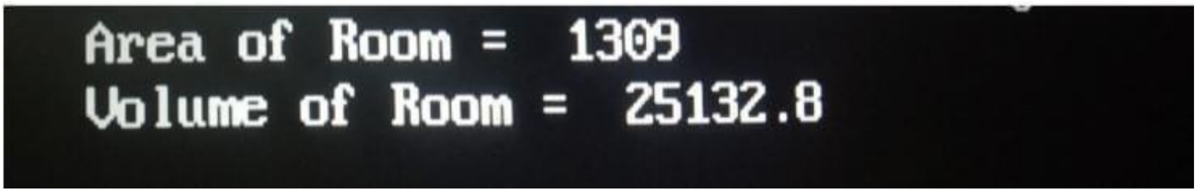
PROBLEM SHEET-1

1.Write a C++ Program that demonstrates the use of Class and Object1.Write a C++ Program that demonstrates the use of Class and Object

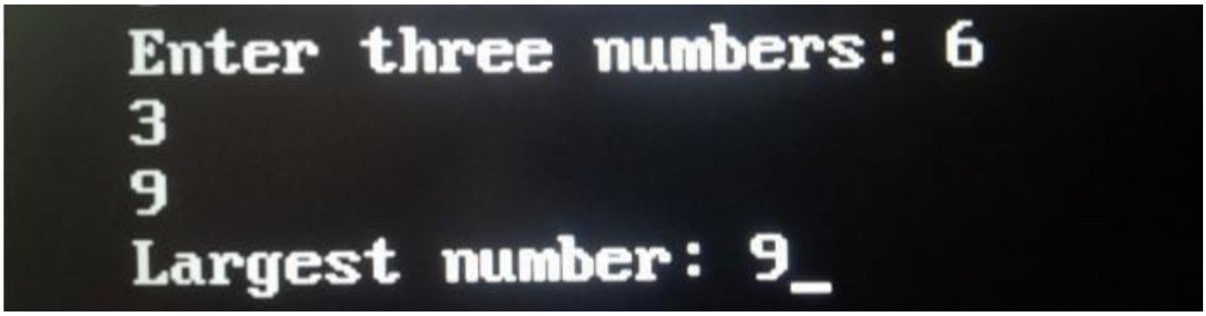
```
#include <iostream.h>
#include<conio.h>
// create a class
class Room {
public:
 double length;  double
breadth;  double height;
double calculateArea() {
return length * breadth;
 }
 double calculateVolume() {
return length * breadth * height;
 } };
int main() {
 // create object of Room class
 Room room1;
 // assign values to data members
room1.length = 42.5;  room1.breadth
= 30.8;  room1.height = 19.2;
 // calculate and display the area and volume of the room
 cout << "Area of Room = " << room1.calculateArea() << endl;  cout
<< "Volume of Room = " << room1.calculateVolume() << endl;
getch();  return 0;
}
```



```
Area of Room =  1309
Volume of Room =  25132.8
```

2.Write a C++ Program that find Largest from 3 Numbers(Take User input)

```
#include <iostream.h> #include<conio.h>

int main() {


 double n1, n2, n3;  cout <<

"Enter three numbers: ";  cin >>

n1 >> n2 >> n3;

 // check if n1 is the largest number

if(n1 >= n2 && n1 >= n3)  cout <<

"Largest number: " << n1;  // check

if n2 is the largest number  else

if(n2 >= n1 && n2 >= n3)  cout <<

"Largest number: " << n2;


 // if neither n1 nor n2 are the largest, n3 is the largest

else

 cout << "Largest number: " << n3;

getch();  return 0;

}
```



