**LAB # 07**



**CSE102L Computer Programming Lab**

**Submitted by: Maaz habib**

**Registration/ No: PWCSE1952**

**Class Section : c**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**Engr. Abdullah Hamid**

Month Day, Year (06, 21, 2021)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar



**Lab 7:** Functions (Part 2 Overload, Templates and Recursion)

## Objectives:

To understand the programming of recursive functions and overloading functions

**Recursion**

* When a function calls itself, new local variables and parameters are allocated storage (usually on the system stack), and the function code is executed with these new variables from the start.
* The existence of functions makes possible a programming technique called recursion.
* Recursion involves a function calling itself.

**Example**

#include <iostream>

using namespace std;

unsigned long factfunc(unsigned long); //declaration

int main()

{

  int n; //number entered by user

  unsigned long fact; //factorial

  cout > n;

  fact = factfunc(n);

  cout << “Factorial of “ << n << “ is “ << fact << endl;

  return 0;

  }

// factfunc()

// calls itself to calculate factorials

unsigned long factfunc(unsigned long n)

{

  if(n > 1)

  return n \* factfunc(n-1); //self call

  else

  return 1;

}

**Output**

Enter a number: 10

Factorial is 3628800

**Function Overloading**

* In this section, you will learn about one of C++’s most exciting features: function overloading. In C++, two or more functions can share the same name as long as their parameter declarations are different.
* In this situation, the functions that share the same name are said to be overloaded, and the process is referred to as function overloading.
* Function overloading is one way that C++ achieves polymorphism.

**Example**

#include <iostream>

using namespace std;

void f(int i);

int main()

{

f(10);

return 0;

}

void f(int i)

{

  cout << " In f(int), i is " << i << '

';

}

**Output**

In f(int), i is 10

|  |  |
| --- | --- |
| [Previous](https://www.cseworldonline.com/cplusplus/cpp_Functions.php) | [Next](https://www.cseworldonline.com/cplusplus/cpp_Pointer.php) |

To understand function programming, its types and function-call

# Functions in C Programming with examples

BY CHAITANYA SINGH | FILED UNDER: [C-PROGRAMMING](https://beginnersbook.com/category/c-programming/)

In this tutorial, we will learn **functions in C programming**. A function is a **block of statements** that performs a specific task. Let’s say you are writing a C program and you need to perform a same task in that program more than once. In such case you have two options:

a) Use the same set of statements every time you want to perform the task  
b) Create a function to perform that task, and just call it every time you need to perform that task.

