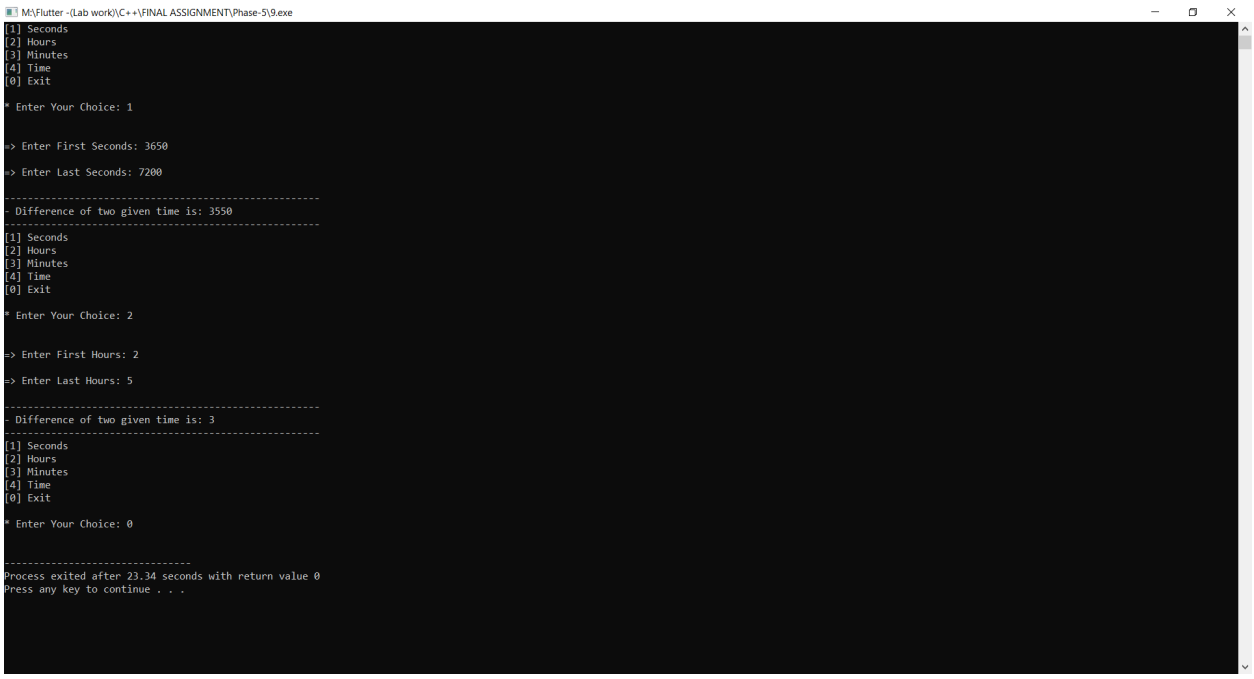
```
	{
		cout<<endl<<"-----------------------------------------------------"<<endl;
		cout<<"- Difference of two given time is: "<<s1-s2<<endl;
		cout<<"-----------------------------------------------------"<<endl;
	}

}

void Hours()
{
	int h1, h2;

	cout<<endl<<"=> Enter First Hours: ";
	cin>>h1;
	cout<<endl<<"=> Enter Last Hours: ";
	cin>>h2;

	if(h1<h2)
	{
		cout<<endl<<"-----------------------------------------------------"<<endl;
		cout<<"- Difference of two given time is: "<<h2-h1<<endl;
		cout<<"-----------------------------------------------------"<<endl;
	}
	else if(h1>h2)
	{
		cout<<endl<<"-----------------------------------------------------"<<endl;
		cout<<"- Difference of two given time is: "<<h1-h2<<endl;
		cout<<"-----------------------------------------------------"<<endl;
	}

}

void Minutes()
{
	int m1, m2;

	cout<<endl<<" Enter First Minutes: ";
	cin>>m1;
	cout<<endl<<" Enter Last Minutes: ";
	cin>>m2;
```

```
	if(m1<m2)
	{
		cout<<endl<<"-----------------------------------------------------"<<endl;
		cout<<" Difference of two given time is: "<<m2-m1<<endl;
		cout<<"-----------------------------------------------------"<<endl;
	}
	else if(m1>m2)
	{
		cout<<endl<<"-----------------------------------------------------"<<endl;
		cout<<" Difference of two given time is: "<<m1-m2<<endl;
		cout<<"-----------------------------------------------------"<<endl;
	}

}

void Time()
{
	int h1,m1,s1,h2,m2,s2,a,b,c,e,h3,m3,s3;

	cout<<endl<<"- Enter Hours: ";
	cin>>h1;
	cout<<"- Enter Minutes: ";
	cin>>m1;
	cout<<"- Enter Seconds: ";
	cin>>s1;

	cout<<endl<<"- Enter Hours: ";
	cin>>h2;
	cout<<"- Enter Minutes: ";
	cin>>m2;
	cout<<"- Enter Seconds: ";
	cin>>s2;

	if(h1>h2)
	{
		a=s1-s2;
		s3=a%60;
		b=a/60;
		c=m1-m2-b;
```

```
                m3=c%60;
                e=c/60;
                h3=h1-h2-e;

                cout<<"Substraction of above time is "
                <<h3<<" hours "
                <<m3<<" minutes "
                <<s3<<" seconds "<<endl;

        }
        else if(h1<h2)
        {
                cout<<"  Please, Enter First Time Greater Than Second Time..."<<endl;
        }
}

class time
{
        public:
                int c;

                void t()
                {
                        do{
                                List();

                                cout<<"* Enter Your Choice: ";
                                cin>>c;
                                cout<<endl;

                                if(c==1)
                                {
                                        Seconds();
                                }
                                else if(c==2)
                                {
                                        Hours();
                                }
                                else if(c==3)
                                {
```

```
                    Minutes();
                }
                else if(c==4)
                {
                    Time();
                }
                else if(c!=0)
                {
                    cout<<"          Enter Valid Value..."<<endl;
                }

            }while(c!=0);
        }
};

int main()
{
    time t1;

    t1.t();

    return 0;
}
```