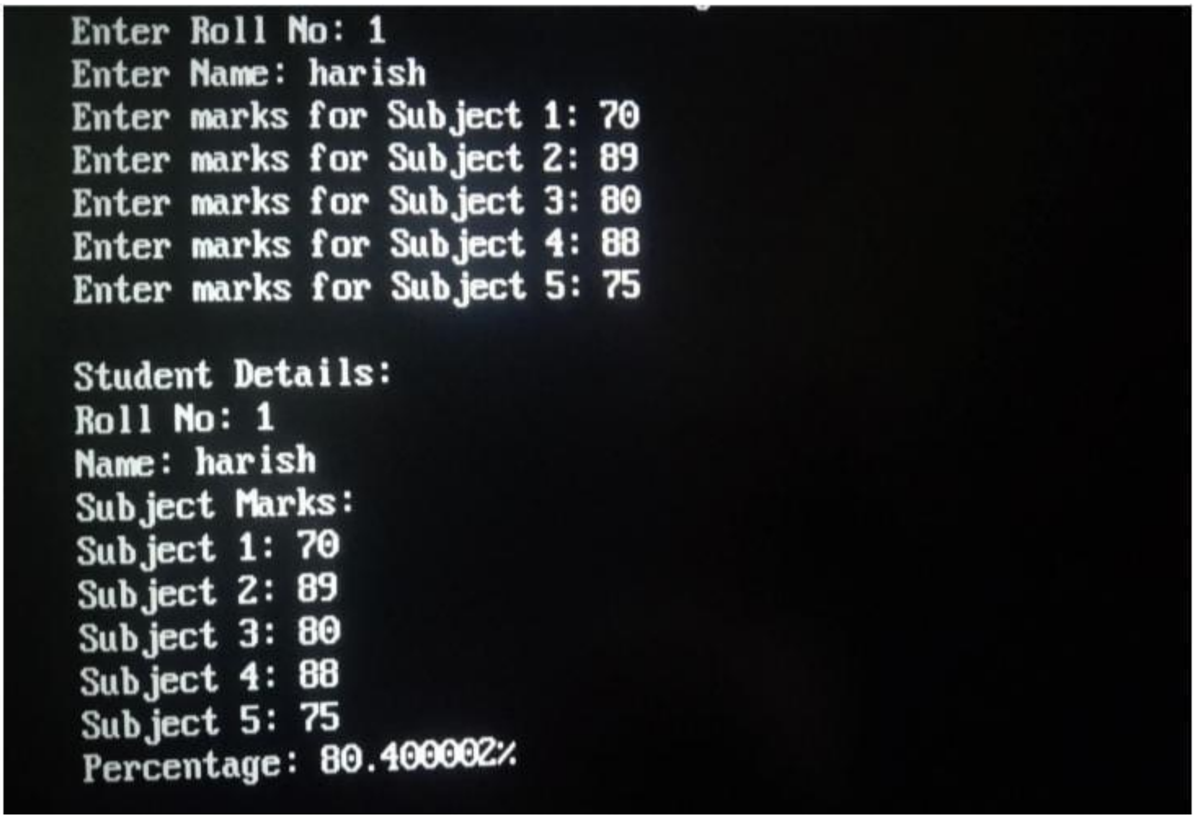
```
Enter three numbers: 6
3
9
Largest number: 9_
```

3.Write a C++ Program that Count Percentage of Student.

Input Roll_No, Name and 5 Subject marks) .

```
#include <iostream.h>

#include <conio.h>

#include <string.h>

int main() {  int

rollNo;

 char name[50];  float marks[5];  cout << "Enter Roll No: ";  cin >> rollNo;  cout <<

"Enter Name: ";  cin.ignore(); // Ignore the newline character left in the buffer after

entering rollNo  cin.getline(name, sizeof(name));  float totalMarks = 0;  for (i = 0; i <

5; ++i) {

 cout << "Enter marks for Subject " << i + 1 << ": ";

cin >> marks[i];

totalMarks += marks[i];

}

 float percentage = (totalMarks / (5 * 100)) * 100;

cout << "\nStudent Details:\n";  cout << "Roll No:

" << rollNo << endl;  cout << "Name: " << name

<< endl;  cout << "Subject Marks:\n";

 for (int i = 0; i < 5; ++i) {
```

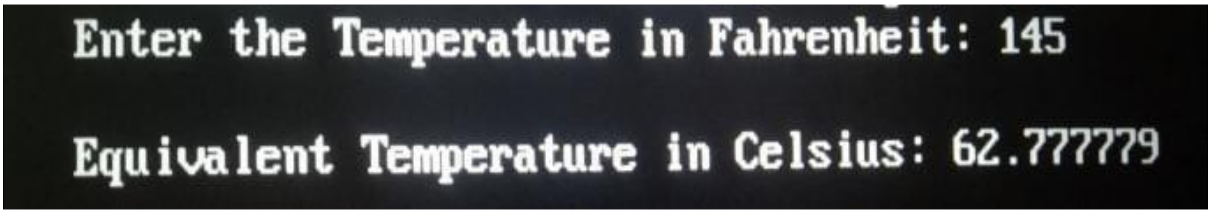
```
 cout << "Subject " << i + 1 << ": " << marks[i] << endl;

 }

 cout << "Percentage: " << percentage << "%" << endl;

 getch();

return 0;

}
```



```
Enter Roll No: 1
Enter Name: harish
Enter marks for Subject 1: 70
Enter marks for Subject 2: 89
Enter marks for Subject 3: 80
Enter marks for Subject 4: 88
Enter marks for Subject 5: 75

Student Details:
Roll No: 1
Name: harish
Subject Marks:
Subject 1: 70
Subject 2: 89
Subject 3: 80
Subject 4: 88
Subject 5: 75
Percentage: 80.400002%
```

4.Write a C++ program that will ask for a temperature in Fahrenheit and display it in Celsius.