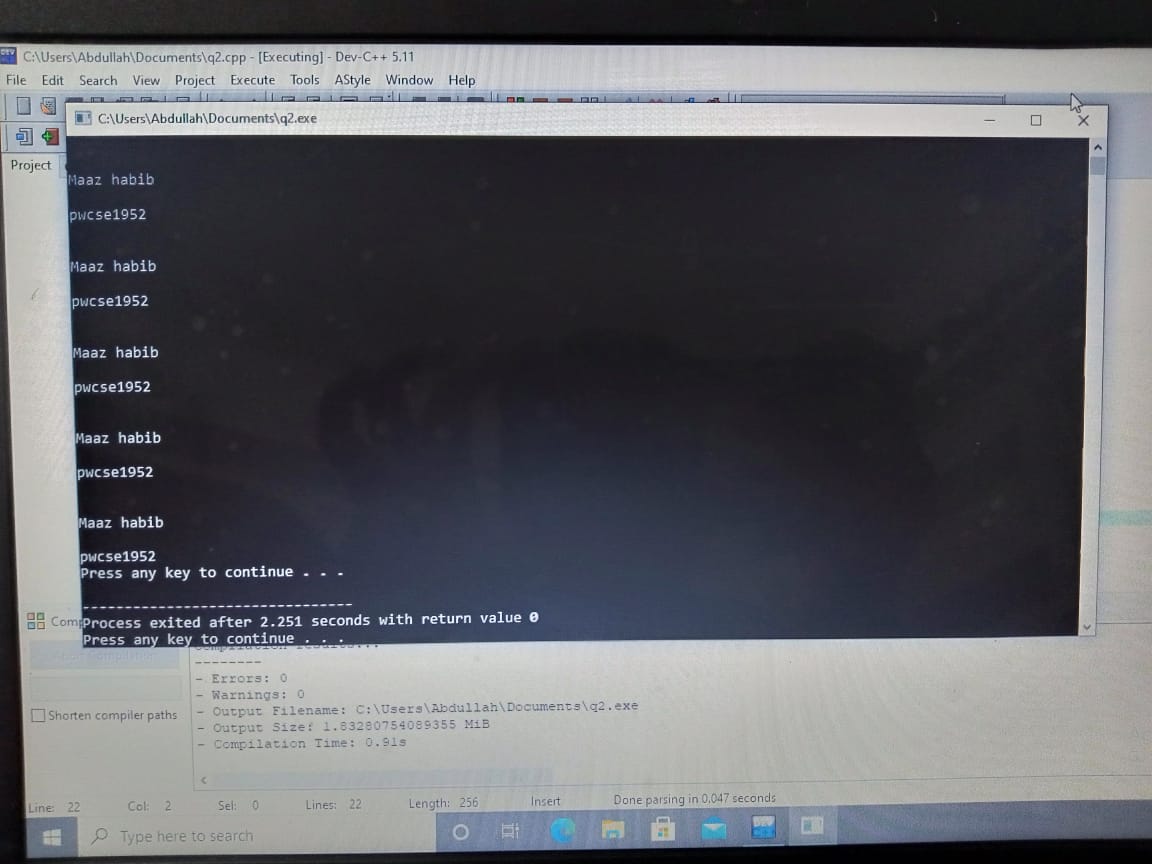
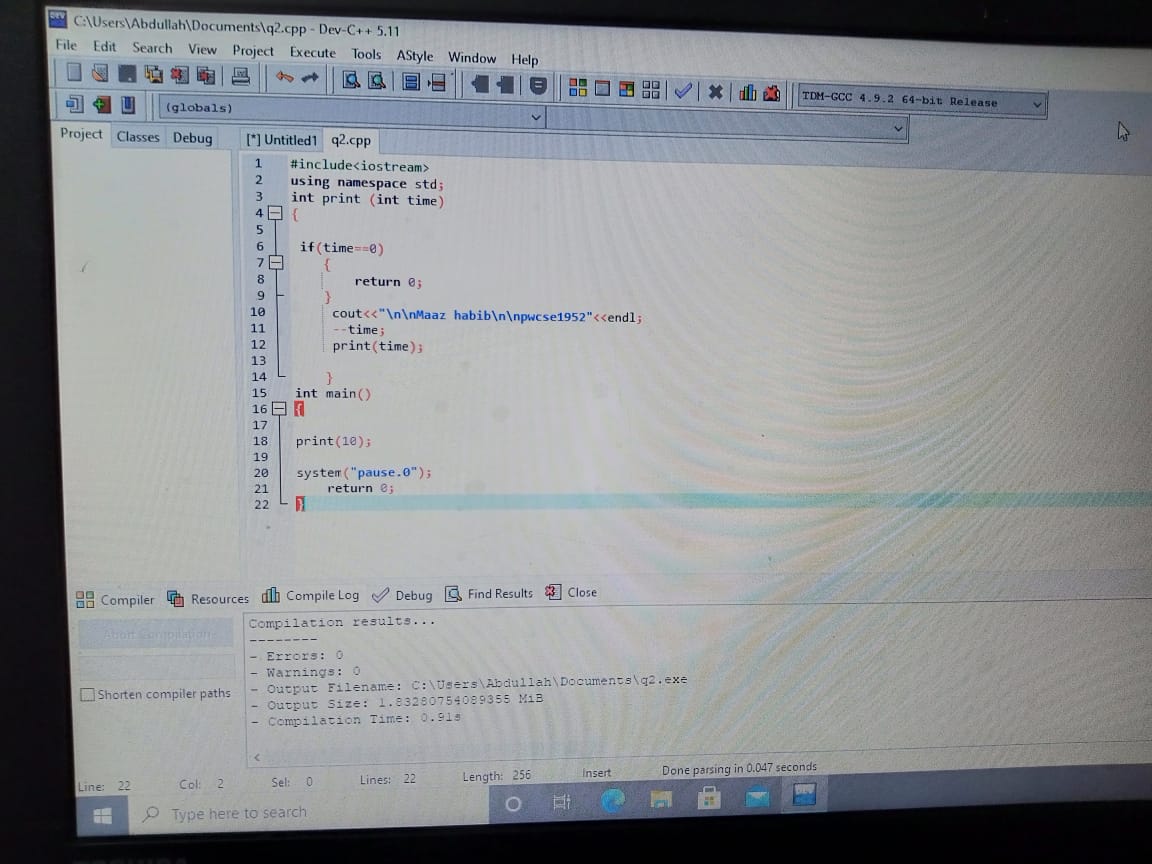
Using option (b) is a good practice and a **good programmer always uses functions while writing code in C**.