Output:


M:\Flutter -(Lab work)\C++\FINAL ASSIGNMENT\Phase-5\9.exe
[1] Seconds
[2] Hours
[3] Minutes
[4] Time
[0] Exit
* Enter Your Choice: 1
=> Enter First Seconds: 3650
=> Enter Last Seconds: 7200
- Difference of two given time is: 3550
[1] Seconds
[2] Hours
[3] Minutes
[4] Time
[0] Exit
* Enter Your Choice: 2
=> Enter First Hours: 2
=> Enter Last Hours: 5
- Difference of two given time is: 3
[1] Seconds
[2] Hours
[3] Minutes
[4] Time
[0] Exit
* Enter Your Choice: 0
Process exited after 23.34 seconds with return value 0
Press any key to continue . . .

Practical-10

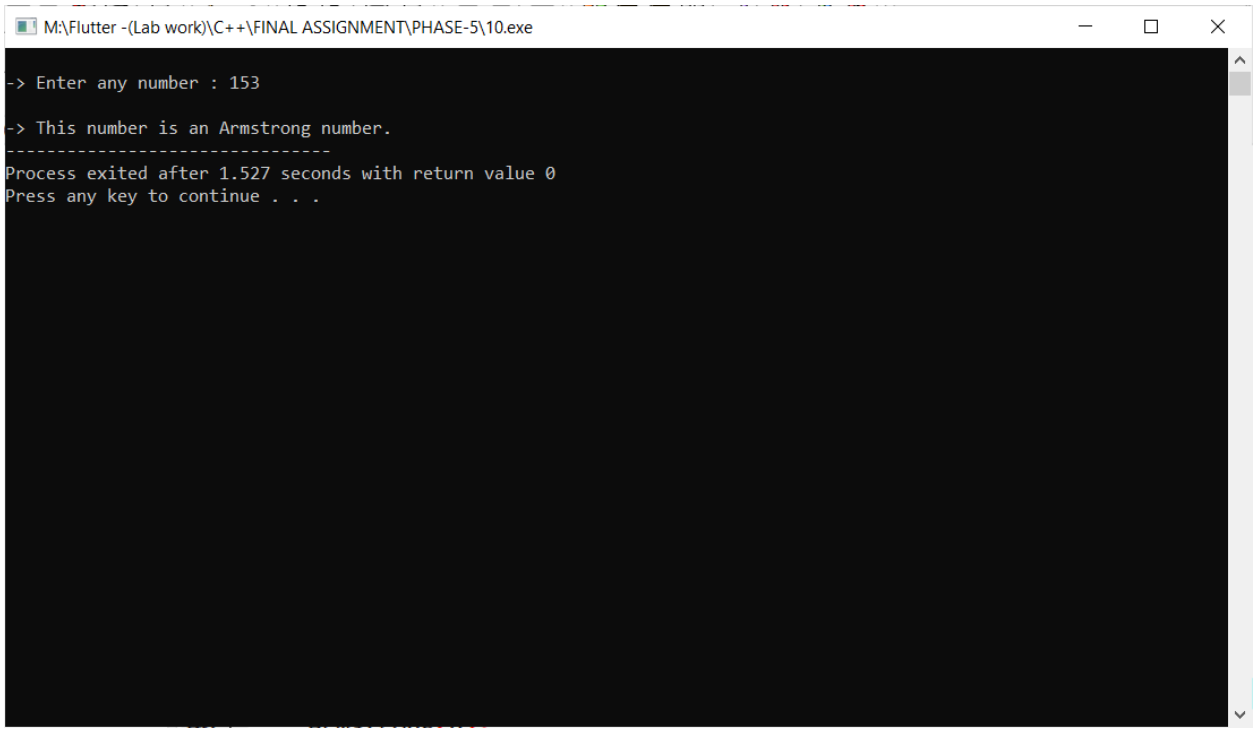
Aim: A bomb is planted at Suratgarh Railway Station. It can be defused by entering any number which is itself an Armstrong number. Design a C++ UDF which figures out if a given number is Armstrong or not.