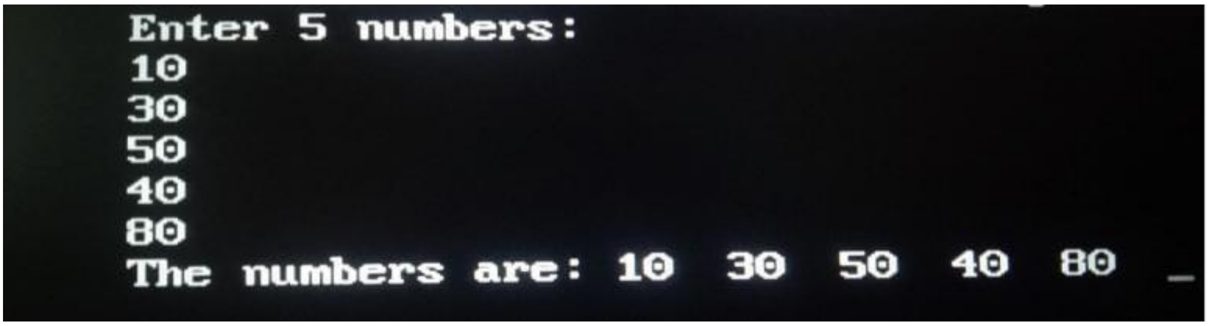
```
#include<iostream.h> #include<conio.h>

int main()

{

 float fahrenheit, celsius;  cout<<"Enter the Temperature

in Fahrenheit: ";  cin>>fahrenheit;  celsius = (fahrenheit-
32)/1.8;  cout<<"\nEquivalent Temperature in Celsius:
"<<celsius;  cout<<endl;  getch();  return 0;
}
```



```
Enter the Temperature in Fahrenheit: 145

Equivalent Temperature in Celsius: 62.777779
```

5.Write a C++ Program that demonstrates the use of Array.

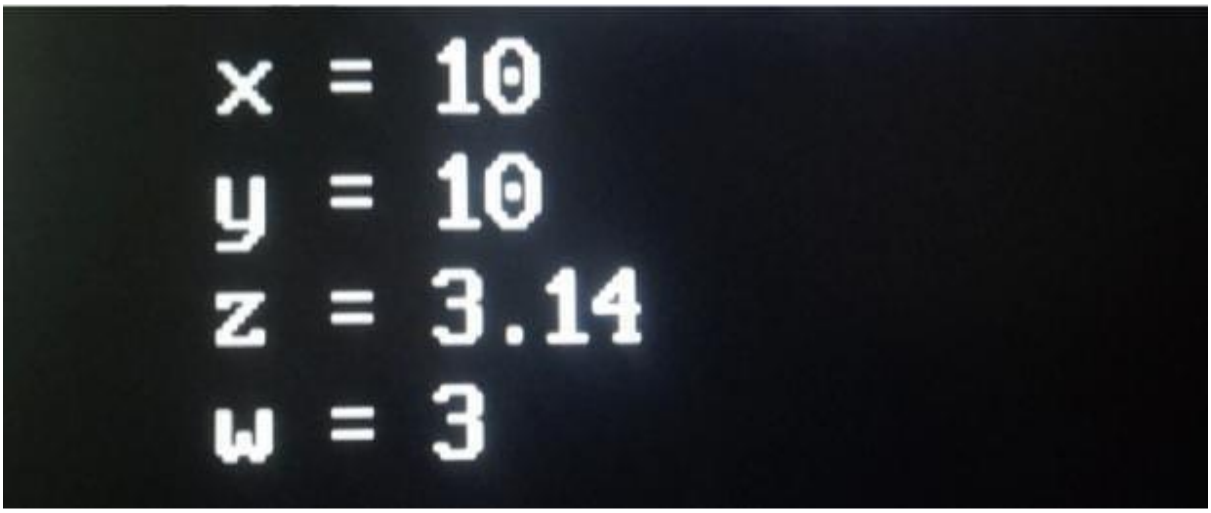
```
#include <iostream.h> #include<conio.h>
int main() {  int numbers[5];  cout <<
"Enter 5 numbers: " << endl;  //
store input from user to array

for (int i = 0; i < 5; ++i) {
cin >> numbers[i];
 }
 cout << "The numbers are: ";
// print array elements  for
(int n = 0; n < 5; ++n) {
 cout << numbers[n] << " ";
 }
 getch();
return 0;
}
```



```
Enter 5 numbers:
10
30
50
40
80
The numbers are: 10 30 50 40 80 _
```