## Why we need functions in C

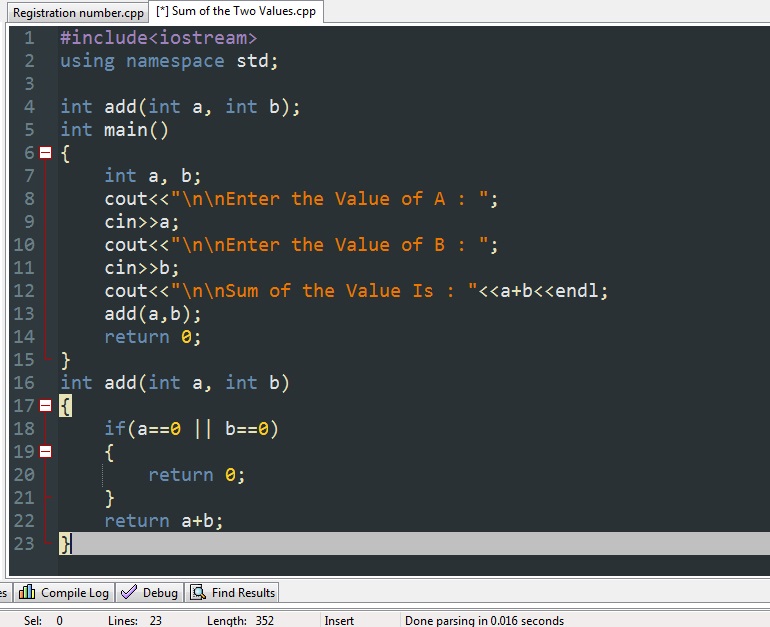
Functions are used because of following reasons –  
a) To improve the readability of code.  
b) Improves the reusability of the code, same function can be used in any program rather than writing the same code from scratch.  
c) Debugging of the code would be easier if you use functions, as errors are easy to be traced.  
d) Reduces the size of the code, duplicate set of statements are replaced by function calls.

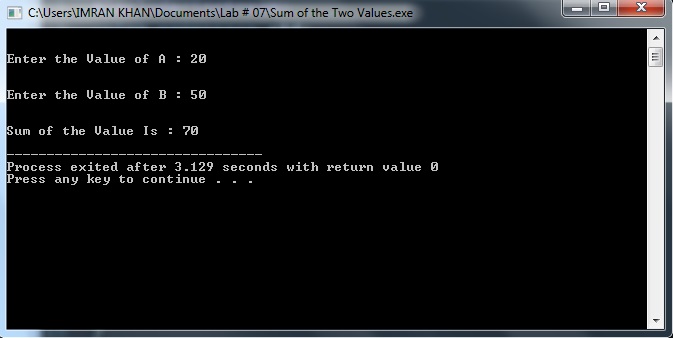
## Tasks:

1. Print your name and registration number 10 times in C++ using recursion.

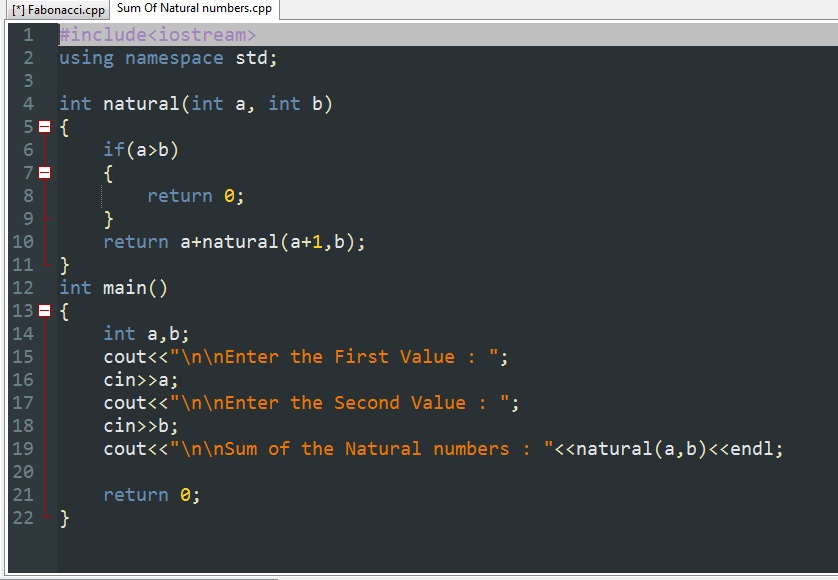


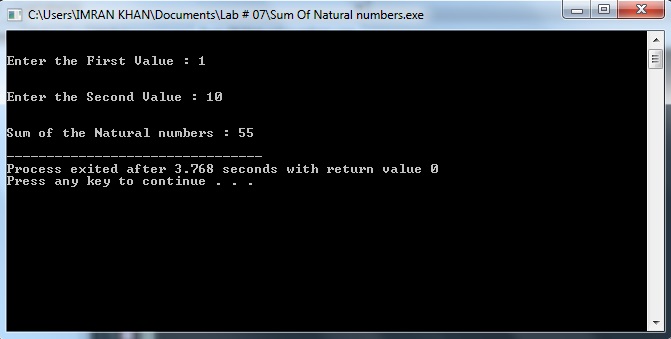
1. Write a C++ program where you take two values from the user if the user enters one or two of the values zero instead of passing the zero values to the function. Let the function calculate default values if the user enters values other than zero and pass them to the function and calculate their sum.