Program:

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

void Armstrong(int n)
{
	int temp,r,y;
	temp = n;

	while(n!=0)
	{
		r = n%10;
		n = n/10;
		y = y + (r*r*r);
	}
	if(y==temp)
	{
		cout <<endl<< "-> This number is an Armstrong number.";
	}
	else
	{
		cout <<endl<< "-> This number is not an Armstrong number.";
	}

}

int main()
{
	int n,r,y=0,temp;
```

```
        cout <<endl<< "-> Enter any number : ";
        cin  >> n;
        Armstrong(n);
        return 0;
}
```

Output:



Practical-11

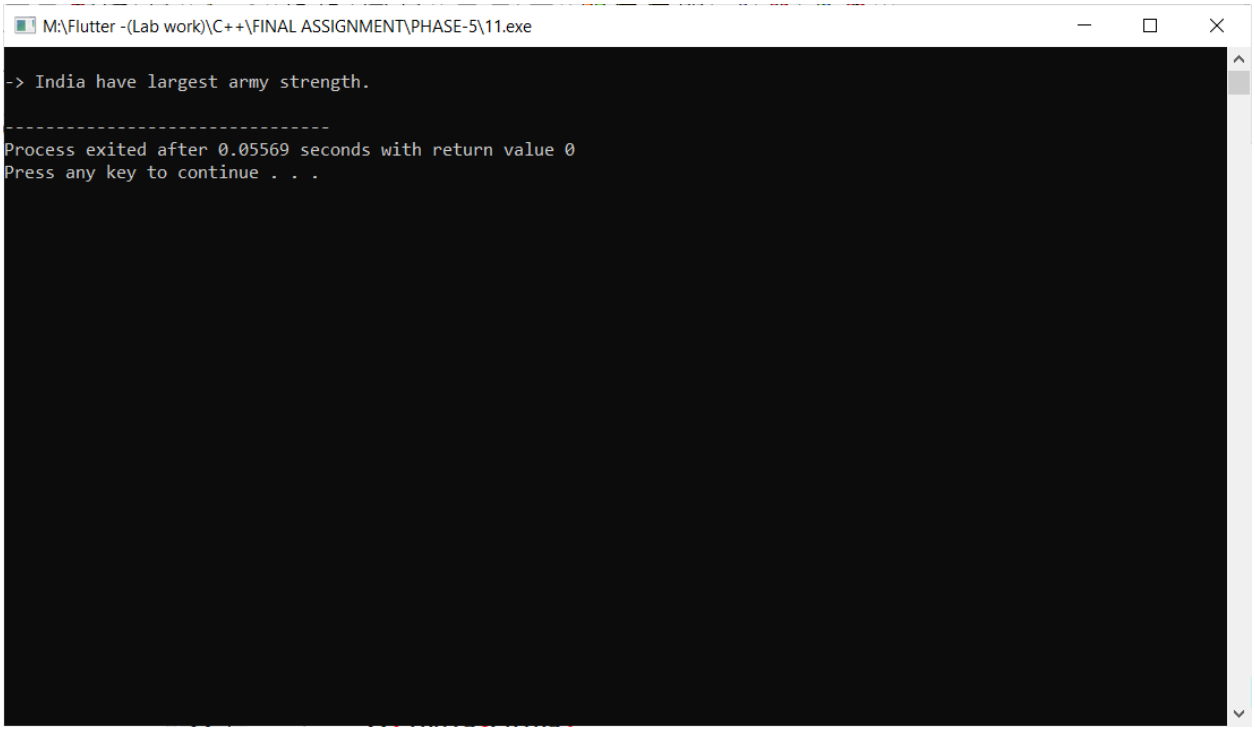
Aim: Declare a result of the survey that tells us which country have largest Army strength, US, China or India.Design a C++ UDF to announce the result of this survey to the public.

Program:

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

void Strength()
{
	int India , US , China ;

	India = 1399989;
	US    = 480893;
	China = 218500;

	if(India>US)
	{
		if(India>China)
		{
			cout <<endl<<"-> India have largest army strength."<<endl;
		}
		else
		{
			cout <<endl<<"-> China have largest army strength."<<endl;
		}
	}
	else
	{
		if(US>China)
		{
			cout <<endl<<"-> US have largest army strength."<<endl;
		}
		else
		{
```

```
                        cout <<endl<<"-> China have largest army strength."<<endl;
                }
        }
}

class LA_strength                                   // Largest Army Strength
{
        public :

                void LA_Data()
                        {
                                Strength();
                        }
};

int main()
{
        LA_strength l1;
        l1.LA_Data();
        return 0;
}
```