6.Write a C++ Program that demonstrates the use of Type Conversion(Implicit & Explicit Both)

```
// A C++ program that demonstrates the use of type conversion

#include <iostream.h>

#include<conio.h> int

main()

{

 // Implicit type conversion

 // The compiler automatically converts the lower data type to the higher data type

int x = 10; // x is an int  double y = x; // y is a double, x is implicitly converted to

double

cout << "x = " << x << endl; // prints 10 cout

<< "y = " << y << endl; // prints 10.0

 // Explicit type conversion

 // The programmer explicitly converts the data type using casting operators

double z = 3.14; // z is a double  int w = (int) z; // w is an int, z is explicitly

converted to int using (int)  cout << "z = " << z << endl; // prints 3.14  cout

<< "w = " << w << endl; // prints 3


 getch();

return 0;

}
```



```
x = 10
y = 10
z = 3.14
w = 3
```

7.Write a C++ Program that demonstrates the use of Scope Resolution

Operator

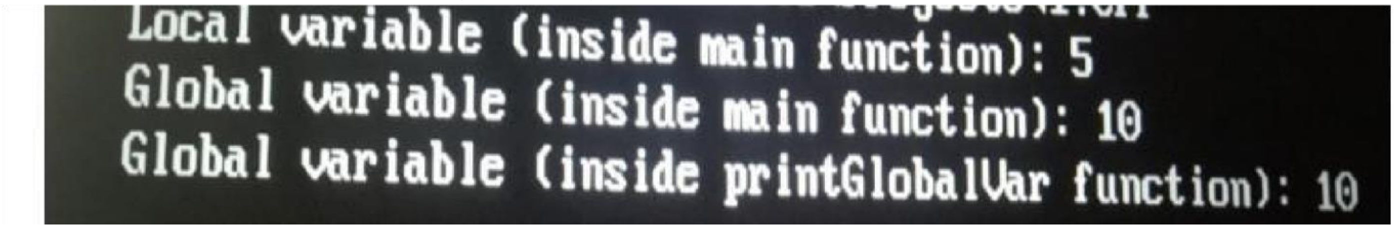
```
#include <iostream.h>

// Global variable int

globalVar = 10;

// Function declaration

void printGlobalVar(); int

main() {

 // Local variable with the same name as the global variable  int

globalVar = 5;  cout << "Local variable (inside main function): " <<

globalVar << endl;  cout << "Global variable (inside main function): " <<

::globalVar << endl;

 // Call the function to print the global variable

printGlobalVar();  return 0;

}
```

```
// Function definition to print the global variable void
printGlobalVar() {
 // Access the global variable using the scope resolution operator  cout << "Global
variable (inside printGlobalVar function): " << ::globalVar << endl;
}
```