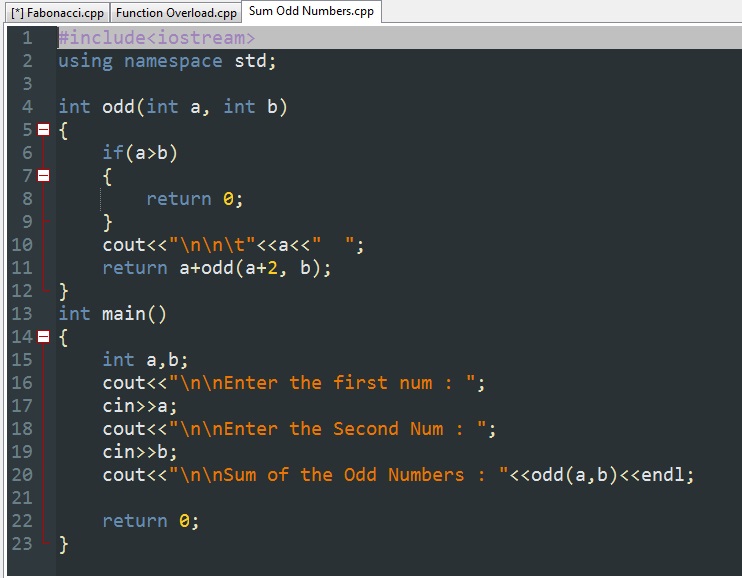


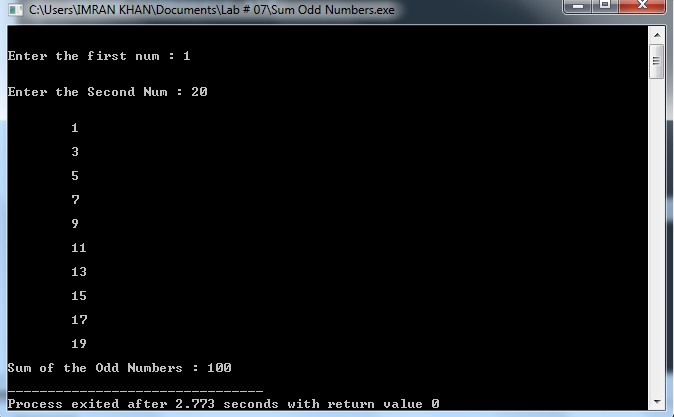
1. Write a function to find Sum of N natural numbers using Recursion.



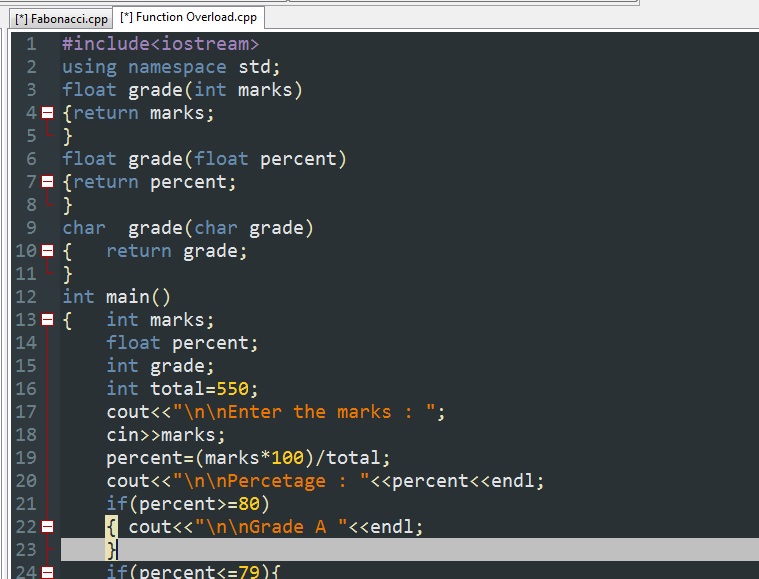


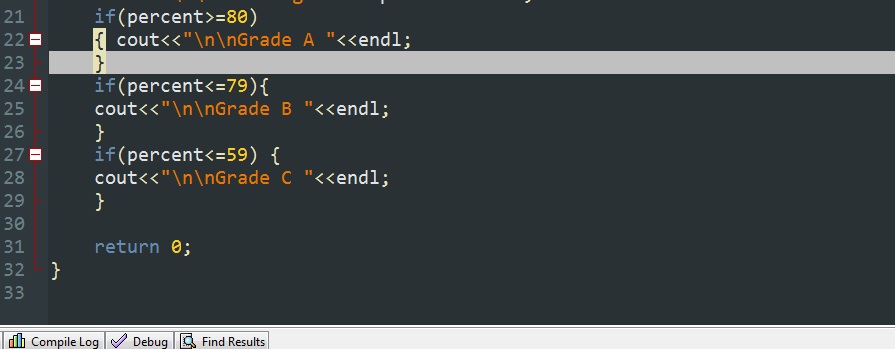
1. Calculate the sum of odd natural numbers 1+3+5+7+……………. + n using Recursion. Take n as input from the user.

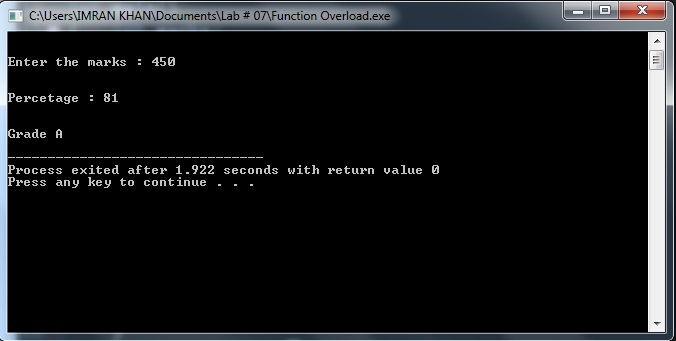




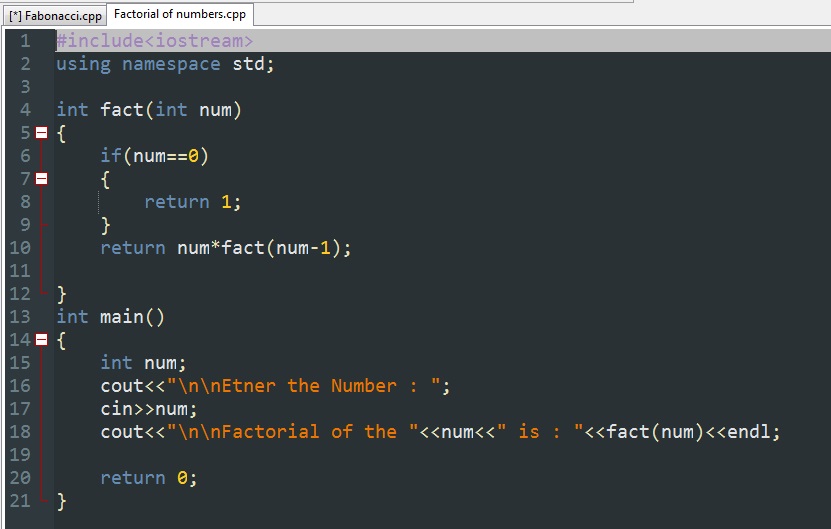
1. Overload three functions with name “grade”, the first grade function should be non-returning void type with no parameter, void grade(), the second should have integer parameter and return type float, float grade(int marks) the third function should have float parameter and its return type should be char, char grade(float percentage). You main should only call the first function, the first function will prompt the user to enter marks then it will pass the marks to the second function where it will calculate the percentage d return the percentage, then the first function will send the percentage to the third function where it will calculate the grade based on the percentage will return the grade to first function in the form of char. Finally, the first function will display the grade as well as the marks and the percentage.

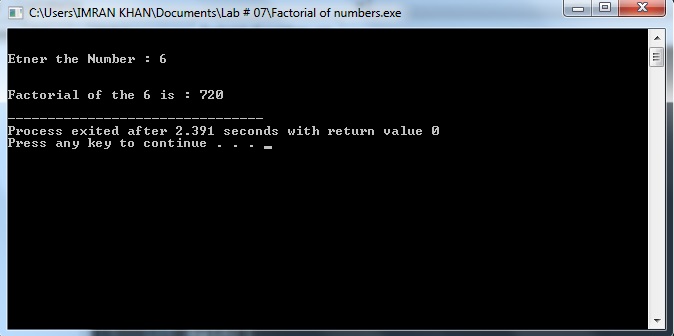




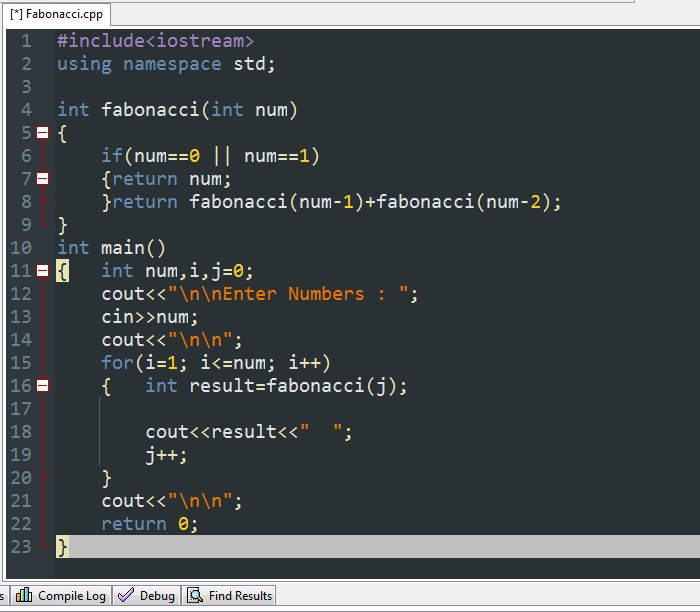


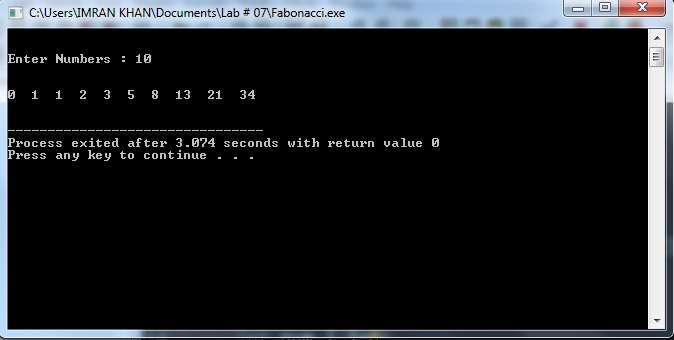
1. Write a C++ Program to Find Factorial of a Number Using Recursion.



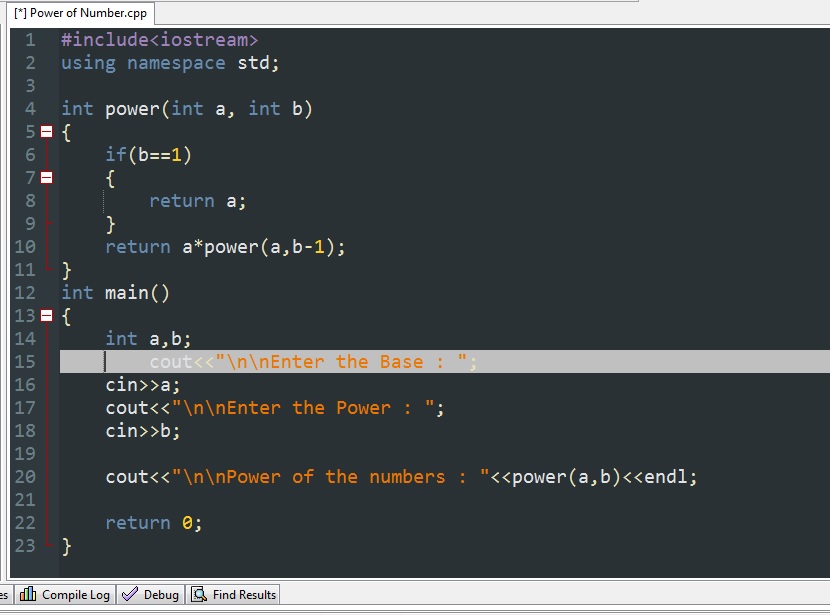


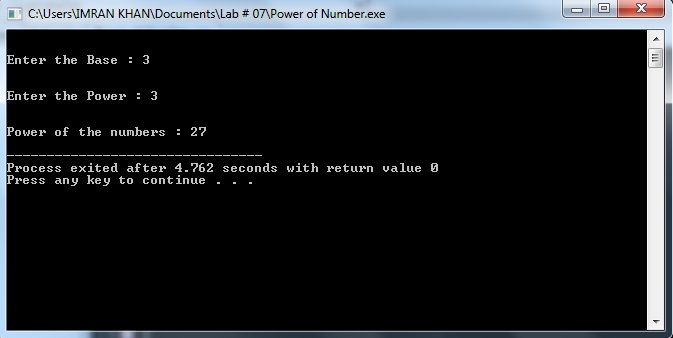
1. C++ program to print Fibonacci series using recursion.





1. C++ program to calculate power of a number using recursion.





\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*