Output:


M:\Flutter -(Lab work)\C++\FINAL ASSIGNMENT\PHASE-5\11.exe
-> India have largest army strength.

Process exited after 0.05569 seconds with return value 0
Press any key to continue . . .

Practical-12

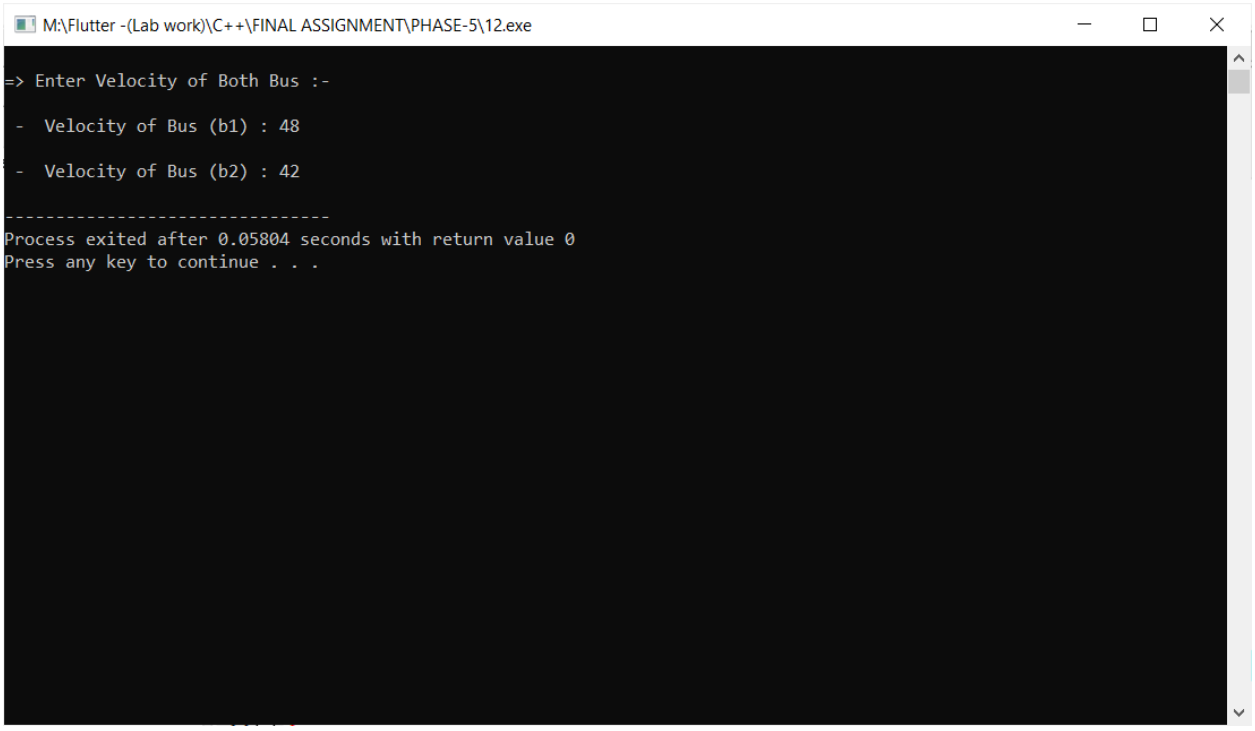
Aim: Two buses(Bus B1 & Bus B2) head forwards from Mumbai to Kolkata. Both of them have to cover total distance of 1933 KM. Bus B1 reached on destination with total time of 40 Hr & Bus B2 takes total time of 46 Hr. Find out velocity of both buses using a C-++ UDF.