8.Write a C++ Program that demonstrates the use of all set and endl

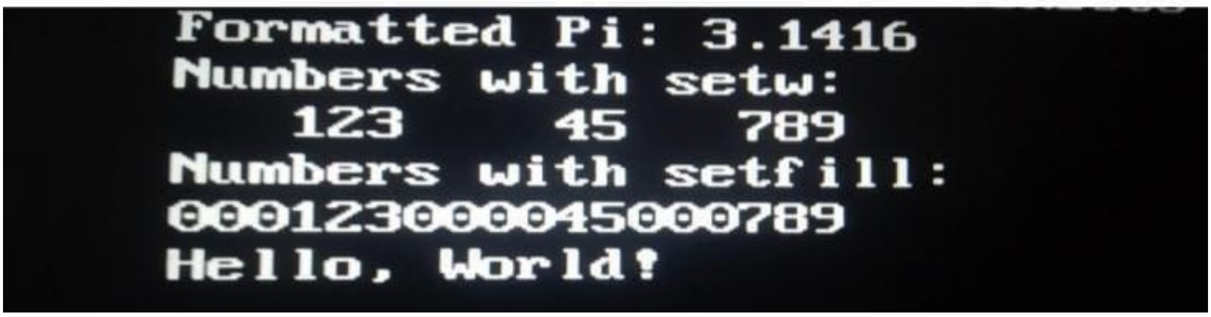
Manipulator.

```
#include <iostream.h>
#include <iomanip.h>
#include<conio.h> int
main() {
 // Using setprecision to format floating-point output
```



```
Local variable (inside main function): 5
Global variable (inside main function): 10
Global variable (inside printGlobalVar function): 10
```

```
double pi = 3.14159265359;  cout << "Formatted Pi: " <<
setprecision(4) << pi << endl;  // Using setw to set the
width of the field  int num1 = 123;  int num2 = 45;  int
num3 = 789;  cout << "Numbers with setw:\n";  cout <<
setw(6) << num1 << setw(6) << num2 << setw(6) <<
num3 << endl;
 // Using setfill to set the fill character
cout << "Numbers with setfill:\n";
 cout.fill('0'); // setfill for Turbo C++
 cout << setw(6) << num1 << setw(6) << num2 << setw(6) << num3 << endl;
 // Using endl to insert a new line and flush the buffer

cout << "Hello,";  cout << " World!" << endl;  getch();

return 0;

}
```

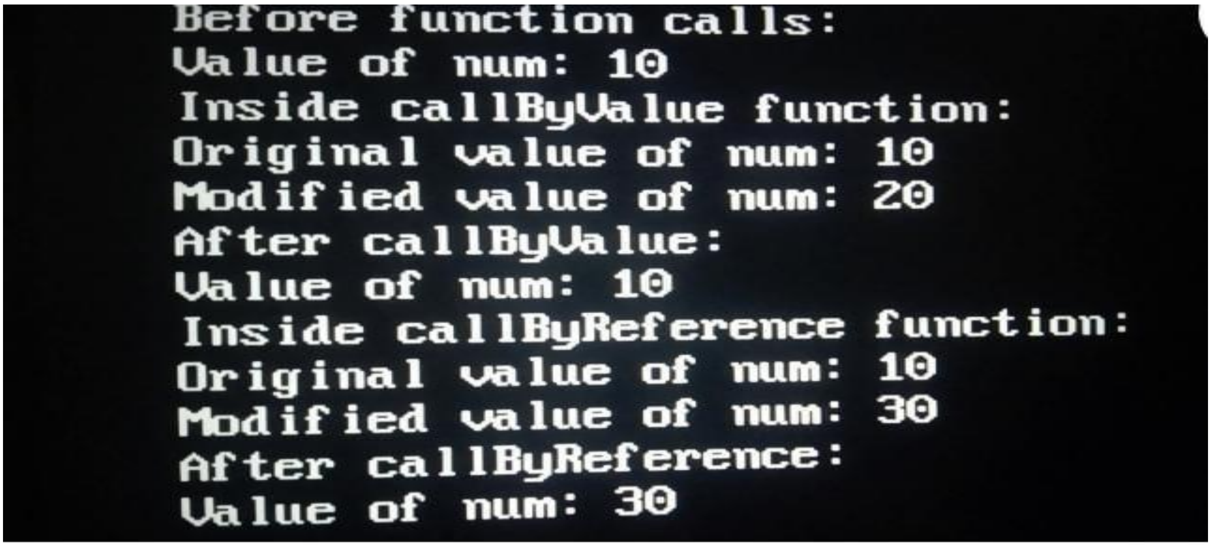


```
Formatted Pi: 3.1416
Numbers with setw:
     123     45     789
Numbers with setfill:
0001230000045000789
Hello, World!
```

9.Write a C++ Program that demonstrates the use of Call by Value and Call By Reference.

```
#include <iostream.h>

#include<conio.h>

// Call by Value: Function takes arguments by value void

callByValue(int num) {

 cout << "Inside callByValue function:\n";  cout <<

"Original value of num: " << num << endl;  num =

20; // Modifying the local copy of num  cout <<

"Modified value of num: " << num << endl;

}

// Call by Reference: Function takes arguments by reference void

callByReference(int& num) {  cout << "Inside callByReference function:\n";

cout << "Original value of num: " << num << endl;  num = 30; // Modifying the
```

```
original value of num (since we have the reference)  cout << "Modified value of
num: " << num << endl;
}
int main() {  int num = 10;  cout <<
"Before function calls:\n";  cout << "Value
of num: " << num <<endl;
callByValue(num);  cout << "After
callByValue:\n";  cout << "Value of num: "
<< num << endl;
 // Call by Reference
callByReference(num);  cout << "After
callByReference:\n";  cout << "Value of
num: " << num << endl; getch();

return 0;
```

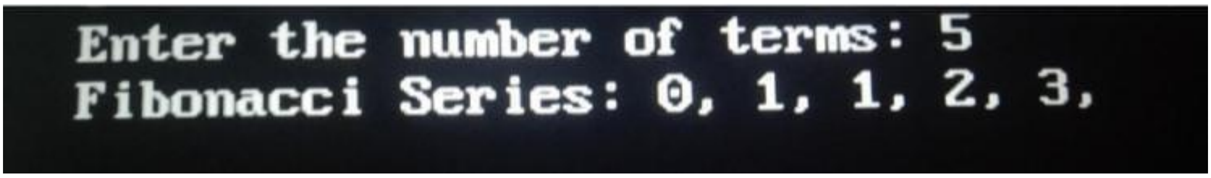


```
Before function calls:
Value of num: 10
Inside callByValue function:
Original value of num: 10
Modified value of num: 20
After callByValue:
Value of num: 10
Inside callByReference function:
Original value of num: 10
Modified value of num: 30
After callByReference:
Value of num: 30
```

10.Write a C++ program that will find Factorial of Given no.

```
#include <iostream.h>
#include<conio.h>
```

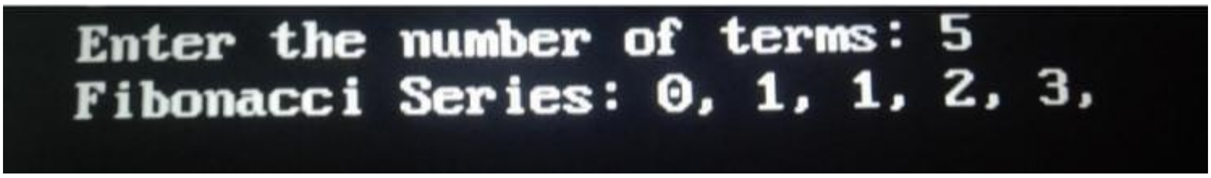
```
int main() {
 int n;  long factorial = 1.0;  cout <<
"Enter a positive integer: ";  cin >>
n;  if (n < 0)
 cout << "Error! Factorial of a negative number doesn't exist.";
 else {  for(int i = 1; i <=
n; ++i) {  factorial *= i;
 }
 cout << "Factorial of " << n << " = " << factorial;
 }
 getch();
return 0;
}
```



```
Enter the number of terms: 5
Fibonacci Series: 0, 1, 1, 2, 3,
```

11.Write C++ programs that will Print Fibonacci Series.

```
#include <iostream.h>
#include<conio.h>
int main() {  int n, t1 = 0, t2 = 1,
nextTerm = 0;  cout << "Enter the
number of terms: ";
 cin >> n;  cout << "Fibonacci
Series: ";
 for (int i = 1; i <= n; ++i) {  //
Prints the first two terms.
 if(i == 1) {
 cout << t1 << ", ";
continue;
 }
 if(i == 2) {
 cout << t2 << ", ";
continue;
 }
 nextTerm = t1 + t2;
 t1 = t2;
 t2 = nextTerm;

 cout << nextTerm << ", ";
 }
 getch();
 return 0;
}
```