Program:

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

void Bus()
{
        int t_dis = 1933 , t_b1 = 40 ,  t_b2 =46 ;                    // t_dis => total distance
        int  V1 , V2;

        V1 = t_dis / t_b1 ;
        V2 = t_dis / t_b2 ;

        cout <<endl <<"=> Enter Velocity of Both Bus :- "<<endl;
        cout <<endl<< " -  Velocity of Bus (b1) : "<<V1 <<endl;
        cout <<endl<< " -  Velocity of Bus (b2) : "<<V2<<endl;
}

class V_Bus
{
        public :

                void B_Data()
                {
                   Bus();
                }
};

int main()
{
        V_Bus v1;
        v1.B_Data();
```

```
        return 0;
}
```

Output:



Practical-13

Aim: Develop a C++ solution for Maths students to solve all types Geometry problems such like :

. Area of Circle
. Perimeter of Circle
. Area of Square
. Area of Rectangle
. Area of Triangle
. Area of Sphere

Program:

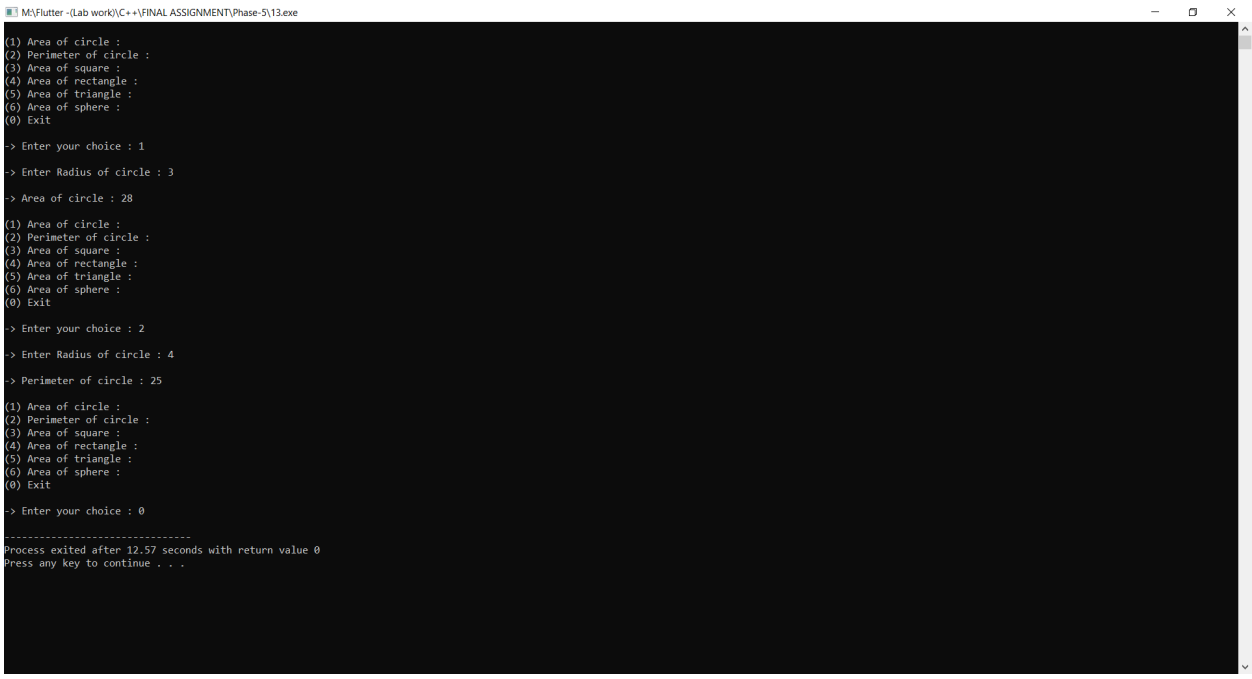
```
#include<iostream>
#include<string.h>
using namespace std;