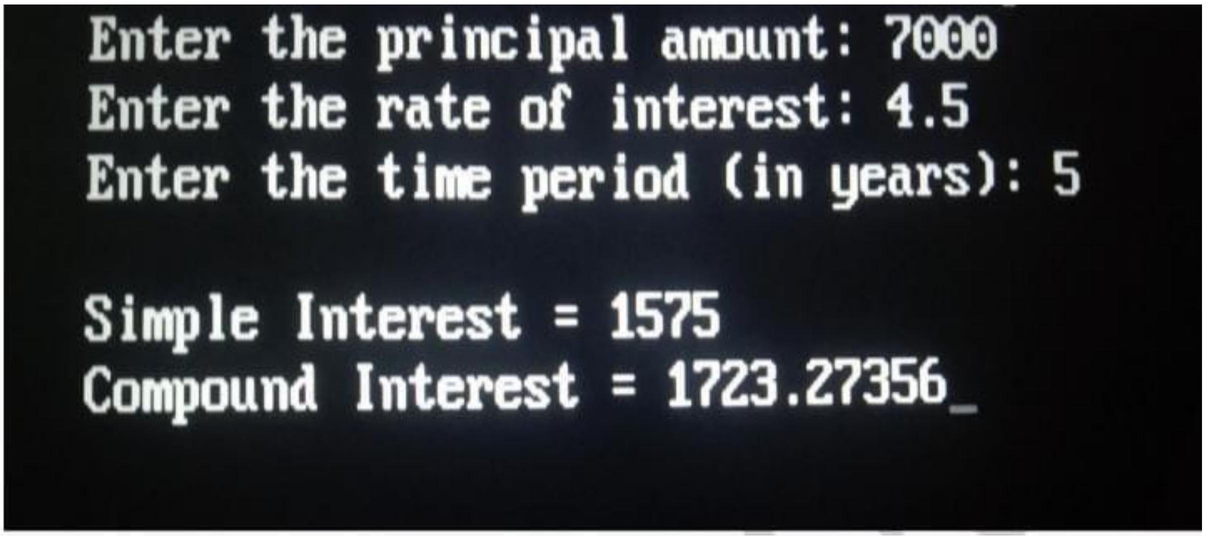
```
Enter the number of terms: 5
Fibonacci Series: 0, 1, 1, 2, 3,
```

12.Write a C++ Program that will find Simple Interest and Compound

Interest.

```
#include <iostream.h>
#include <math.h> #include<conio.h>
int main()
{  float p, r, t, si,
ci;
```

```
 cout << "Enter the principal amount: ";  cin
>> p;  cout << "Enter the rate of interest: ";
cin >> r;  cout << "Enter the time period (in
years): ";
 cin >> t;  si = (p * r * t) / 100;  ci = p *
pow((1 + r / 100), t) - p;  cout <<
"\nSimple Interest = " << si;  cout <<
"\nCompound Interest = " << ci;
 getch();
return 0;
}
```

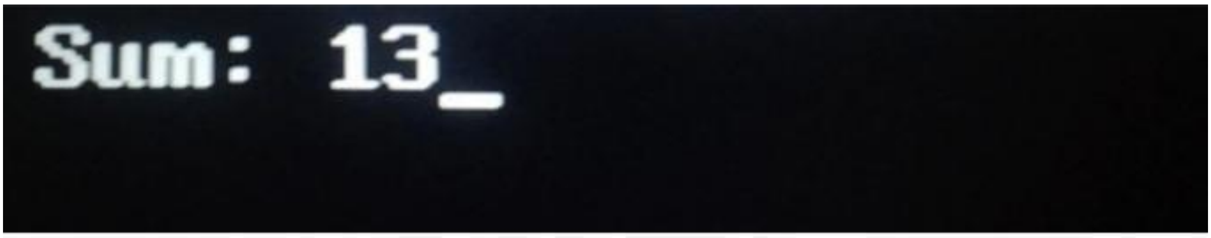


```
Enter the principal amount: 7000
Enter the rate of interest: 4.5
Enter the time period (in years): 5

Simple Interest = 1575
Compound Interest = 1723.27356_
```

13.Write a C++ Program that demonstrates the use of Friend Function.

```
#include<iostream.h>
#include<conio.h> class B; //
forward declaration class A {
private:  int numA;
```

```
 public:
 A(): numA(12) { }
 // friend function declaration
friend int add(A, B);
};
class B {
private:  int
numB;
 public:
 B(): numB(1) { }
 // friend function declaration
 friend int add(A, B);
};
// Function add() is the friend function of classes A and B
int add(A objectA, B objectB) {  return (objectA.numA +
objectB.numB);
}
int main() {  A objectA;  B objectB;
cout<<"Sum: "<< add(objectA, objectB);
 getch();
return 0;
}
```



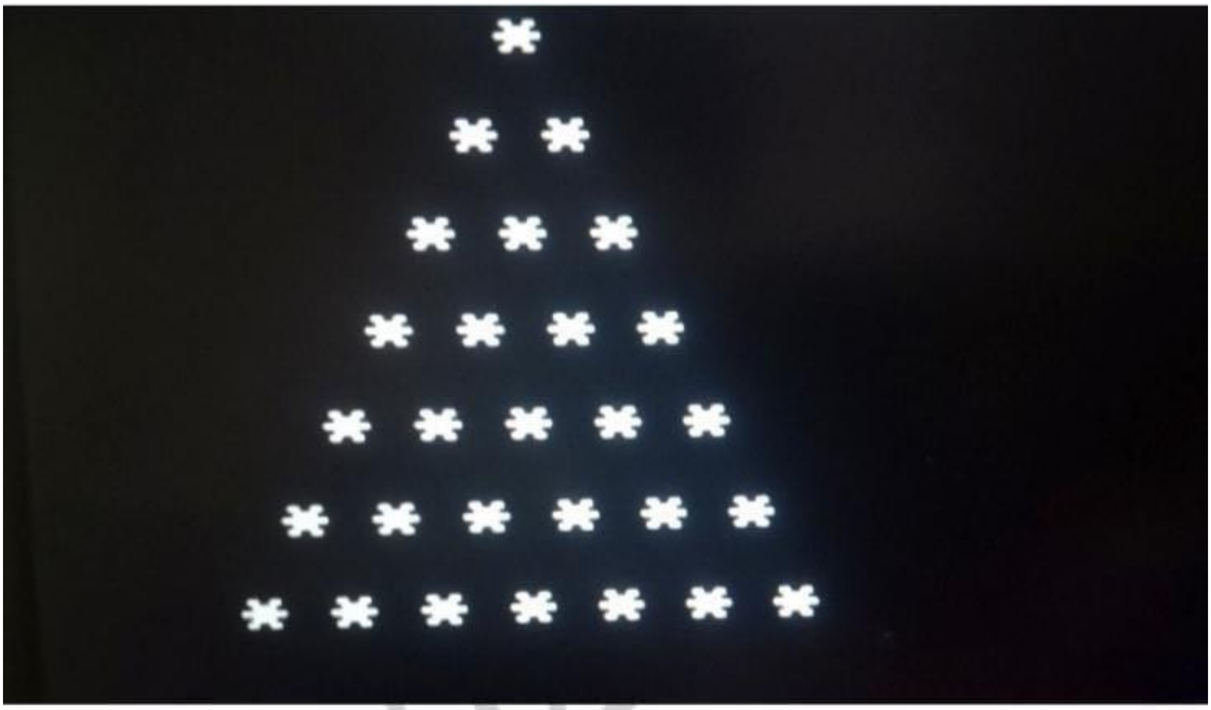
```
Sum: 13_
```

14.Write a C++ program that will print pyramid pattern of stars(*).

PROBLEM SHEET-1

```
// C++ code to demonstrate star pattern
#include <iostream.h>
#include<conio.h>
// Function to demonstrate printing pattern void
pypart2(int n)
{
// Number of spaces
int i, j, k = n;
// Outer loop to handle number of rows
// n in this case
for (i = 1; i <= n; i++) { //
Inner loop for columns
for (j = 1; j <= n; j++) {
// Condition to print star pattern
if (j >= k)
cout << "* ";
else cout <<
" ";
}
k--;
cout << "\n";
}
}
// Driver Code int
main()
{
```

```
int n = 7; //
Function Call
pypart2(n);
 getch(); return
0;
}
```

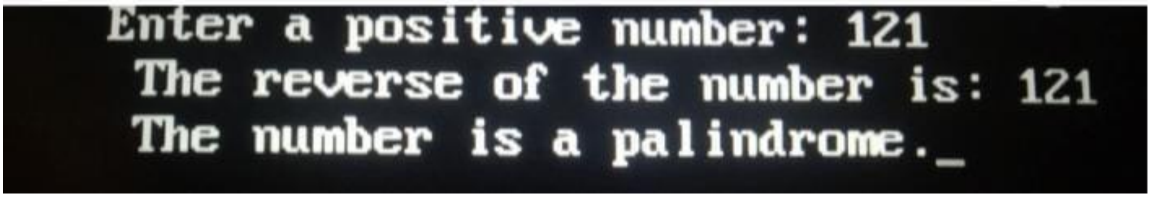


```
      *
     * *
    * * *
   * * * *
  * * * * *
 * * * * * *
* * * * * * *
```

15.Write a C++ program that will check no is palindrome or not.

```
#include <iostream.h>
#include<conio.h>
int main()
{
 int n, num, digit, rev = 0;  cout <<
"Enter a positive number: ";  cin >>
num;  n = num;  do
 {
```

```
 digit = num % 10;  rev
= (rev * 10) + digit;
 num = num / 10;
 } while (num != 0);  cout << " The reverse of the number
is: " << rev << endl;
 if (n == rev)

 cout << " The number is a palindrome.";
 else
 cout << " The number is not a palindrome.";


 getch();
return 0
```



```
Enter a positive number: 121
 The reverse of the number is: 121
 The number is a palindrome._
```