void List()
{
        cout <<endl<< "(1) Area of circle : ";
        cout <<endl<< "(2) Perimeter of circle : ";
        cout <<endl<< "(3) Area of square : ";
        cout <<endl<< "(4) Area of rectangle : ";
        cout <<endl<< "(5) Area of triangle : ";
        cout <<endl<< "(6) Area of sphere : ";
        cout <<endl<< "(0) Exit ";
}

void Area_of_circle()
{
        int r,pi=3.14,area;

        cout <<endl<< "-> Enter Radius of circle : ";
        cin  >>r;

        area = 3.14*r*r;
```

```
	cout <<endl<<"-> Area of circle : "<<area <<endl;
}

void Perimeter_of_circle()
{
	int r,pi=3.14,peri;

	cout <<endl<< "-> Enter Radius of circle : ";
	cin  >>r;

	peri = 2*3.14*r;
	cout <<endl<<"-> Perimeter of circle : "<<peri <<endl;
}

void Area_of_square()
{
	int l,area;

	cout <<endl<< "-> Enter length : ";
	cin  >>l;

	area = l*l;
	cout <<endl<<"-> Area of square : "<<area <<endl;
}

void Area_of_rectangle()
{
	int l,area,b;

	cout <<endl<< "-> Enter length : ";
	cin  >>l;

	area = l*b;
	cout <<endl<<"-> Area of rectangle : "<<area <<endl;
}

void Area_of_triangle()
{
	int l,b,area;
```

```
        cout <<endl<< "-> Enter length : ";
        cin  >>l;
        cout <<endl<< "-> Enter breadth : ";
        cin  >>b;

        area = (l*b)/2;
        cout <<endl<<"-> Area of triangle : "<<area <<endl;
}

void Area_of_sphere()
{
        int r,pi=3.14,area;

        cout <<endl<< "-> Enter Radius : ";
        cin  >>r;

        area = 4*3.14*r*r;
        cout <<endl<<"-> Area of sphere : "<<area <<endl;
}

class All_geometry
{
        public:
                int choice;

        public:

                void AllData()
                {
                  do
                  {
                        List();
                        cout <<endl<<"-> Enter your choice : ";
                        cin  >> choice;

                        if(choice==1)
                        {
                                Area_of_circle();
                        }
                        else if(choice==2)
```

```
                {
                        Perimeter_of_circle();
                }
                else if(choice==3)
                {
                        Area_of_square();
                }
                else if(choice==4)
                {
                        Area_of_rectangle();
                }
                else if(choice==5)
                {
                        Area_of_triangle();
                }
                else if(choice==6)
                {
                        Area_of_sphere();
                }
                else if(choice!=0)
                {
                        cout <<"break";
                }
        }while (choice!=0);
    }
};

int main()
{
        All_geometry g1;
        g1.AllData();
        return 0;
}
```

Output:


M:\Flutter -(Lab work)\C++\FINAL ASSIGNMENT\Phase-5\13.exe
(1) Area of circle :
(2) Perimeter of circle :
(3) Area of square :
(4) Area of rectangle :
(5) Area of triangle :
(6) Area of sphere :
(0) Exit
-> Enter your choice : 1
-> Enter Radius of circle : 3
-> Area of circle : 28
(1) Area of circle :
(2) Perimeter of circle :
(3) Area of square :
(4) Area of rectangle :
(5) Area of triangle :
(6) Area of sphere :
(0) Exit
-> Enter your choice : 2
-> Enter Radius of circle : 4
-> Perimeter of circle : 25
(1) Area of circle :
(2) Perimeter of circle :
(3) Area of square :
(4) Area of rectangle :
(5) Area of triangle :
(6) Area of sphere :
(0) Exit
-> Enter your choice : 0

Process exited after 12.57 seconds with return value 0
Press any key to continue . . .

Practical-14

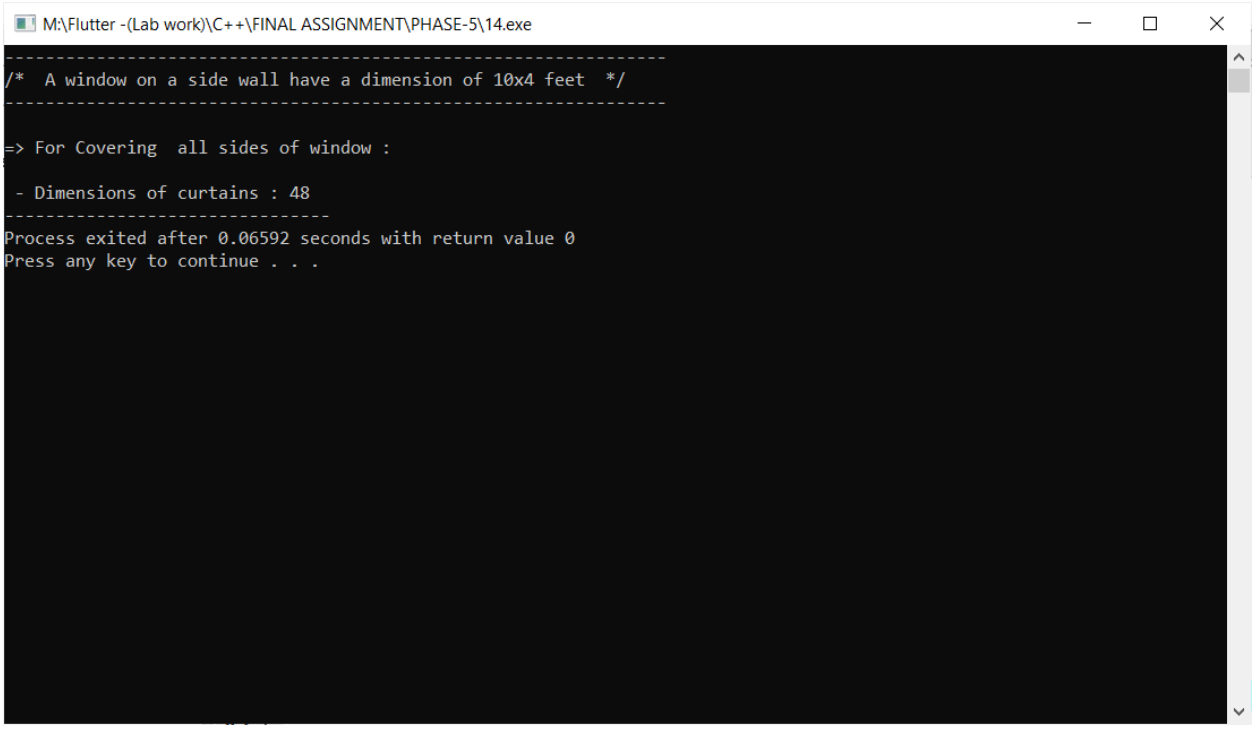
Aim: A window on a side wall have a dimension of 10x4 feet. Kaveri wants to apply curtains on that window such that a window will perfectly covered from all sides with extra 2 feet. Design a C++ UDF with figures out if a given dimensions of curtains satisfies mentioned criteria or not.

Program:

```
#include<iostream>
#include<string.h>
using namespace std;

void Dimension()
{
	int l=10 , w=4 , e_l=2 , k;

	cout <<"----------------------------------------------------------------";
	cout <<endl<<"/*  A window on a side wall have a dimension of 10x4 feet  */"<<endl;
	cout <<"----------------------------------------------------------------"<<endl;

	cout <<endl<<"=> For Covering  all sides of window : "<<endl;

	k = (l*w)+(4*e_l);

	cout <<endl<<" - Dimensions of curtains : "<<k;
}

class WC_Dimension
{
	public:
		void setData()
		{
			Dimension();
		}
```

```
};

int main()
{
	WC_Dimension w1;
	w1.setData();

	return 0;
}
```

Output:



Practical-15

Aim: Determine how many phrases(of 350x90 px dimensions) are perfectly arranged in an A4 size Canvas with distance of 8 px between all phrases. Develop a C++ UDF to count total number of phrases arranged in an A4 size Canvas.

Program:

Output: