## (SE - FUNDAMENTALS OF PROGRAMMING)

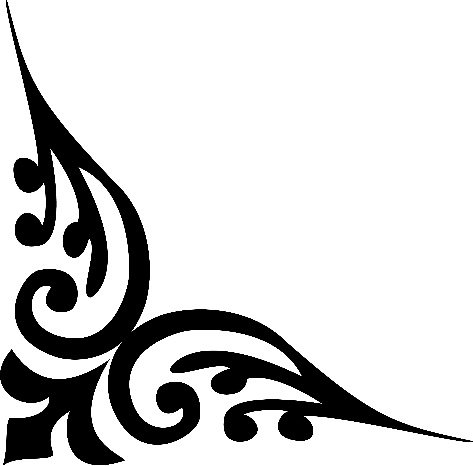
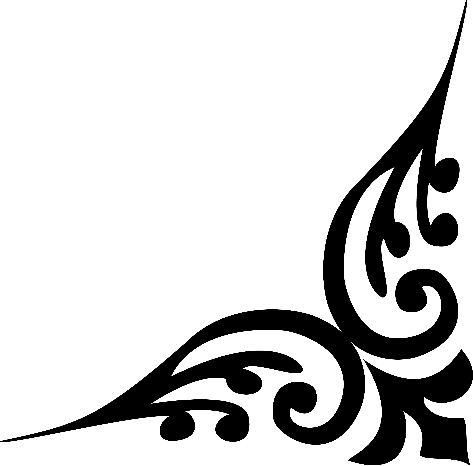
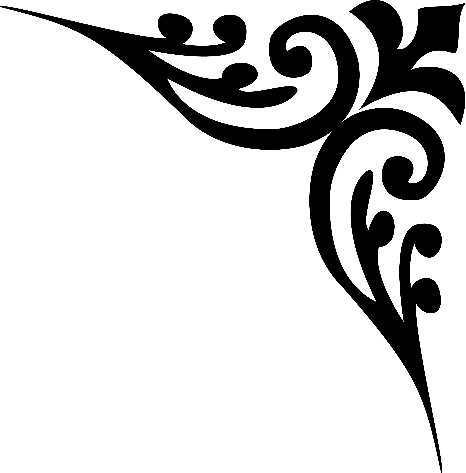
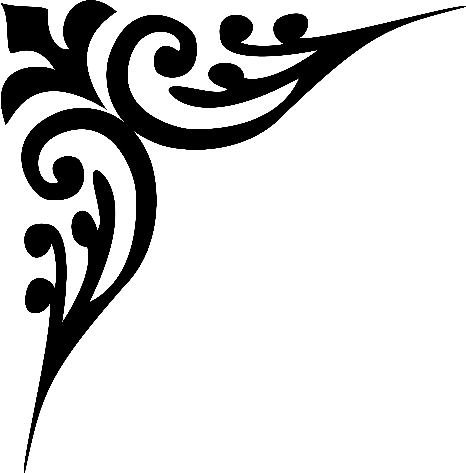
{C LANGUAGE} –

‘Fundamentals Of Programing’

**MODULES:- [*2.1,2.2,2.3*]**

**Submitted By**

**Mitesh Bhavsar**



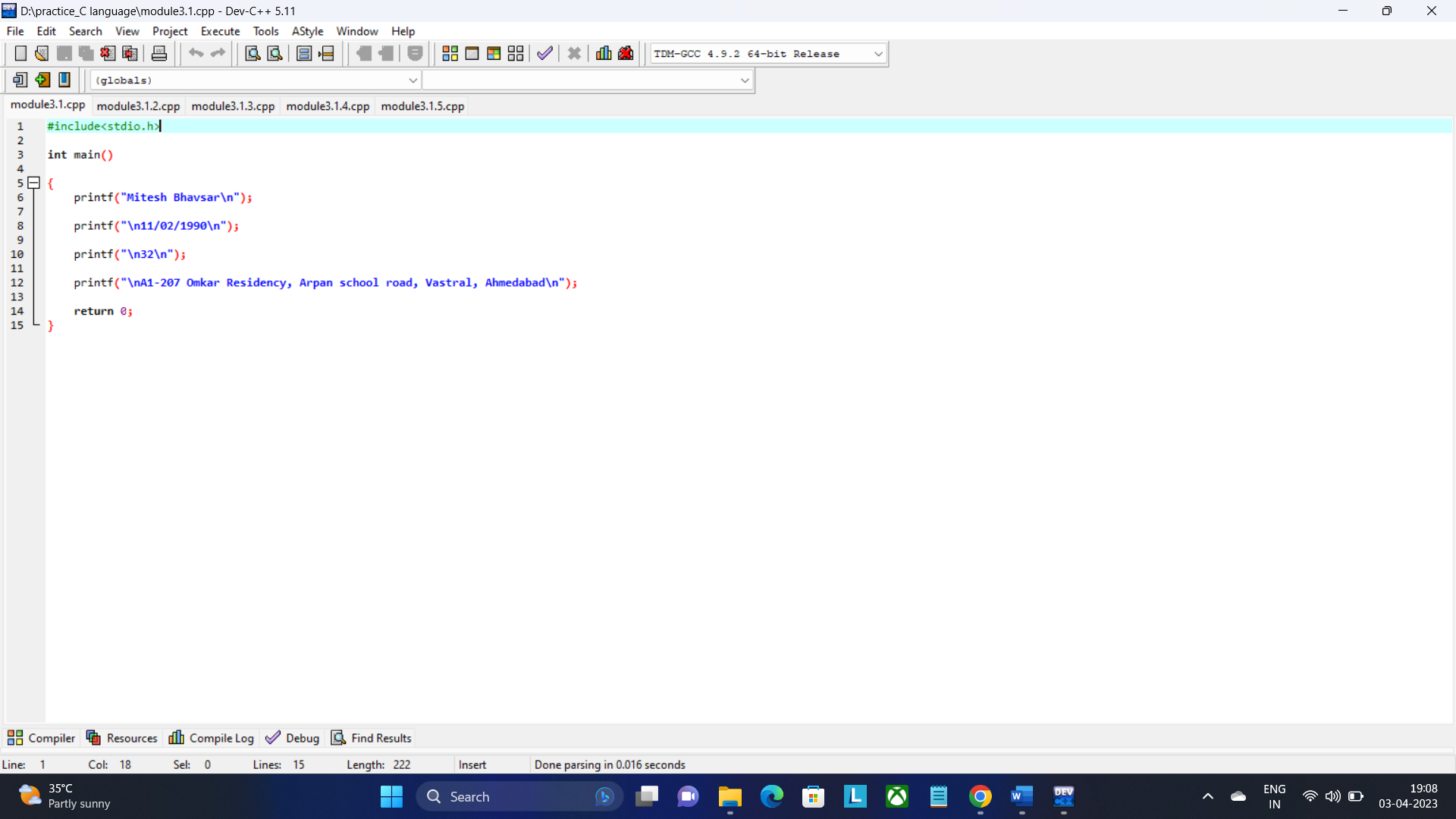
# MODULE: 2.1 (C Language Fundamental)

## Display This Information using printf :

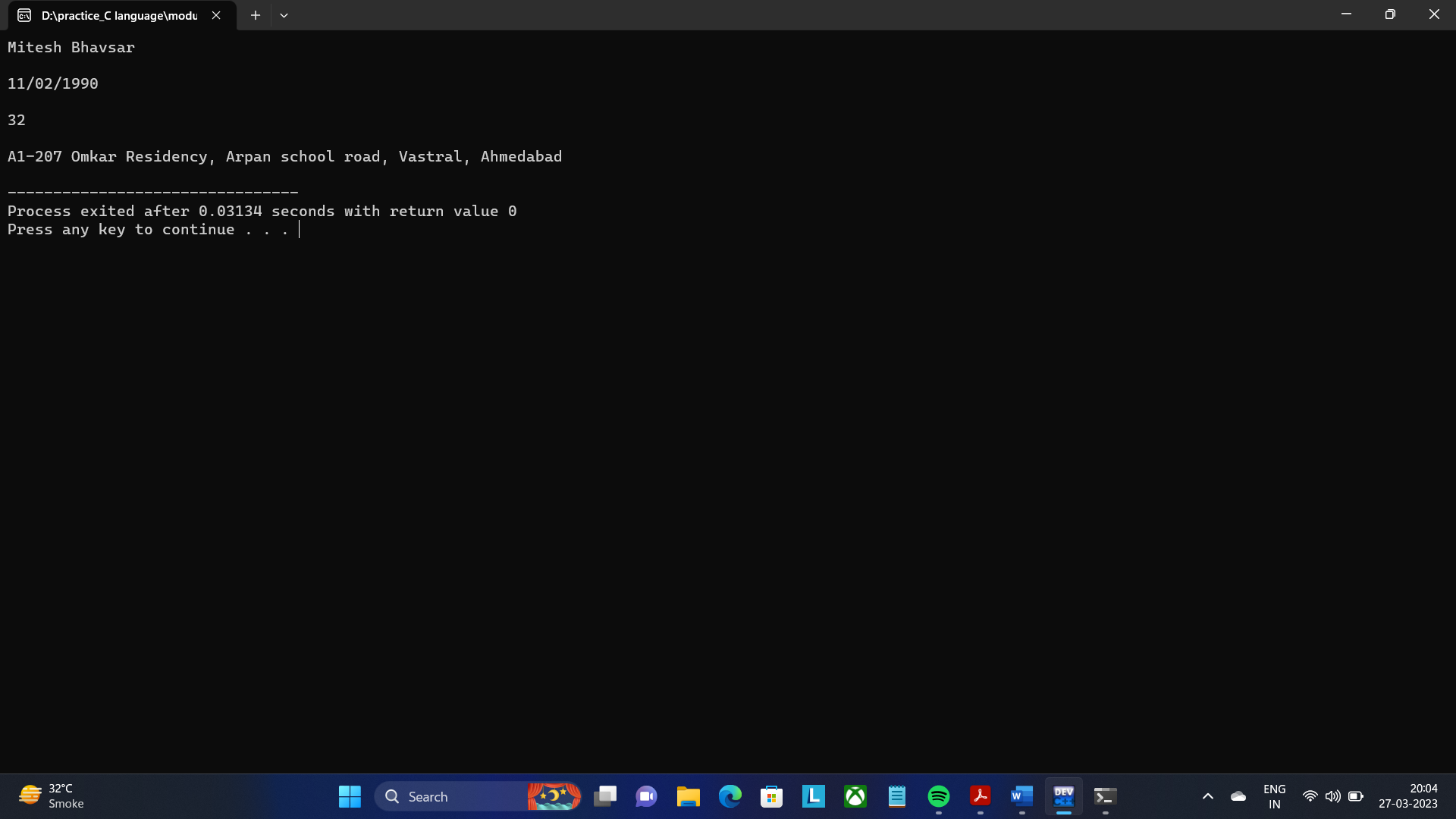
* 1. **Your Name**
  2. **Your Birth date**
  3. **Your Age**
  4. **Your Address**

**Ans :-**

**Practical:-**



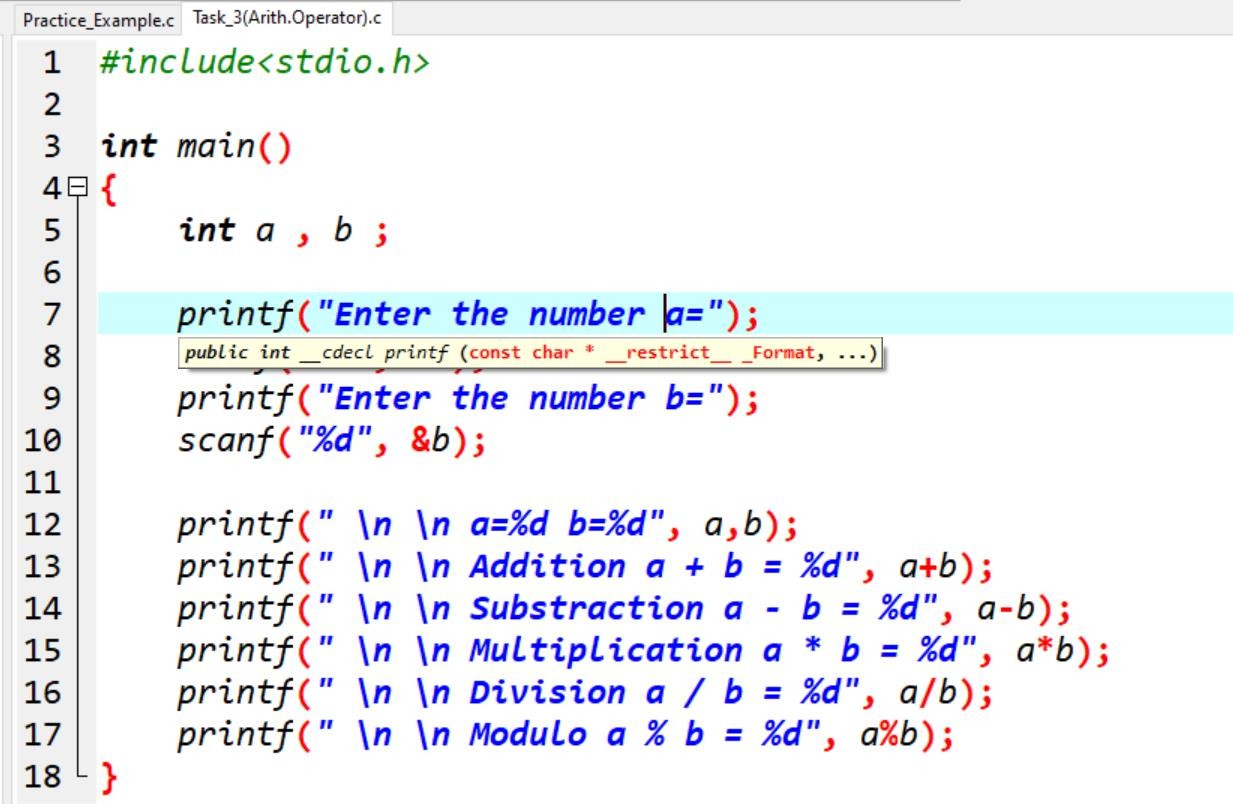
**Output:-**



## Write a program to make Simple calculator (to make addition, subtraction, multiplication, division and modulo)?

**Ans:-**

**Practical:-**



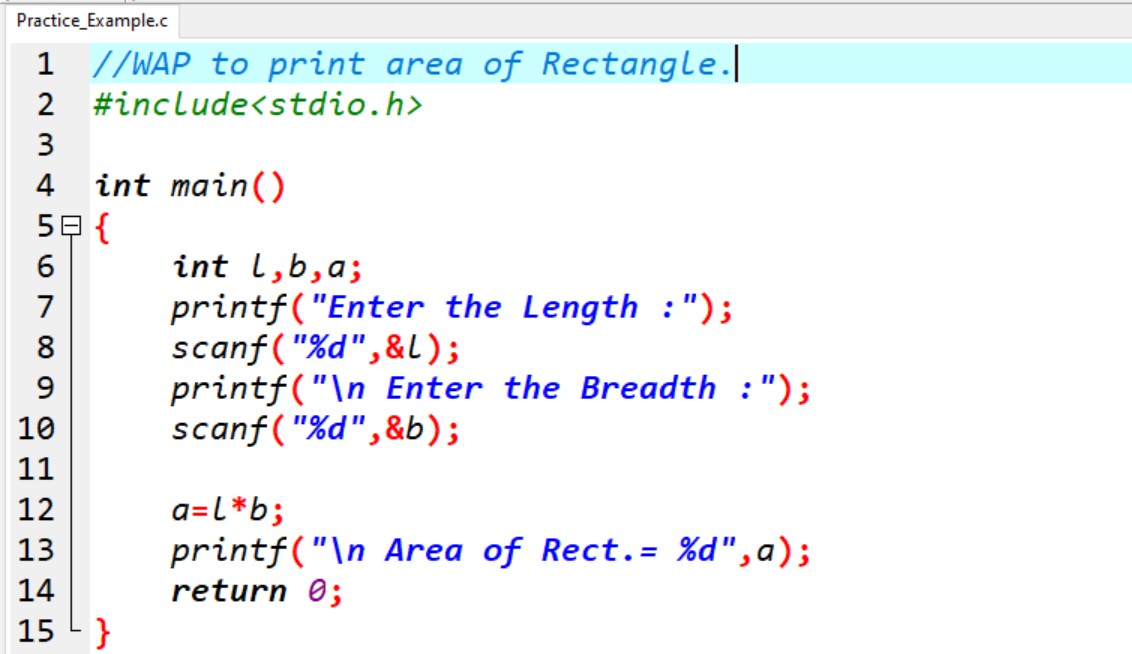
## Output:-

1. **WAP to find area of circle, rectangle and triangle.**

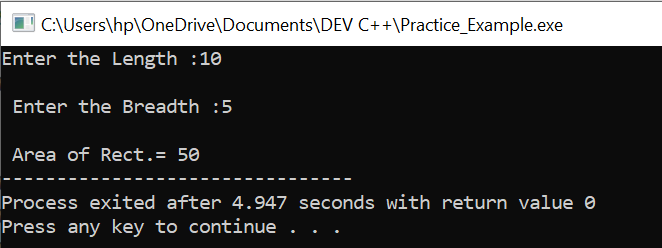
## Ans:-

* 1. **Area Of Rectangle:**

## Practical:-

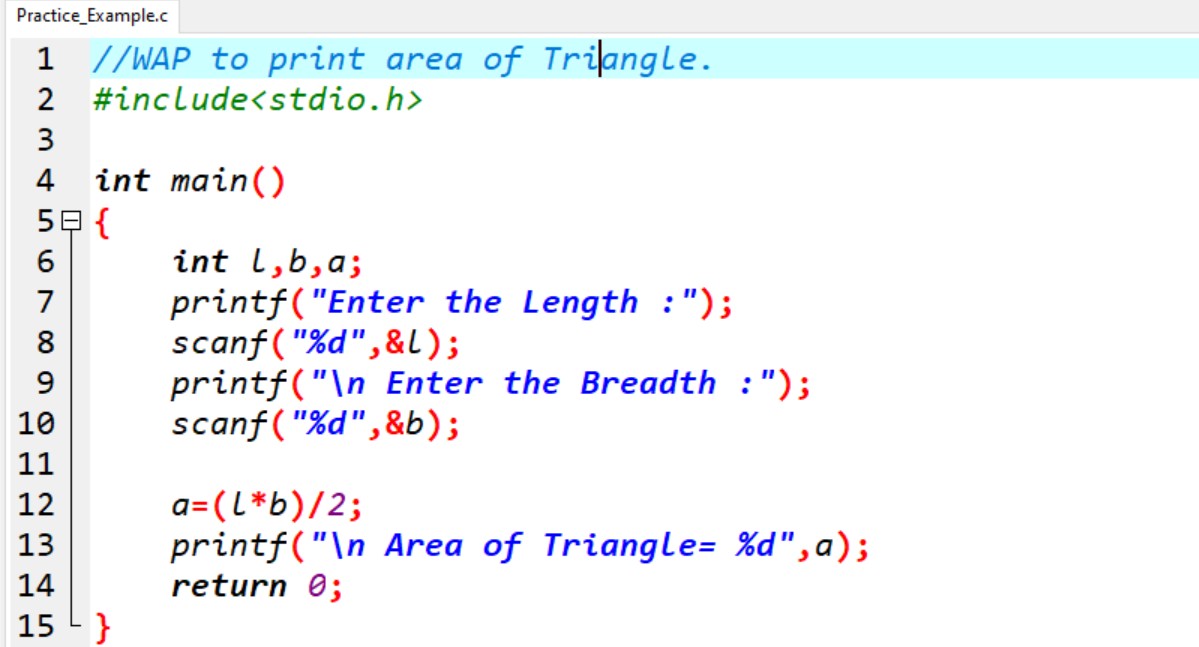


**Output:-**

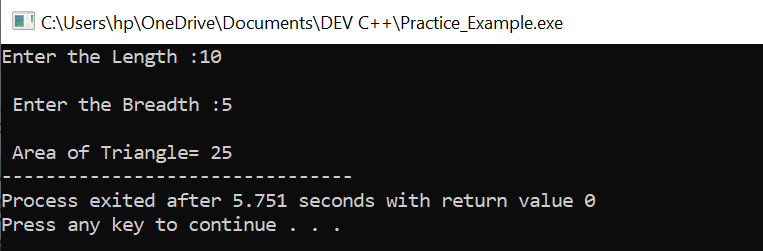


## Area Of Triangle:

**Practical:-**

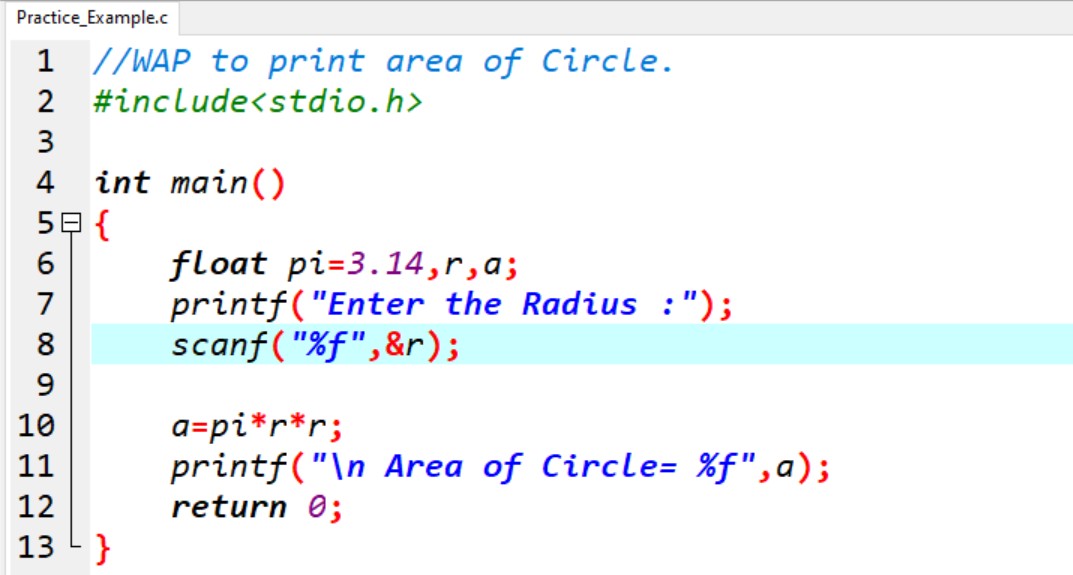


## Output:-

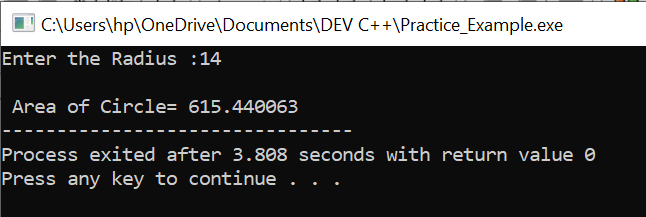


* 1. **Area Of Circle:**

## Practical:-



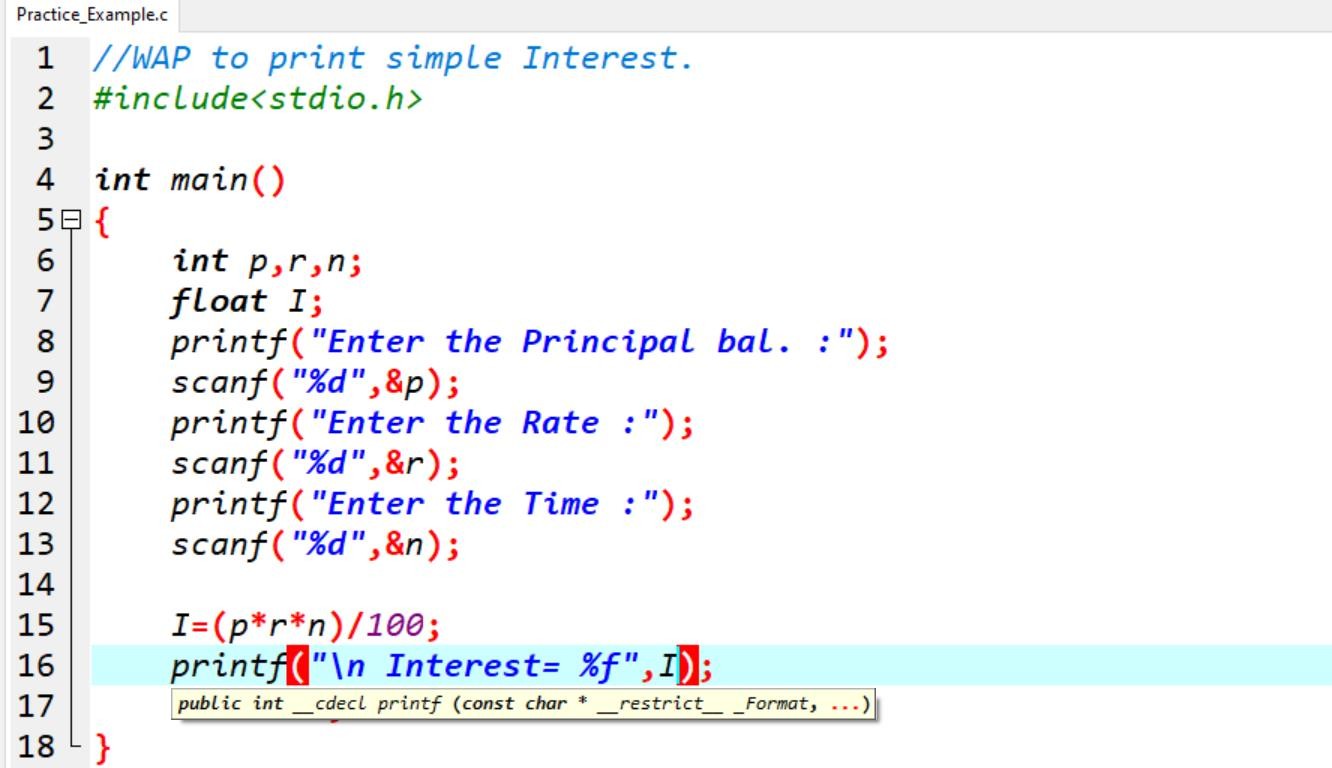
**Output:-**



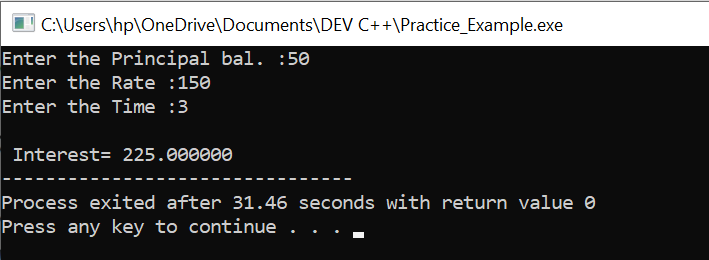
1. **WAP to find simple interest. Ans:-**

# Simple Interest:

**Practical:-**



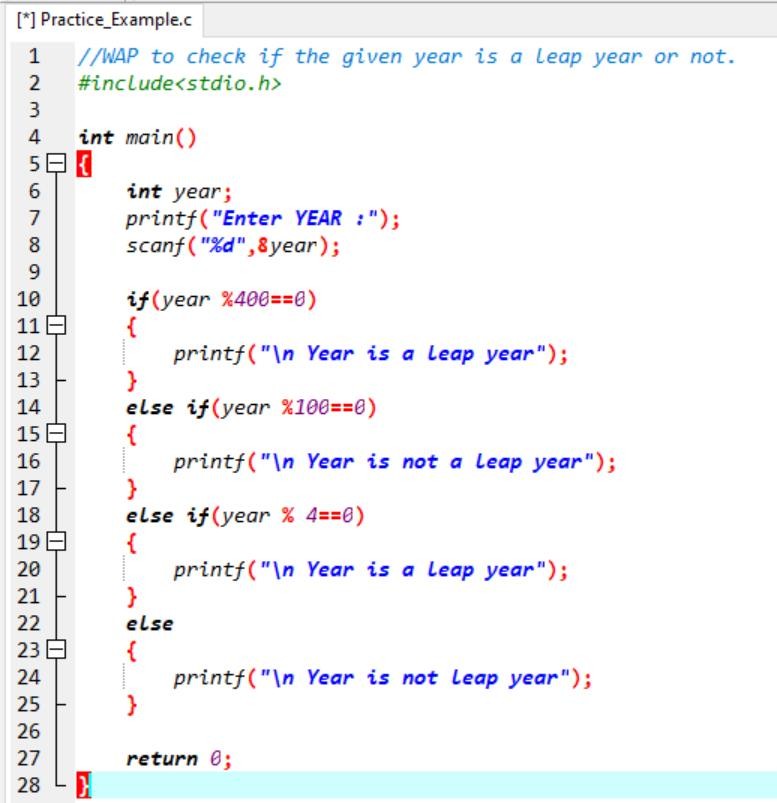
# Output:-



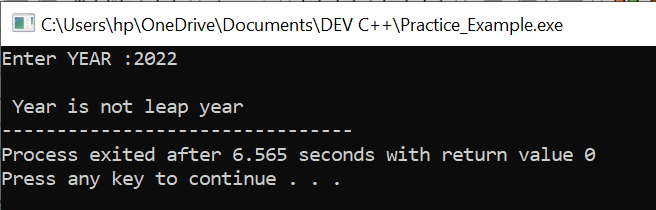
## WAP to check if the given year is a leap year or not.

**Ans:-**

## Practical:-



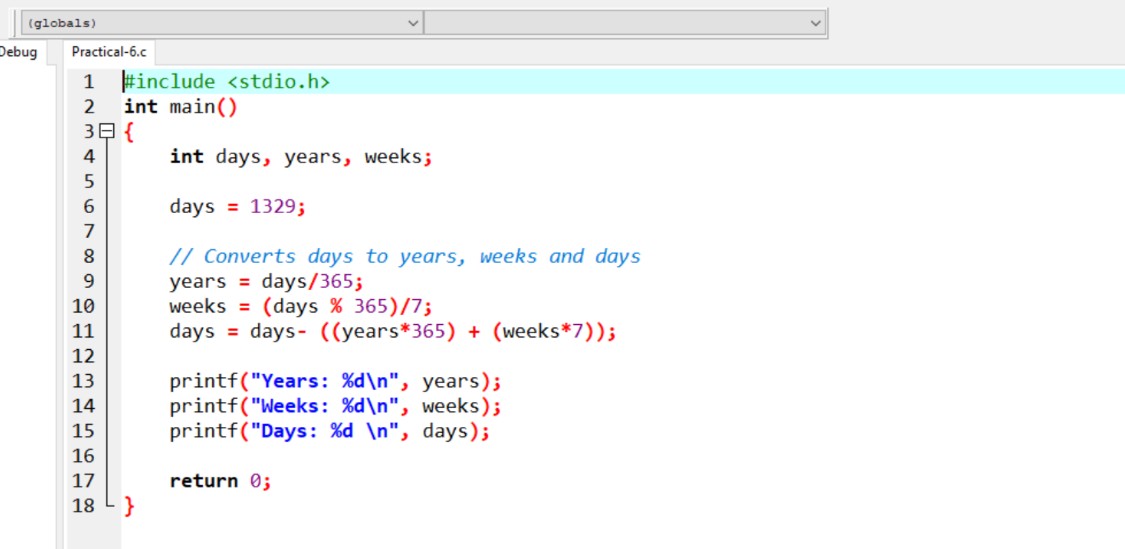
**Output:-**



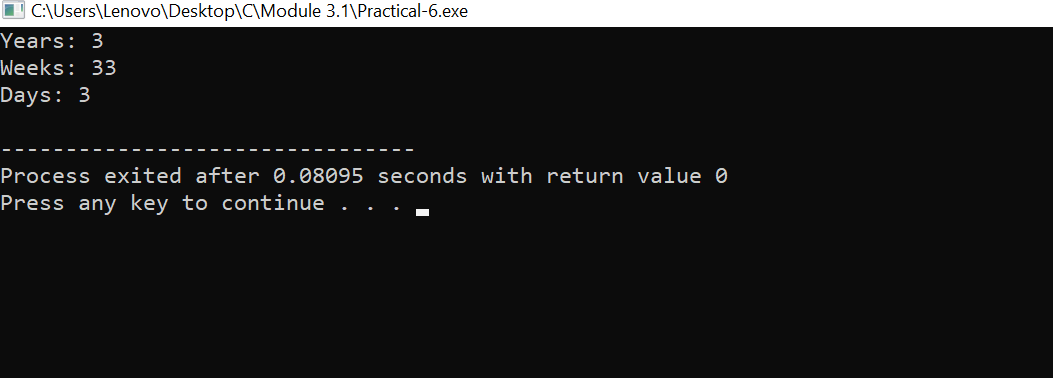
## WAP to convert years into days and days into years.

**Ans:-**

## Practical:-



**Output:-**

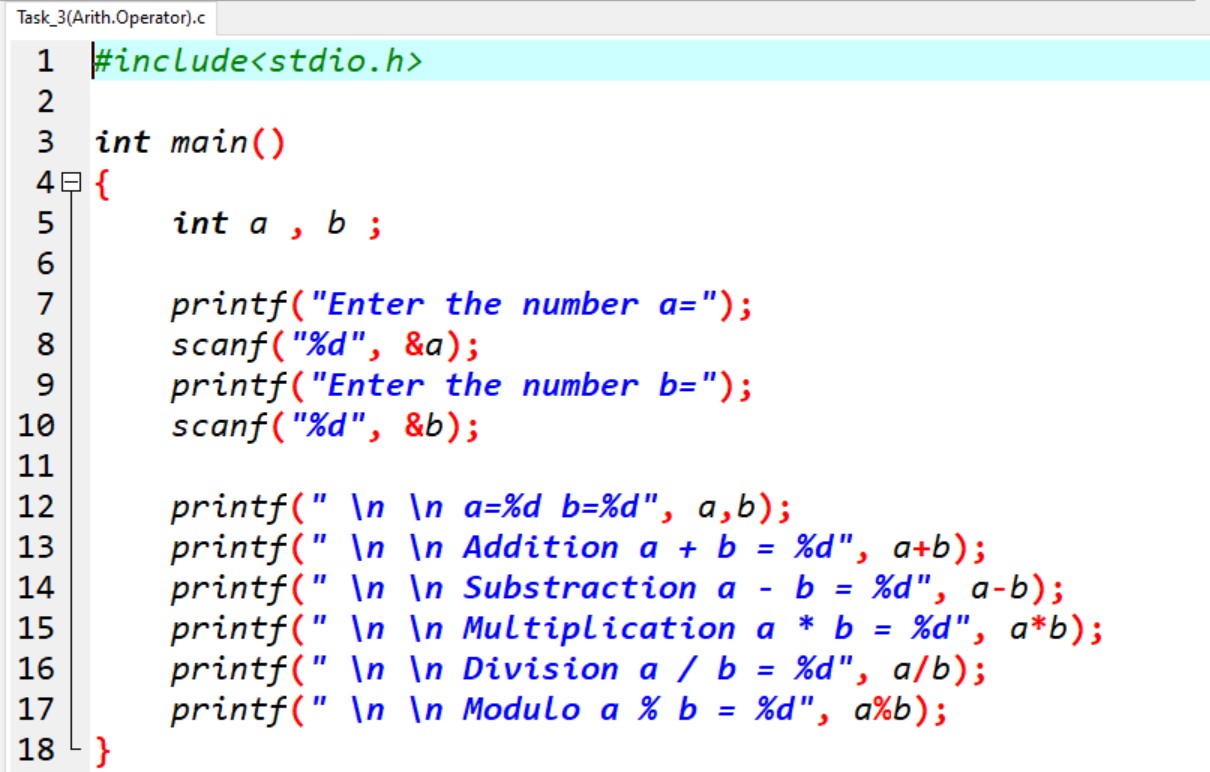


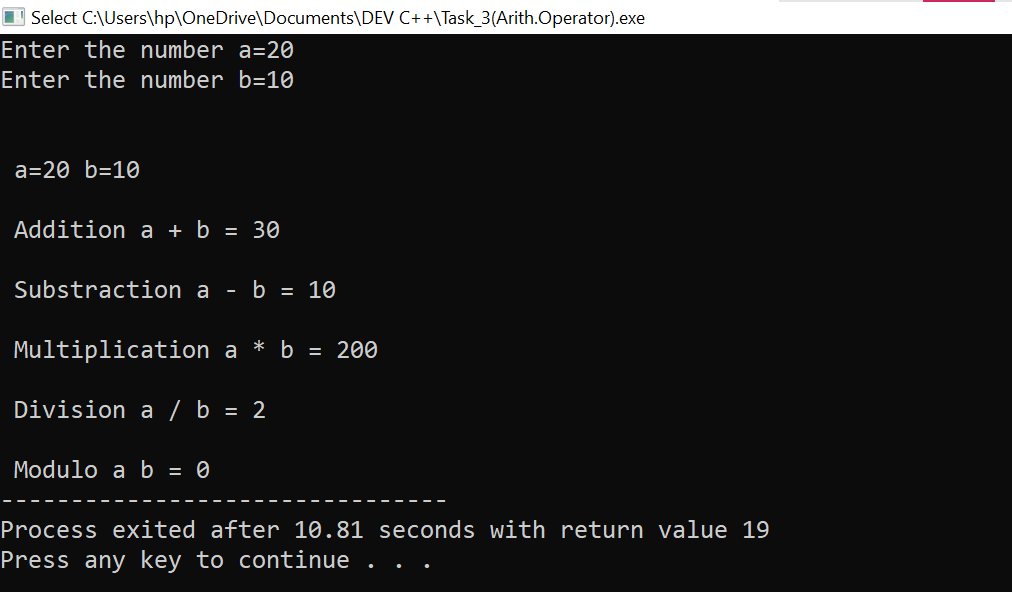
**MODULE: 2.2 (C Language Programing with C)**

1. **WAP to make simple calculator (operation include Addition, Subtraction, Multiplication, Division, modulo).**

**Ans:-**

## Practical:-

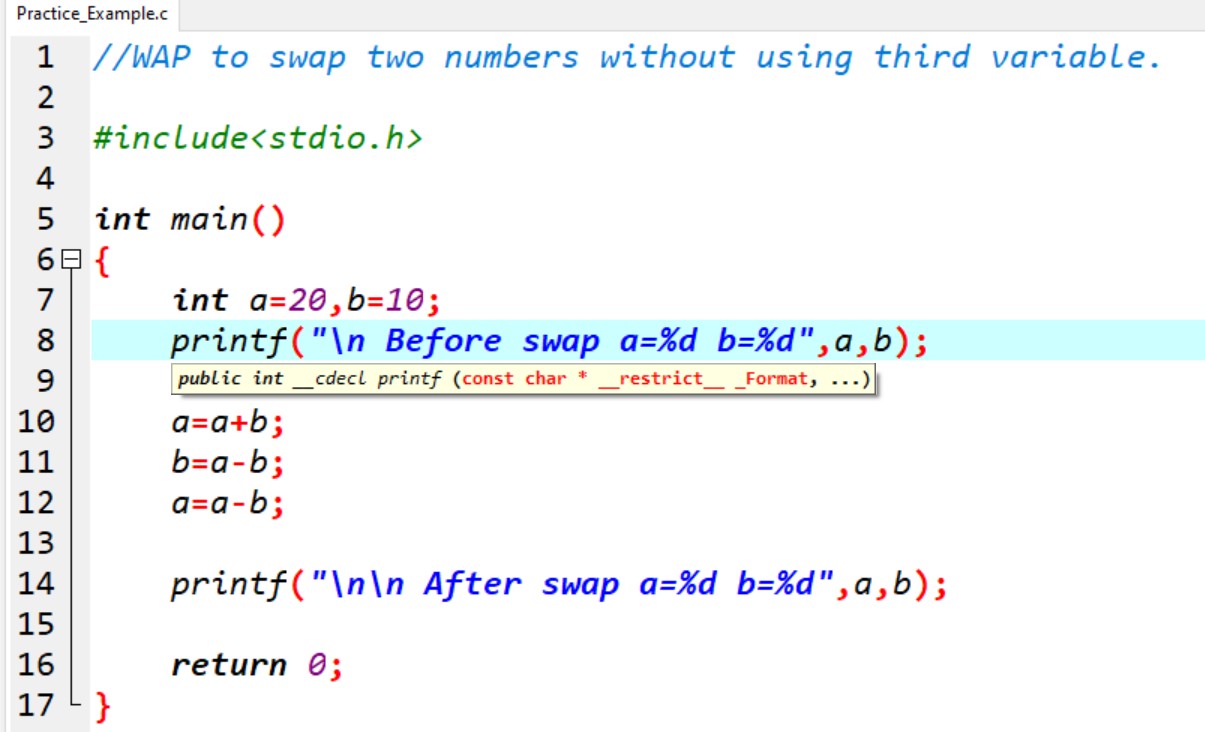


**Output:-**

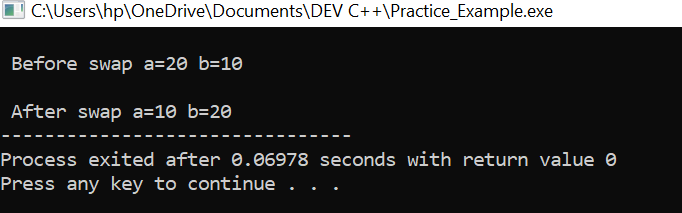
1. **WAP to swap two numbers without using third variable.**

**Ans:-**

**Practical:-**



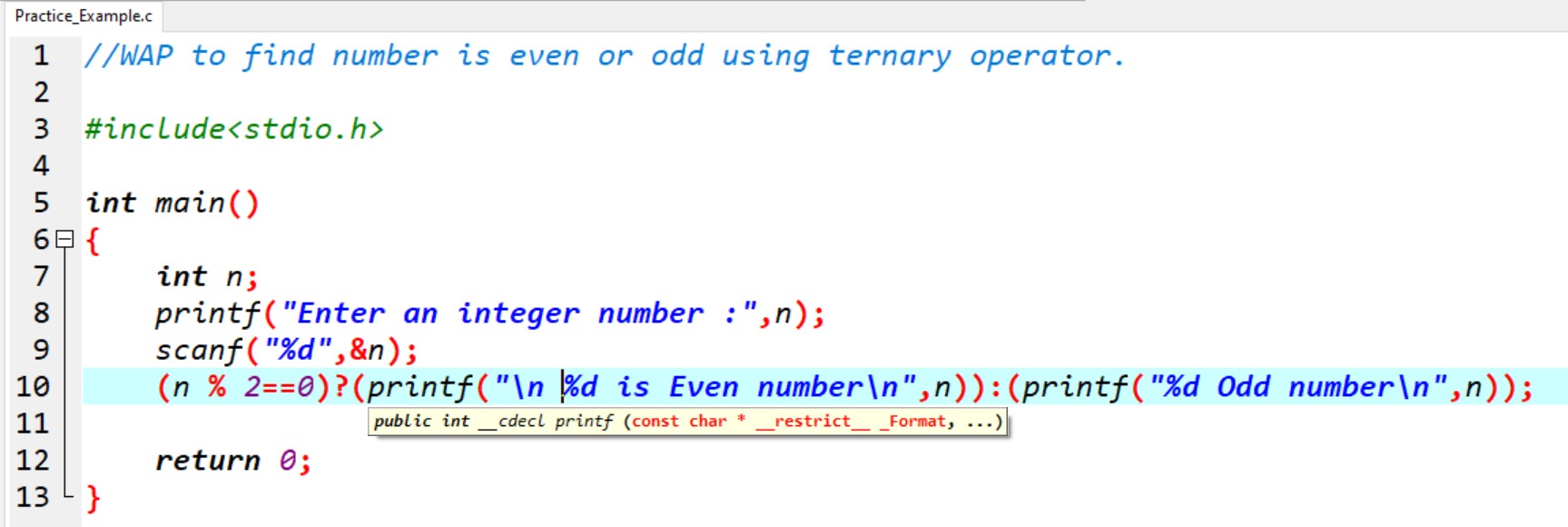
**Output:-**



1. **WAP to find number is even or odd using ternary operator.**

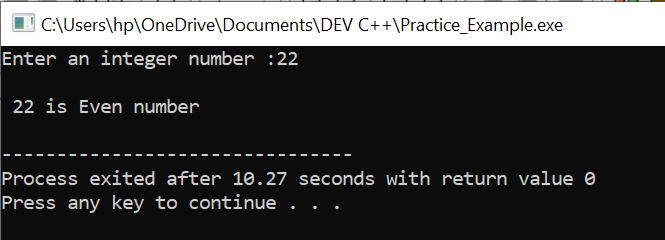
## Ans:-

**Practical:-**

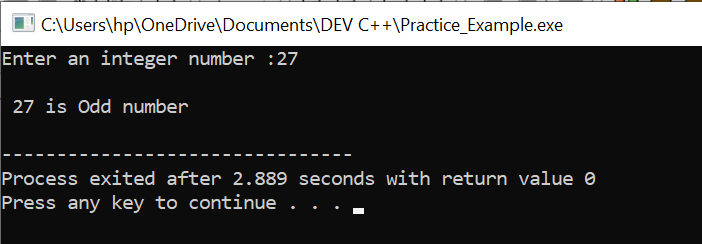


**Output:-**

[EVEN]:-

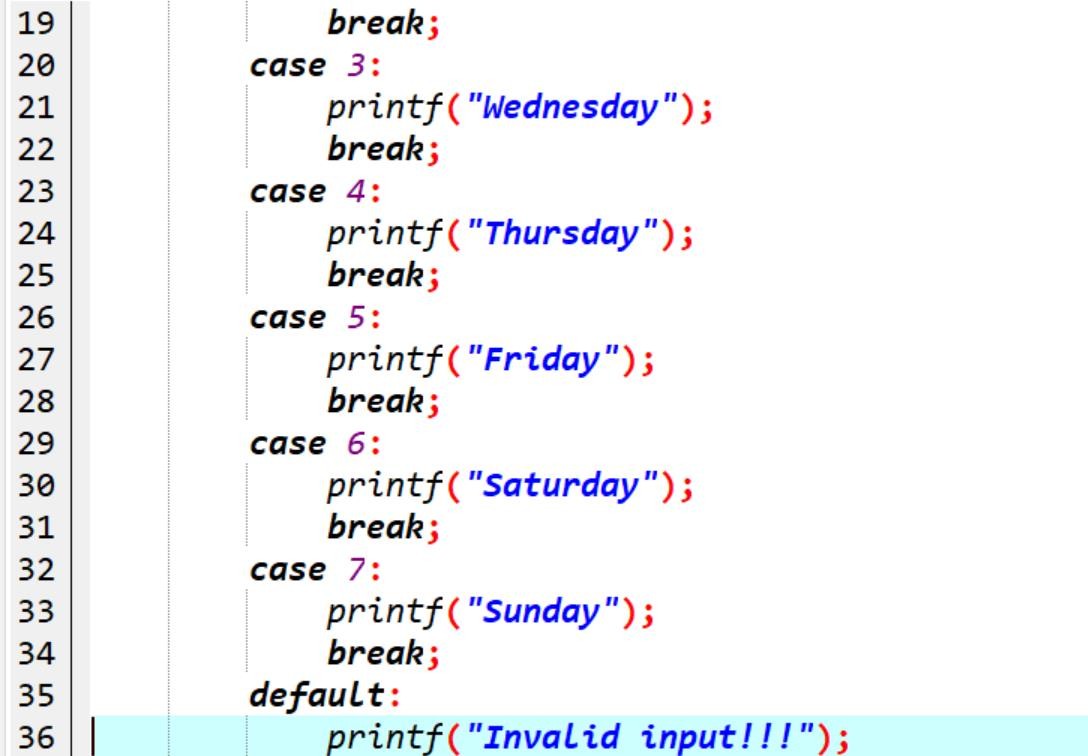


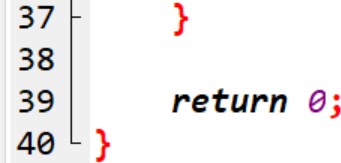
[Odd]:-



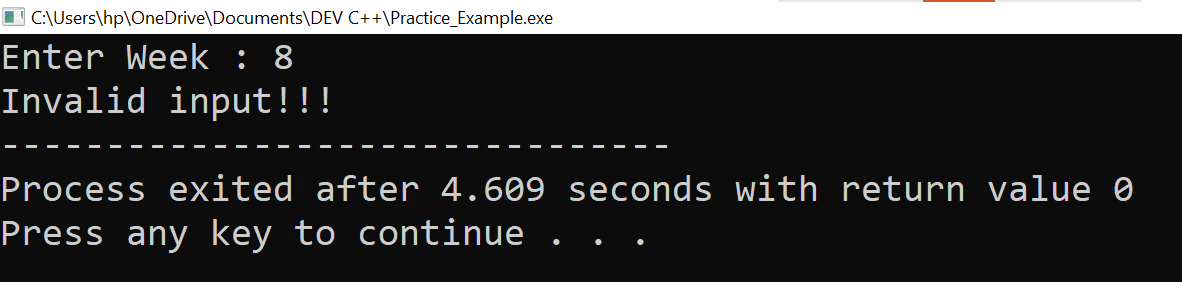
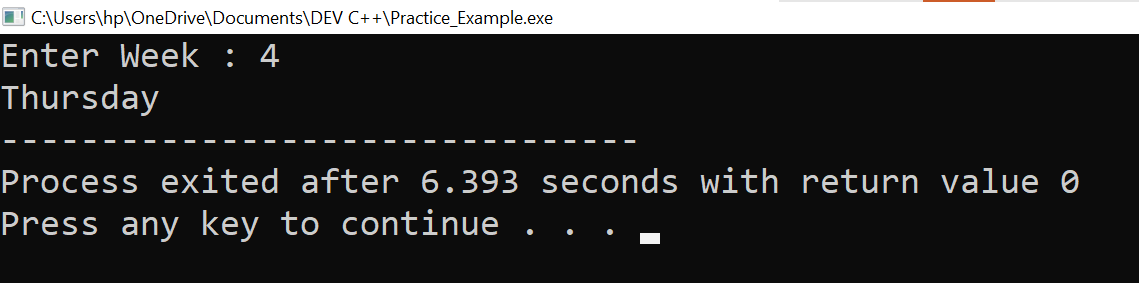
1. **WAP to show:-**
   1. **Monday to Sunday using switch case.**
   2. **Vowel or Consonant using switch case. Ans:-**
2. **Monday to Sunday using switch case**

## Practical:-



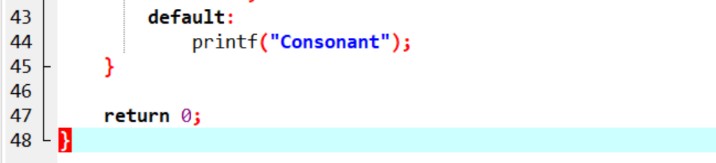
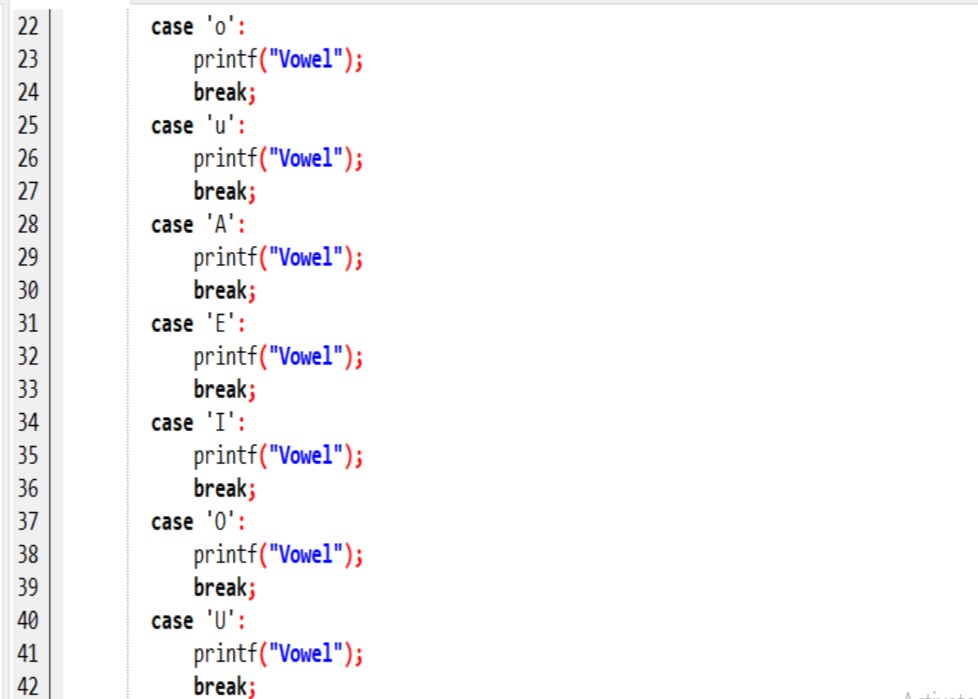
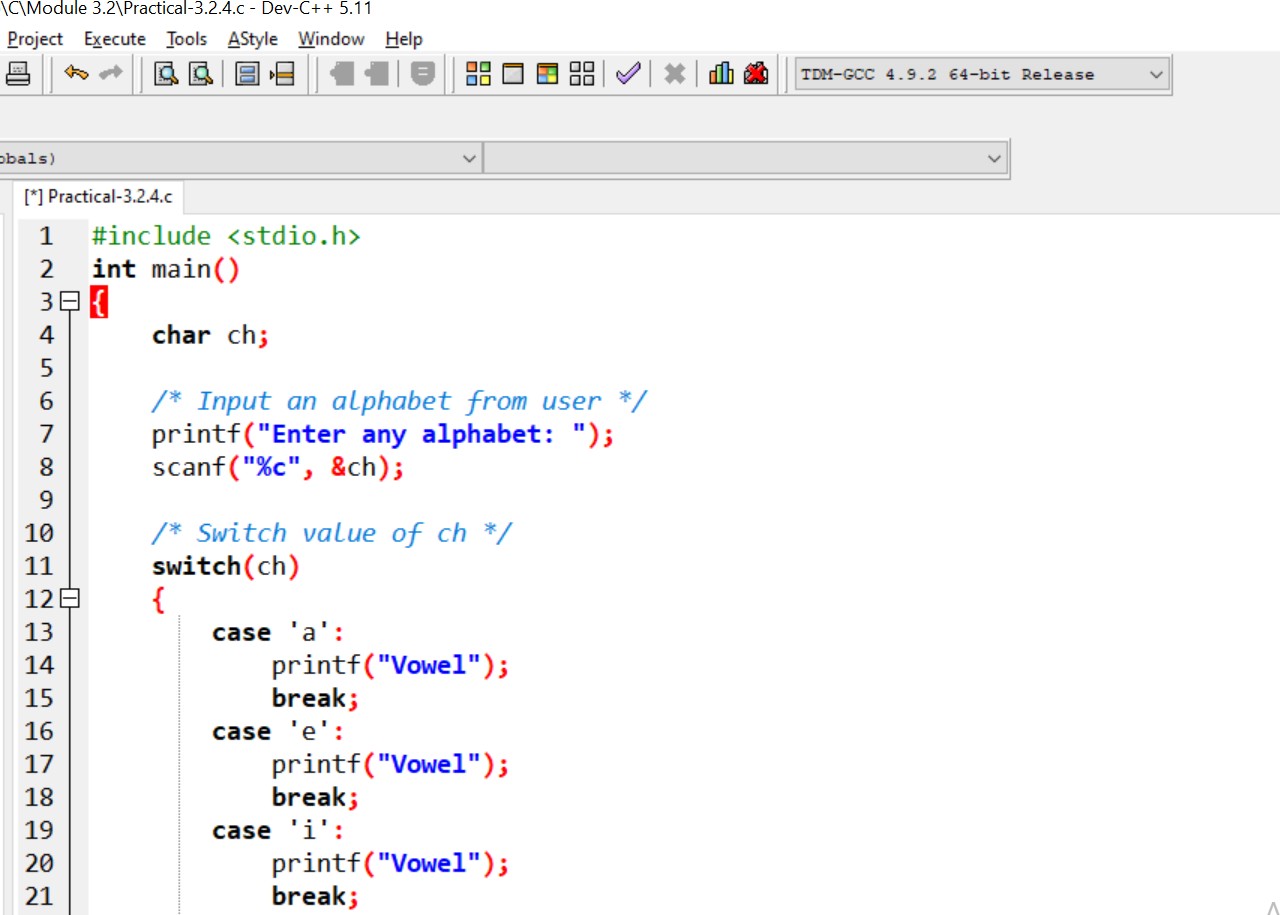


**Output:-**

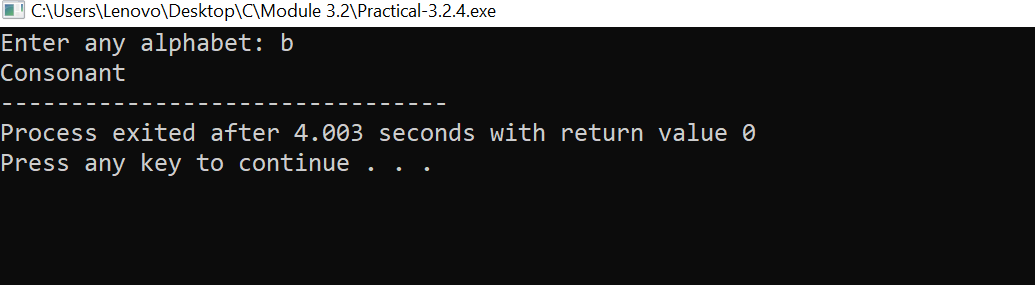
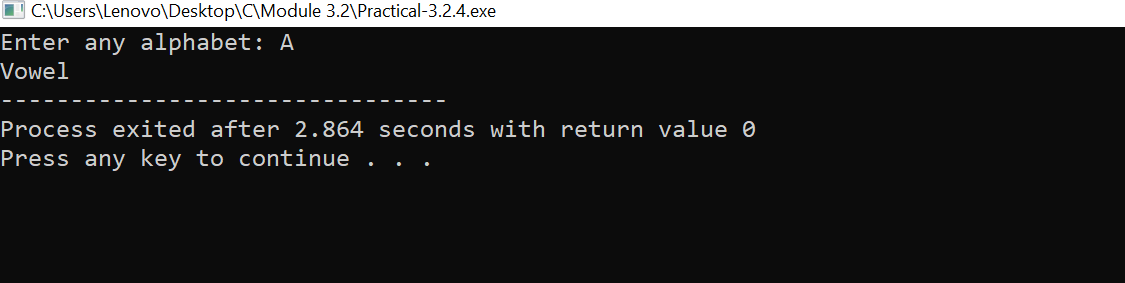


1. **Vowel or Consonant using switch case.**

## Practical:-

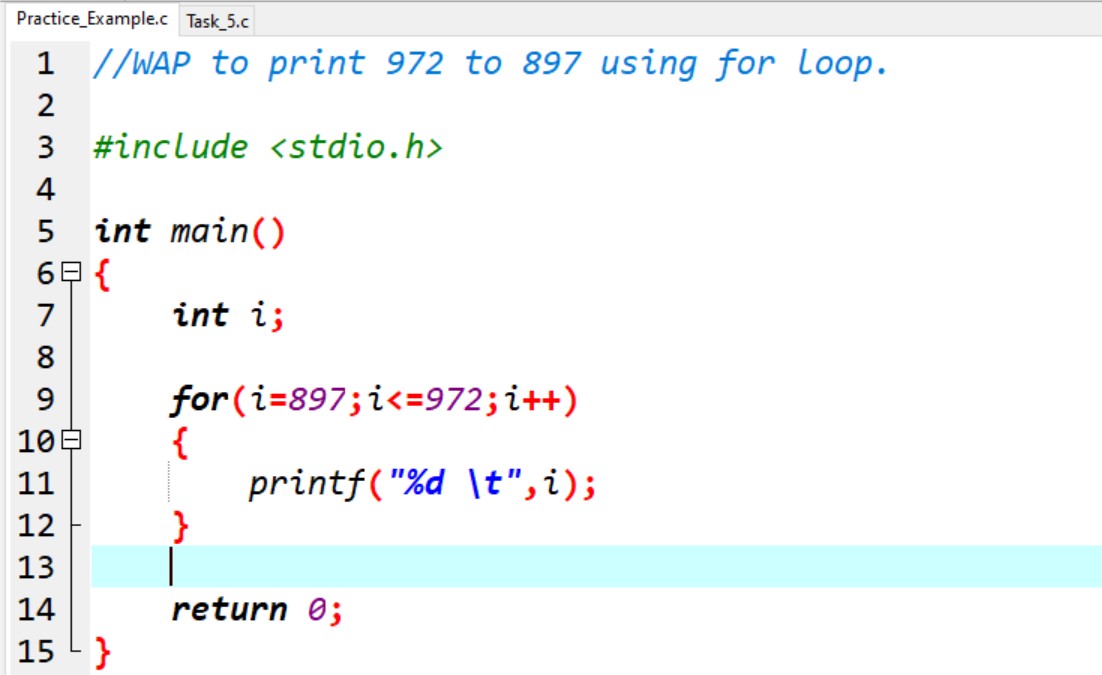


**Output:-**

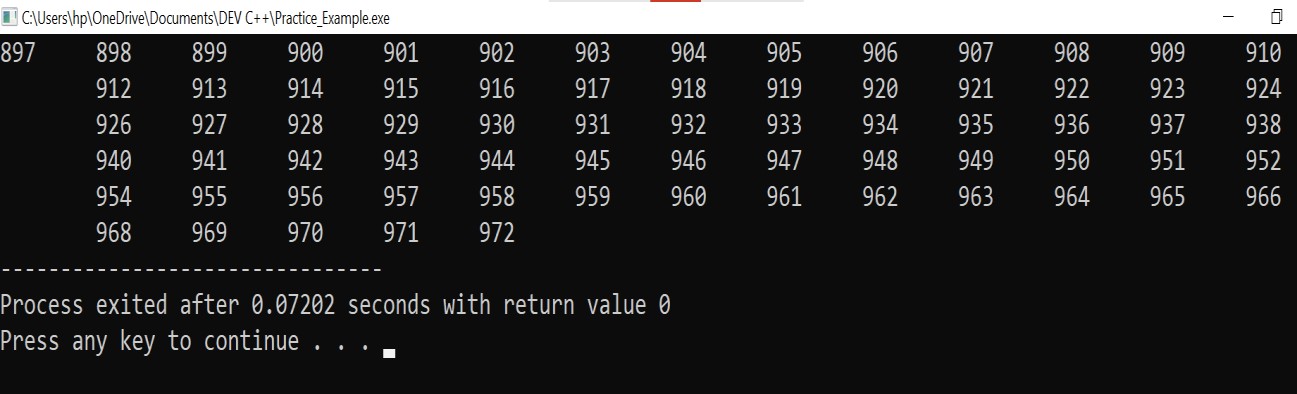


1. **Looping programs:**
   1. **WAP to print 972 to 897 using for loop.**
   2. **WAP to take 10 no. Input from user and find out..**
      1. **How many Even numbers are there.**
      2. **How many odd numbers are there.**
      3. **Sum of even numbers.**
      4. **Sum of odd numbers.**
      5. **WAP to print table up to given numbers. Ans:-**
2. **WAP to print 972 to 897 using for loop.**

## Practical:-

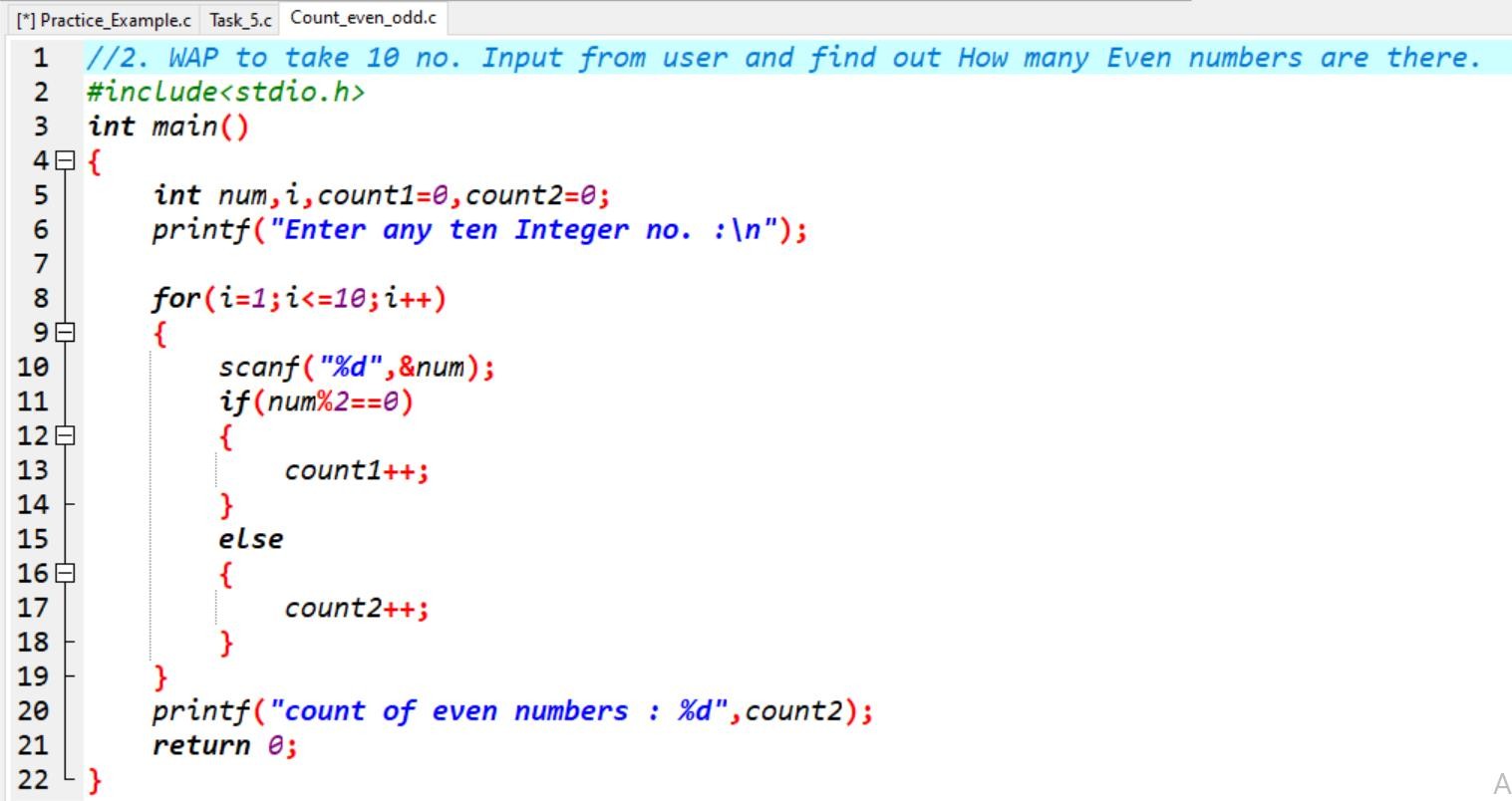


**Output:-**

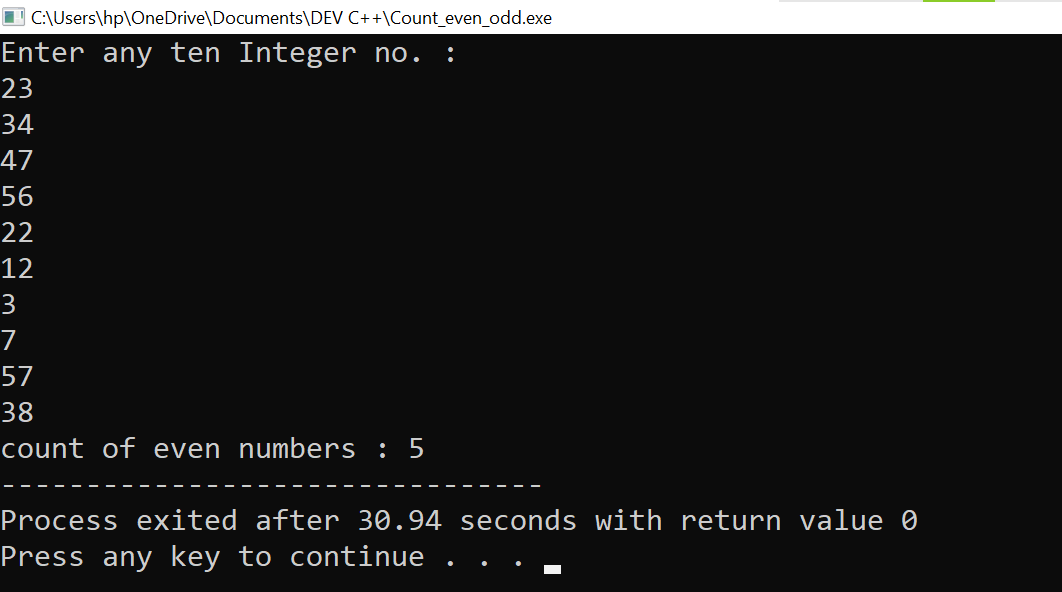


1. **WAP to take 10 no. Input from user and find out..**
   1. **How many Even numbers are there.**

**Practical:-**

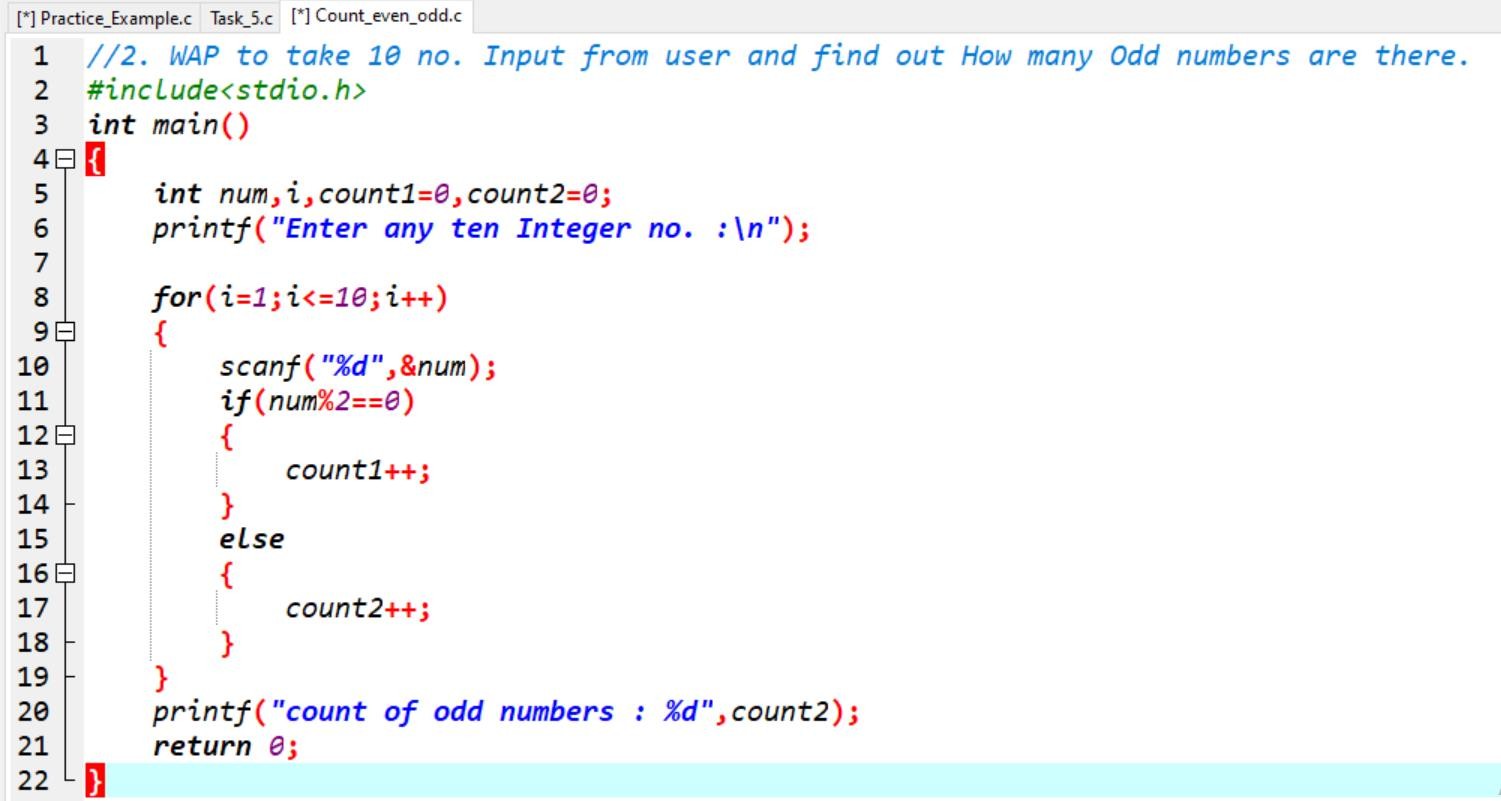


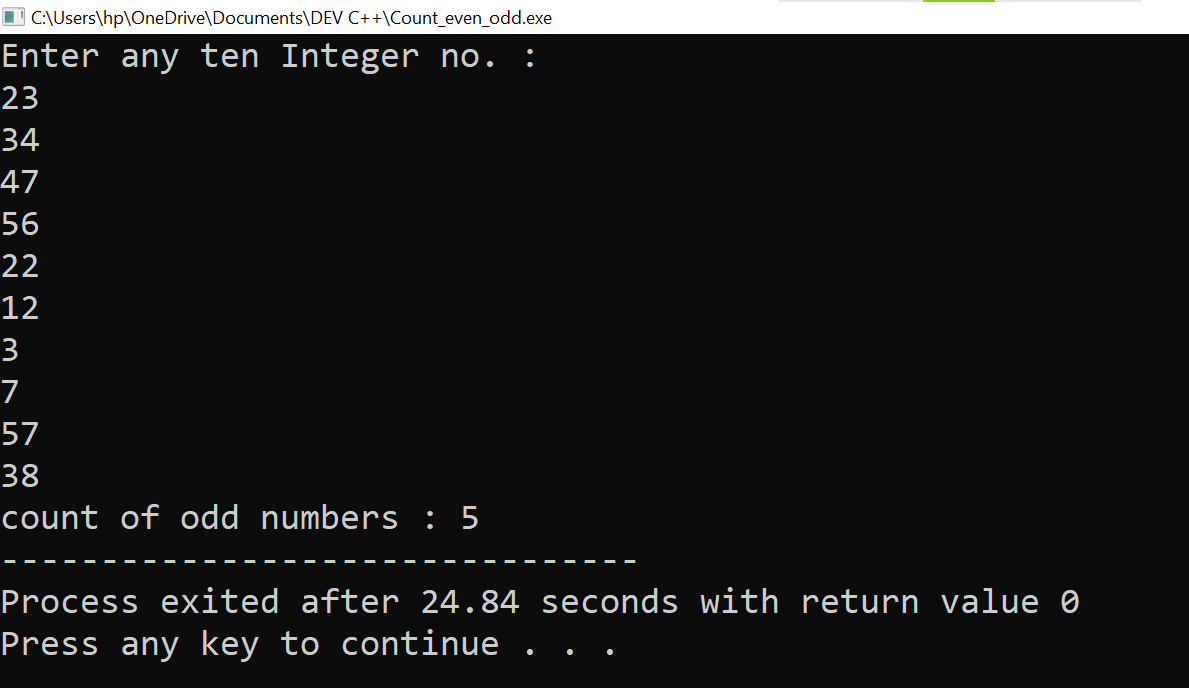
**Output:-**



* 1. **How many odd numbers are there.**

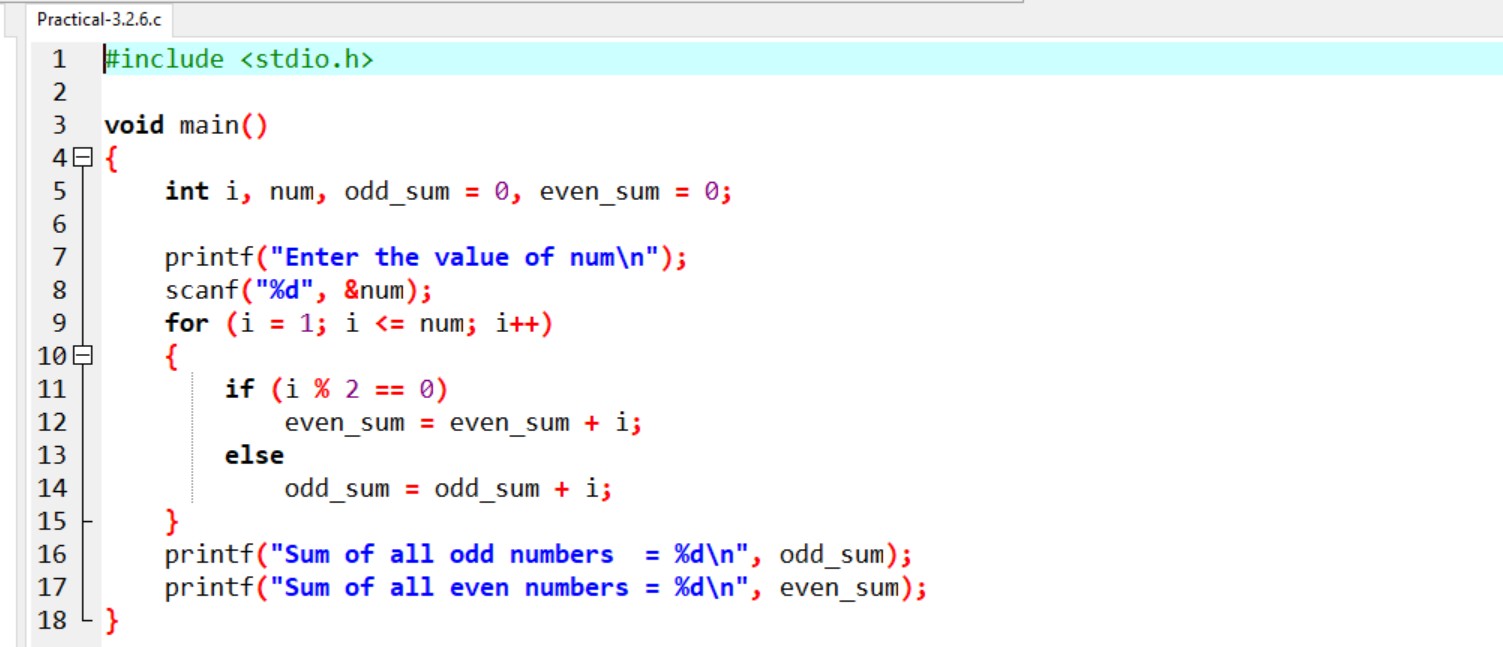
**Practical:-**



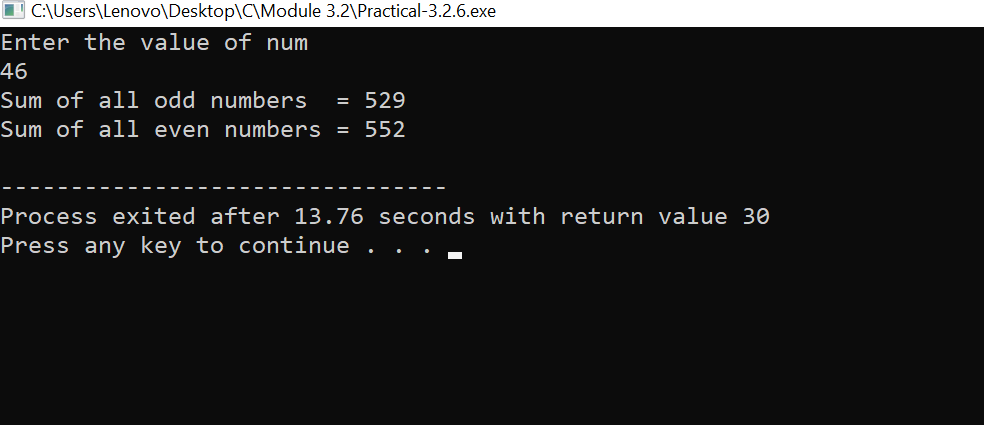
**Output:**-

* 1. **Sum of even numbers.**
  2. **Sum of odd numbers.**

**Practical:-**

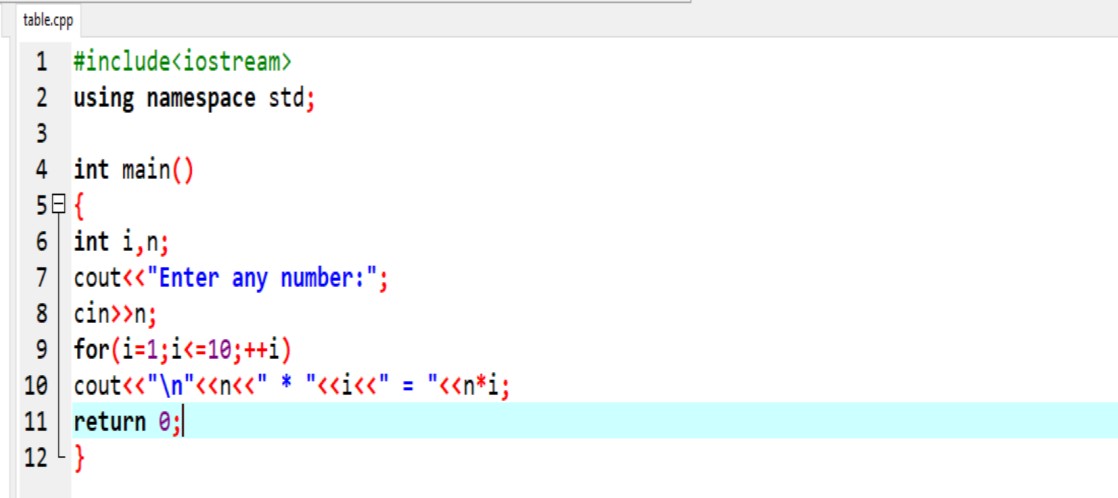


**Output:-**

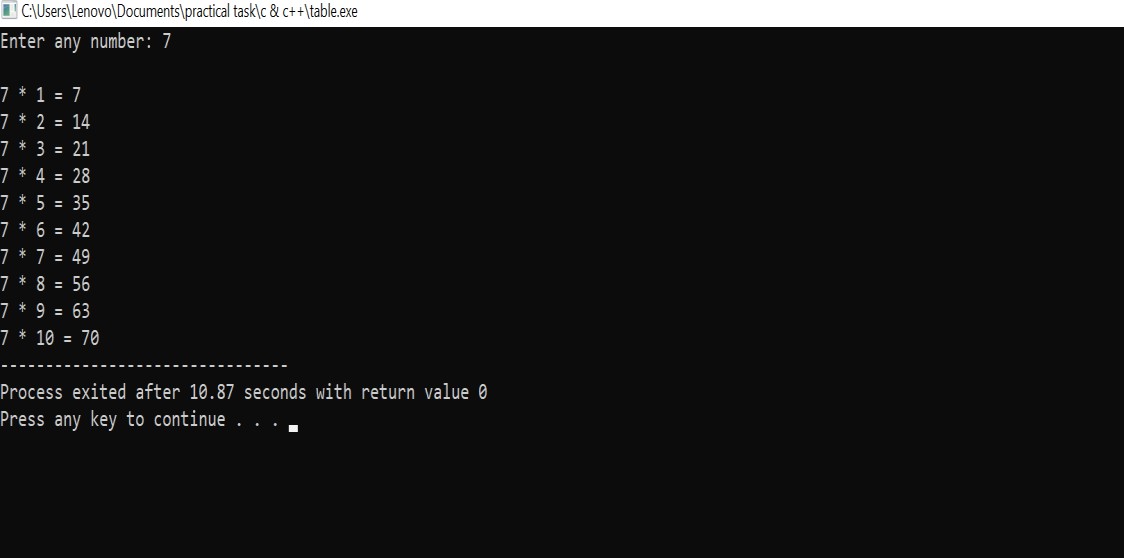


1. **WAP to print table up to given numbers.**

**Practical:-**

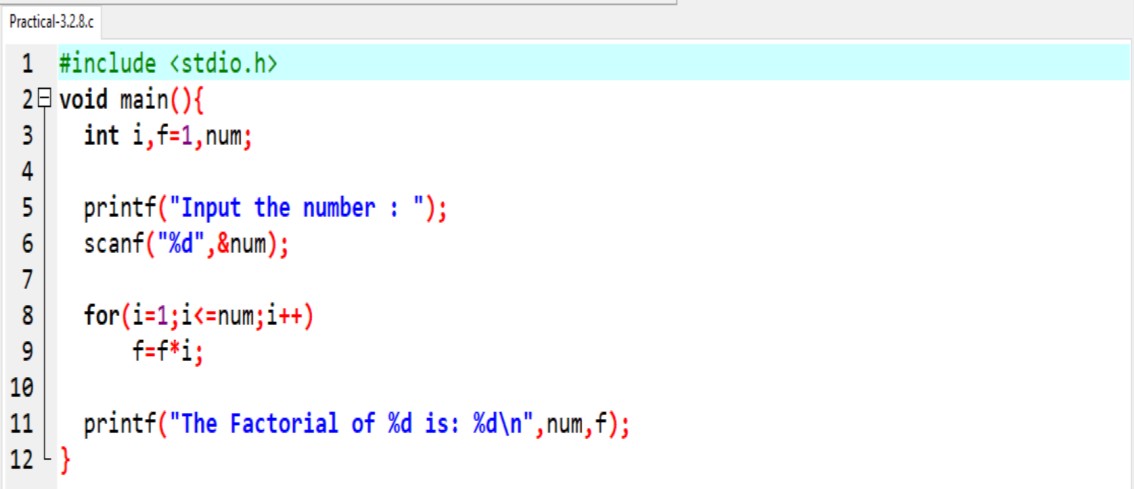


**Output:-**

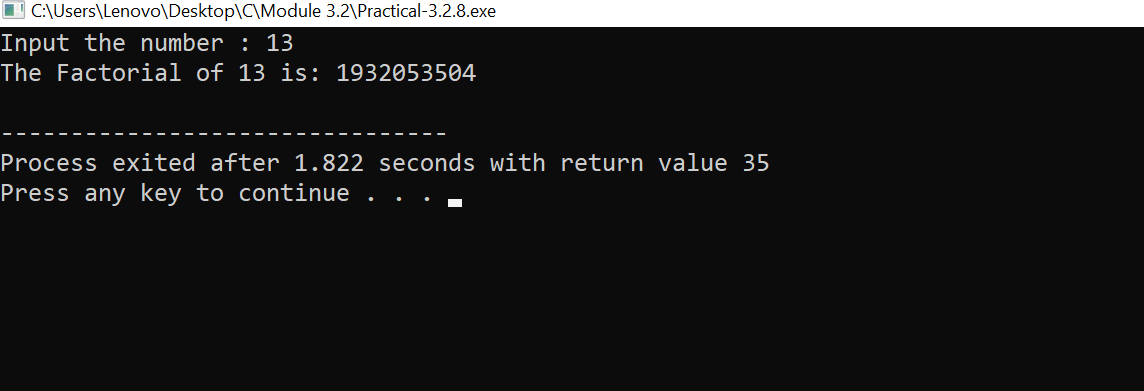


1. **WAP to print factorial of given number. Ans:-**

**Practical:**-

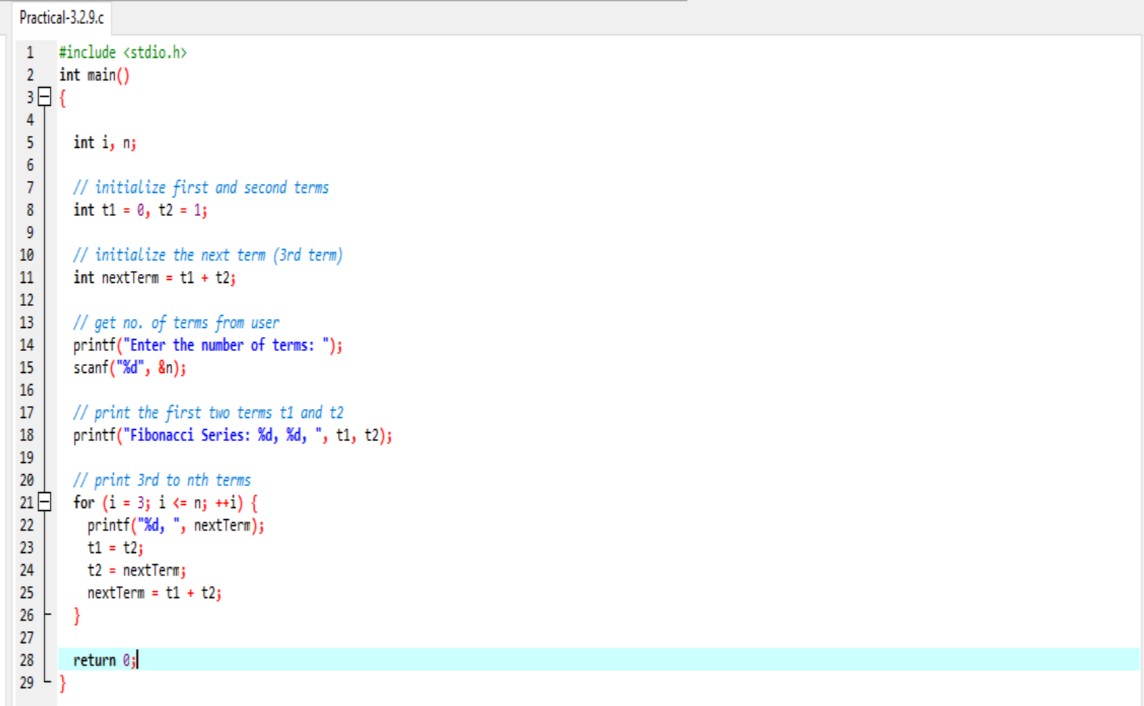


**Output:-**

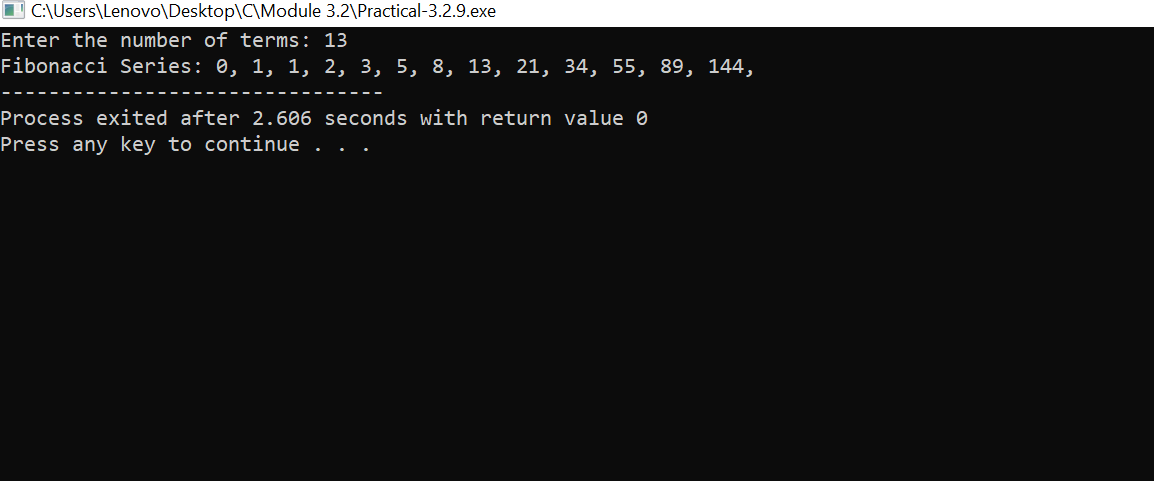


1. **WAP to print Fibonacci series up to given numbers. Ans:-**

## Practical:-



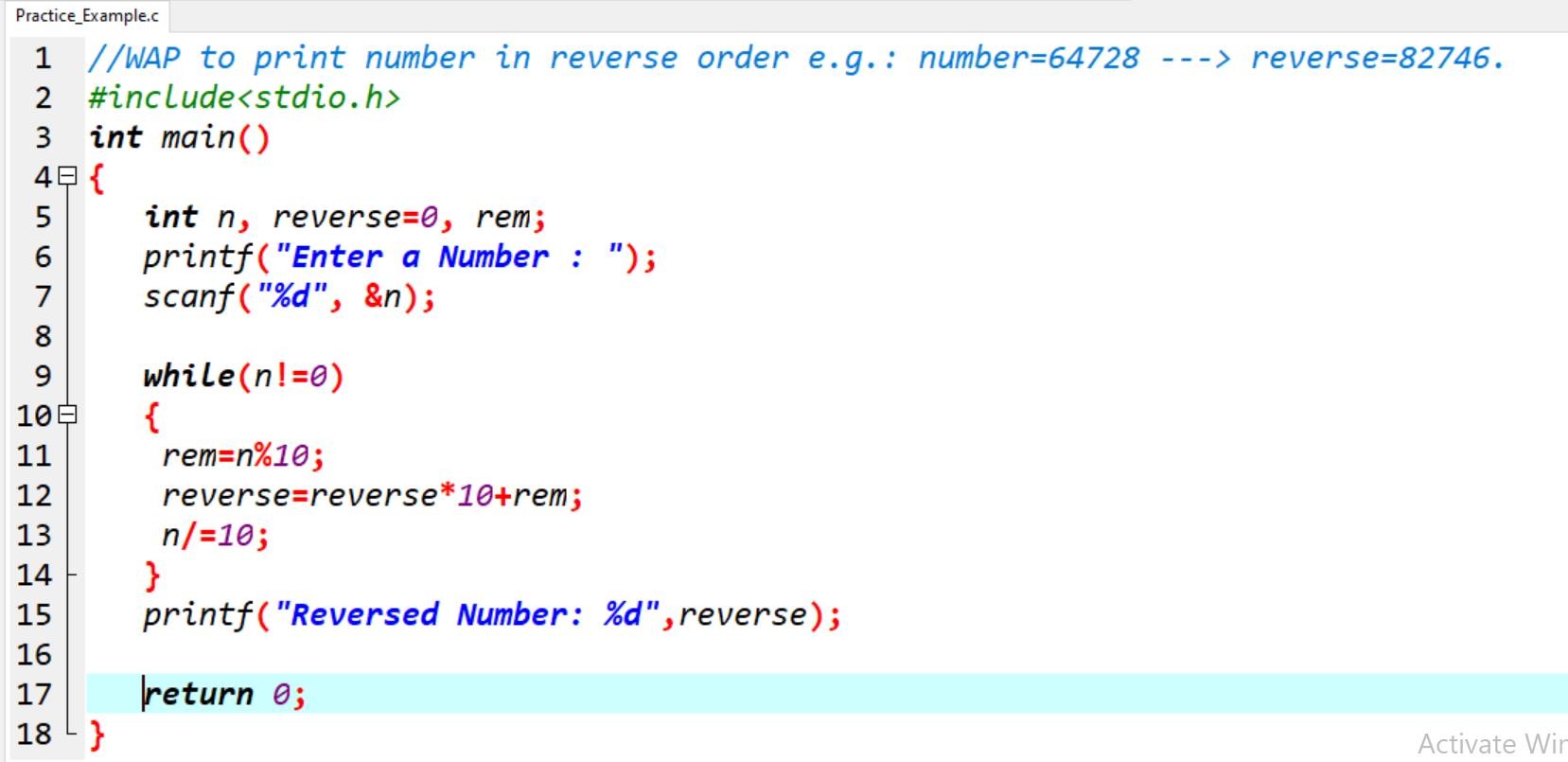
**Output:-**



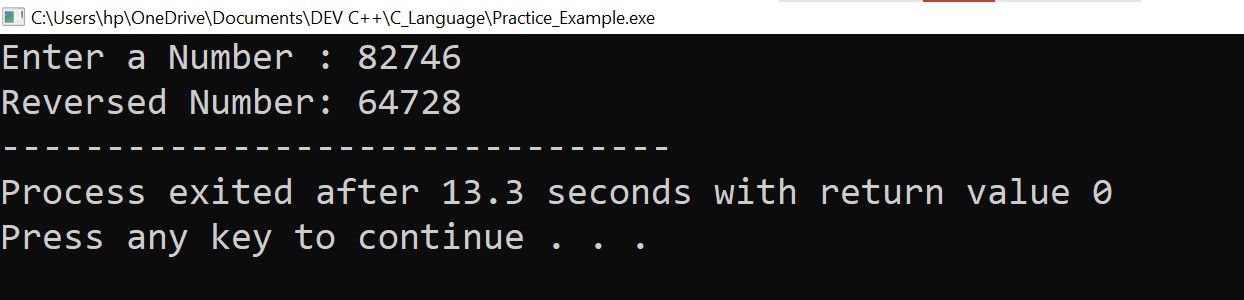
1. **WAP to print number in reverse order e.g.: number = 64728 ---> reverse = 82746.**

**Ans:-**

**Practical:-**



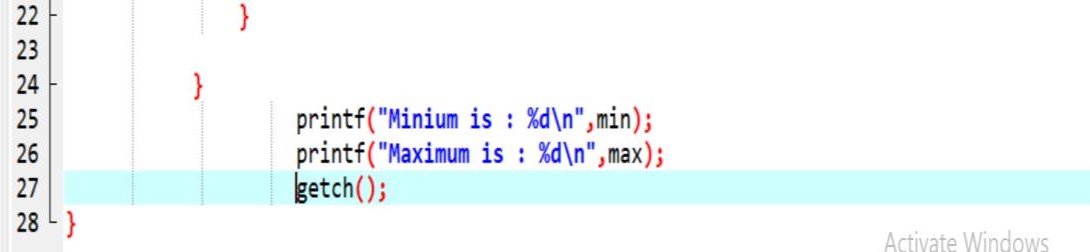
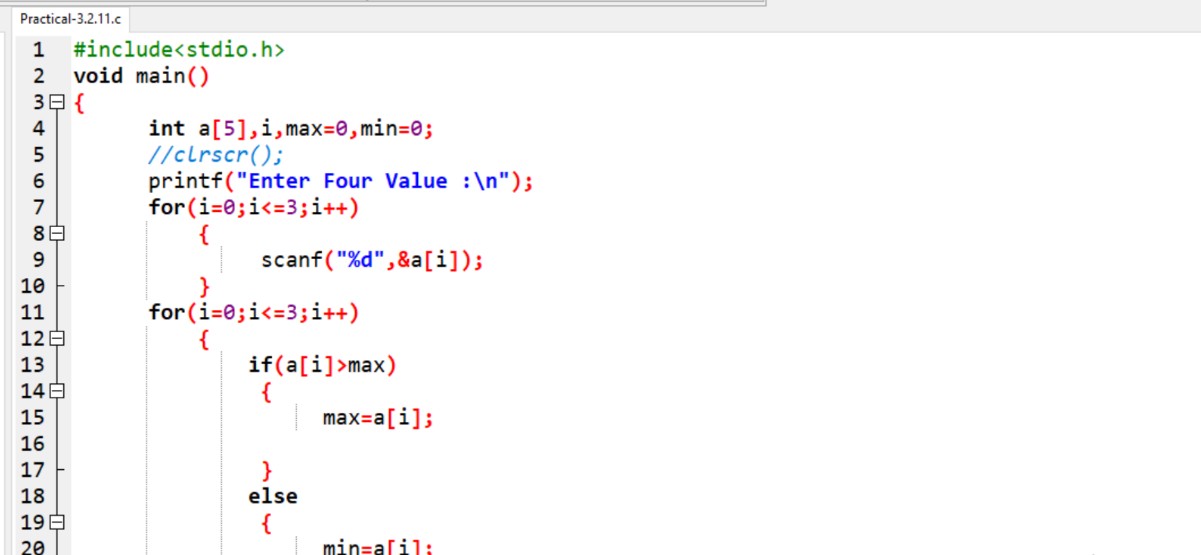
**Output:-**



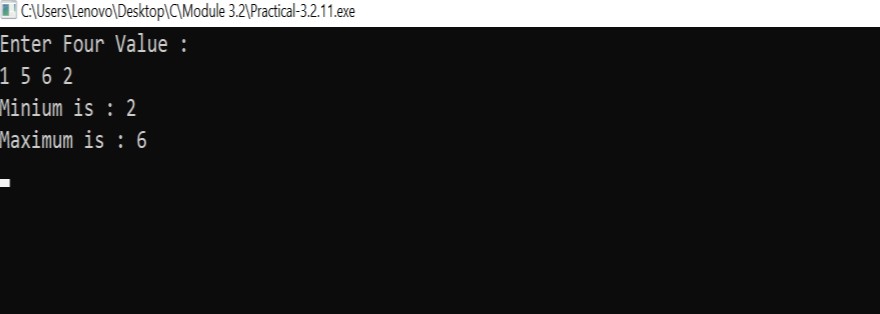
1. **Write a program to find out the max from given number (E.g., No: -1562 Max number is 6).**

**Ans:-**

## Practical:-



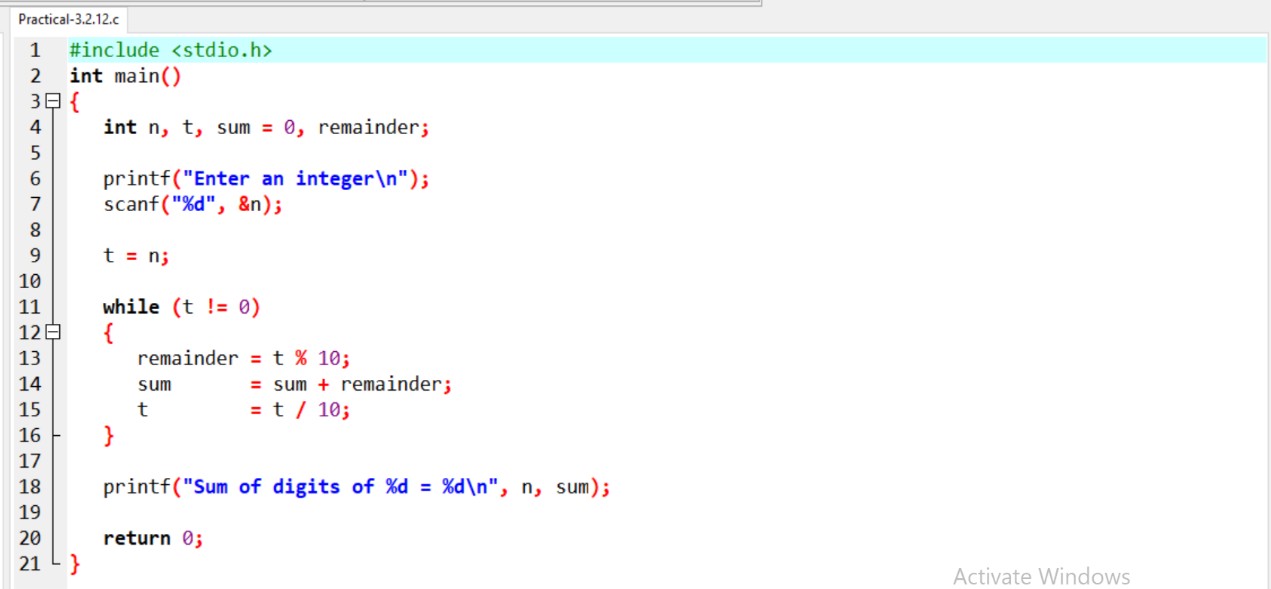
**Output:-**



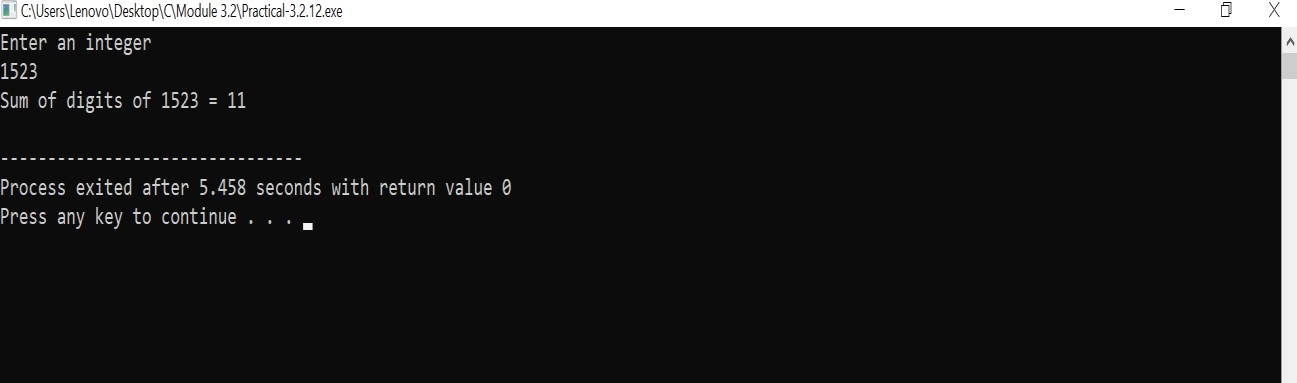
1. **Write a program make a summation of given number (E.g., 1523 Ans: -11).**

**Ans:-**

## Practical:-



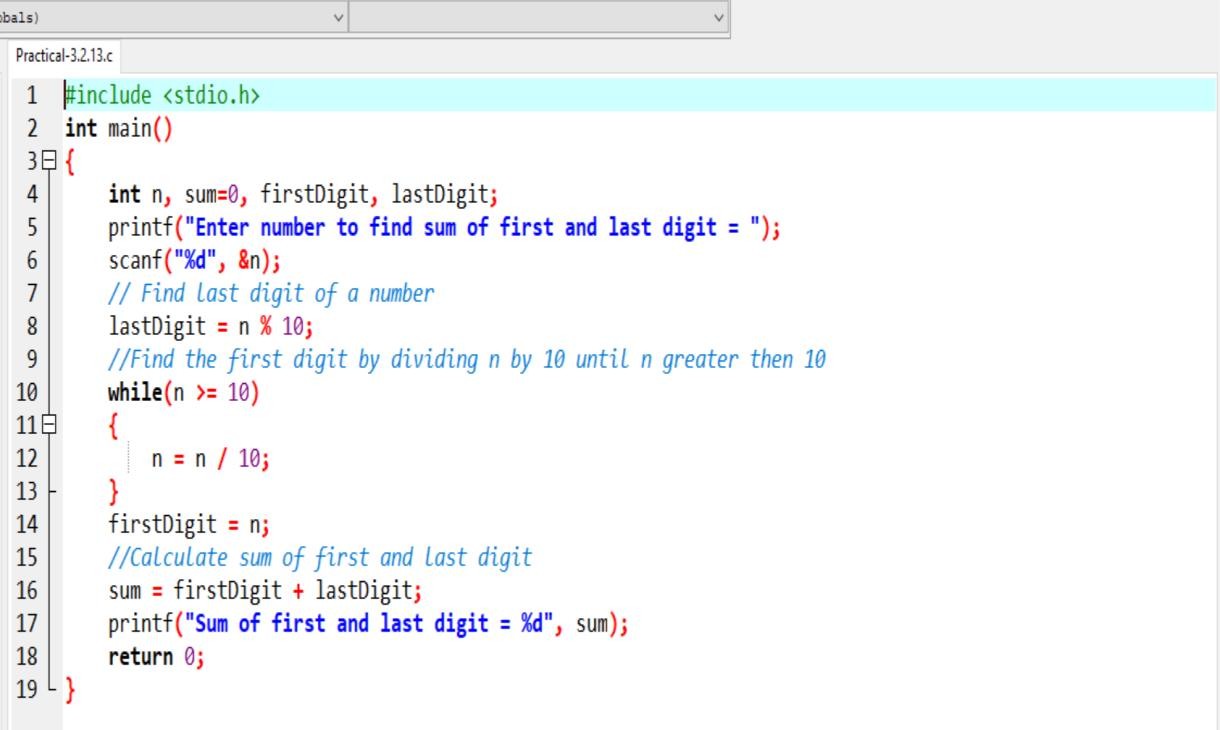
**Output:**-



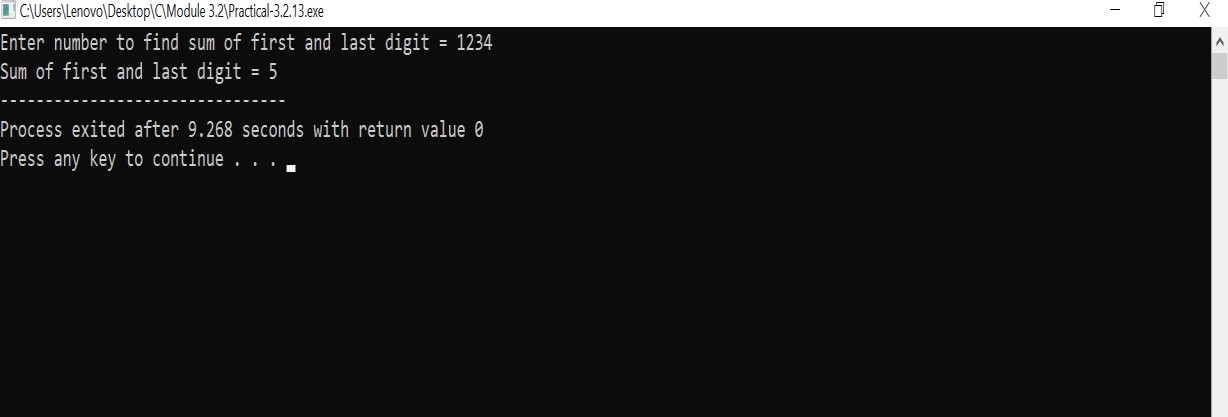
1. **Write a program you have to make a summation of first and last Digit. (E.g., 1234 Ans: -5).**

**Ans:-**

## Practical:-

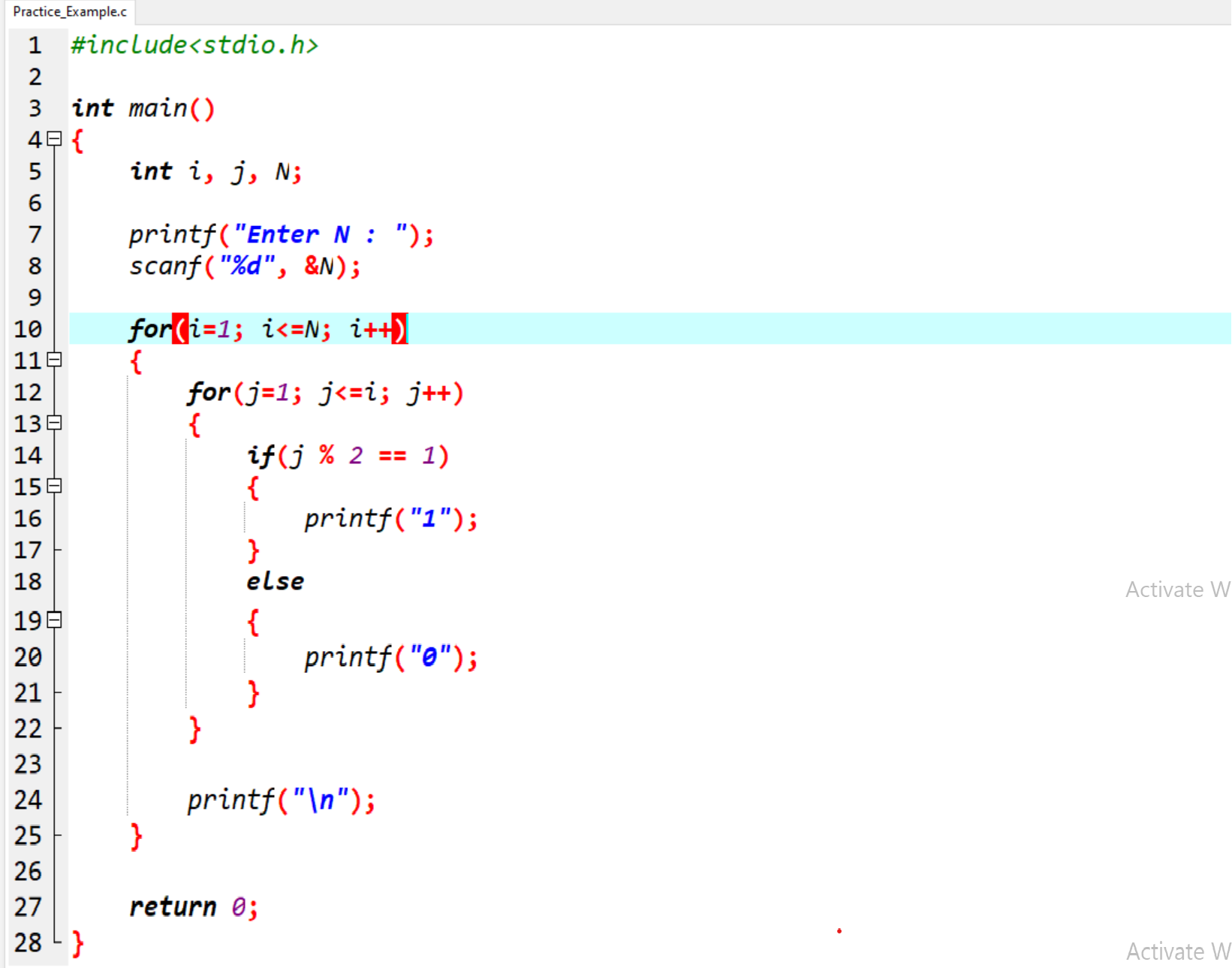


**Output:-**

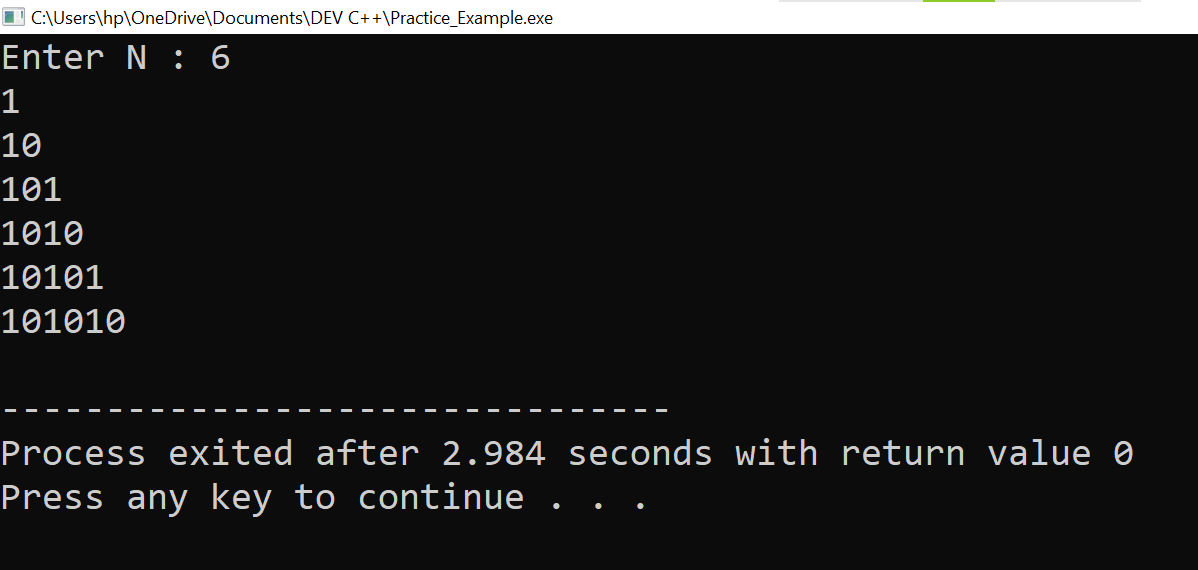


**PATTERNS** 1.

## Practical:-

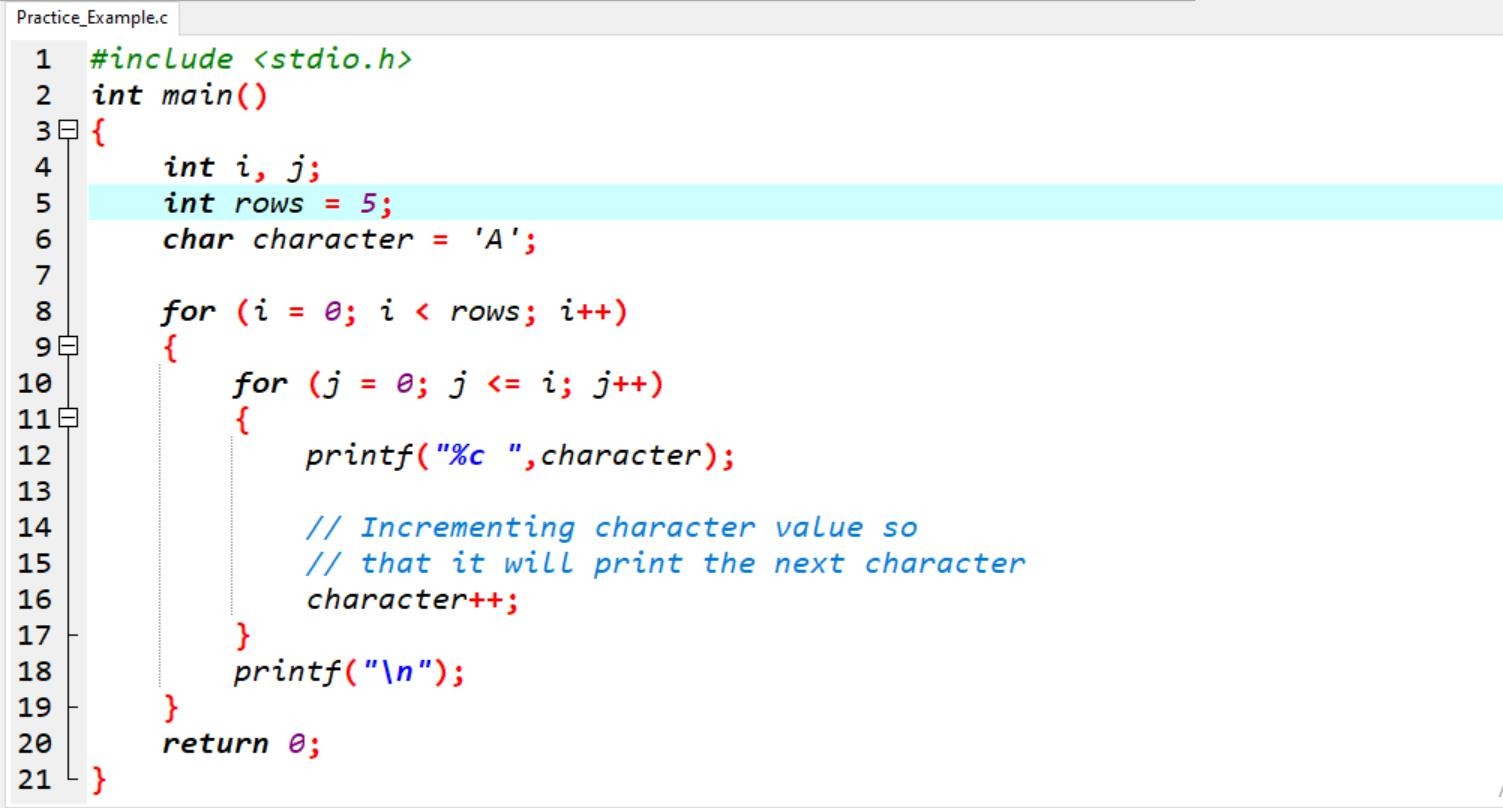


**Output:-**



2.

## Practical:-

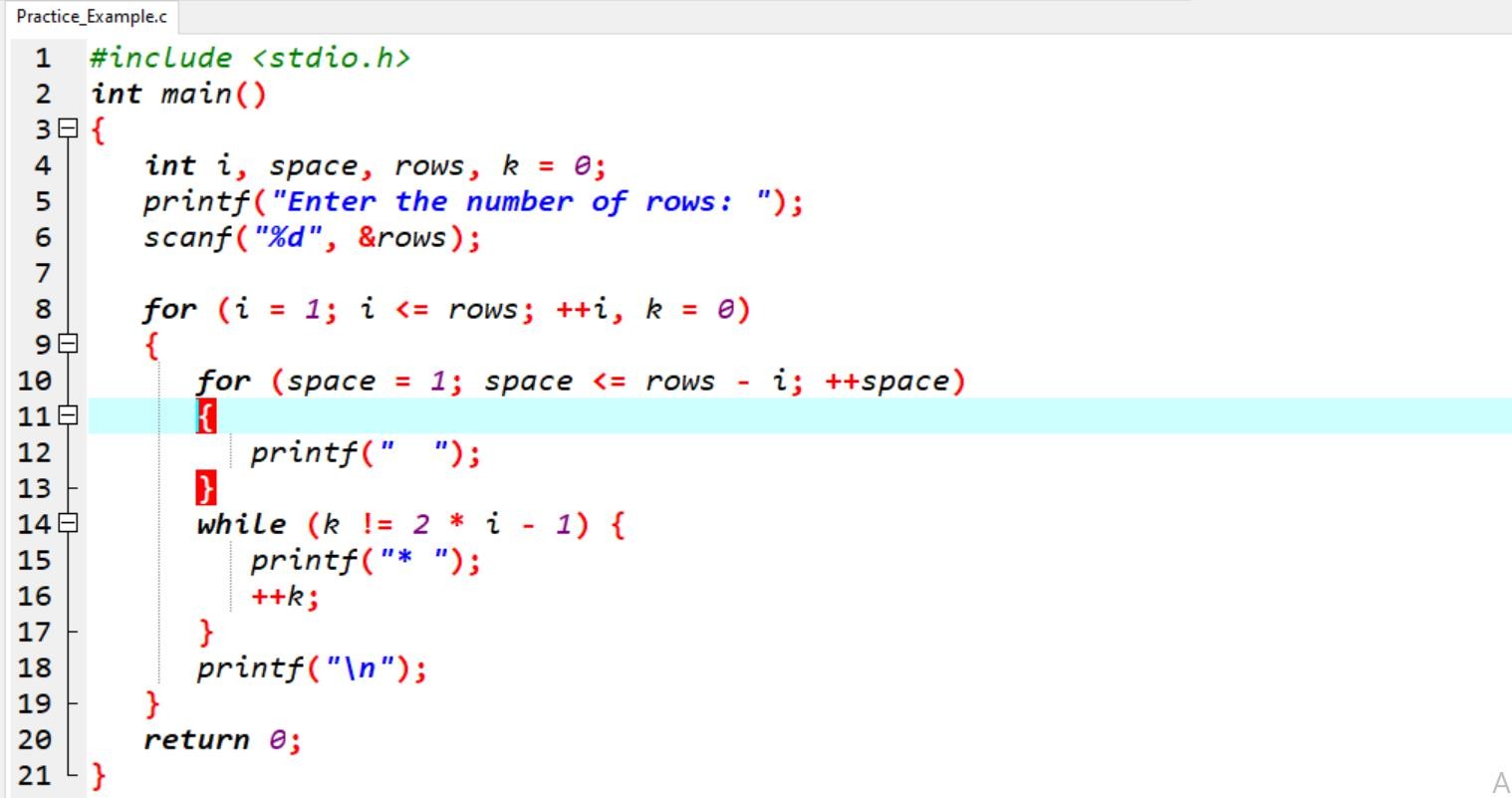


**Output:-**

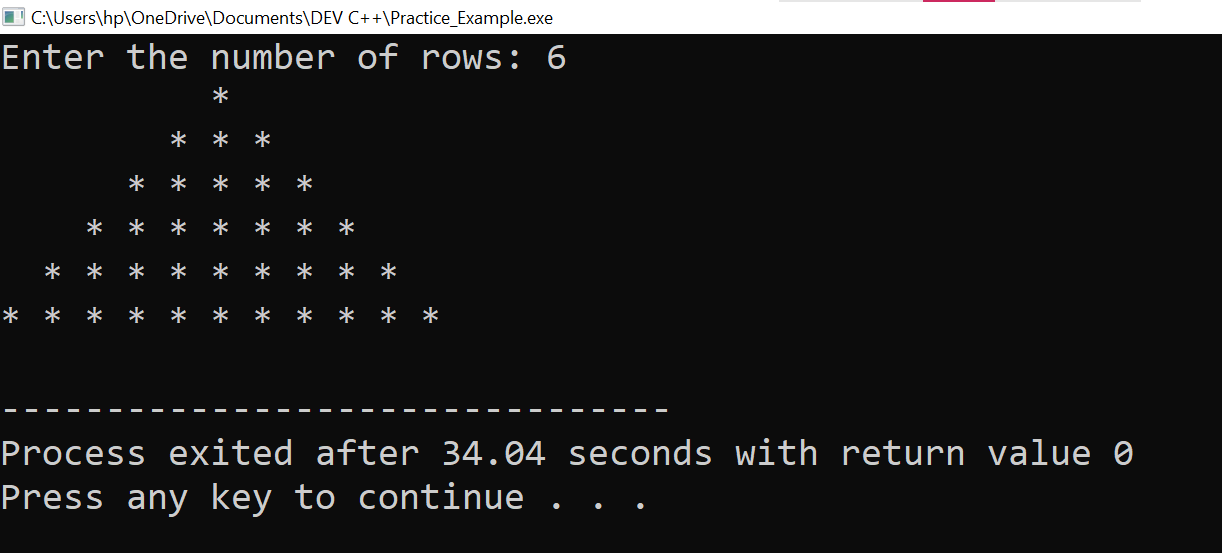


3.

## Practical:-



**Output:-**

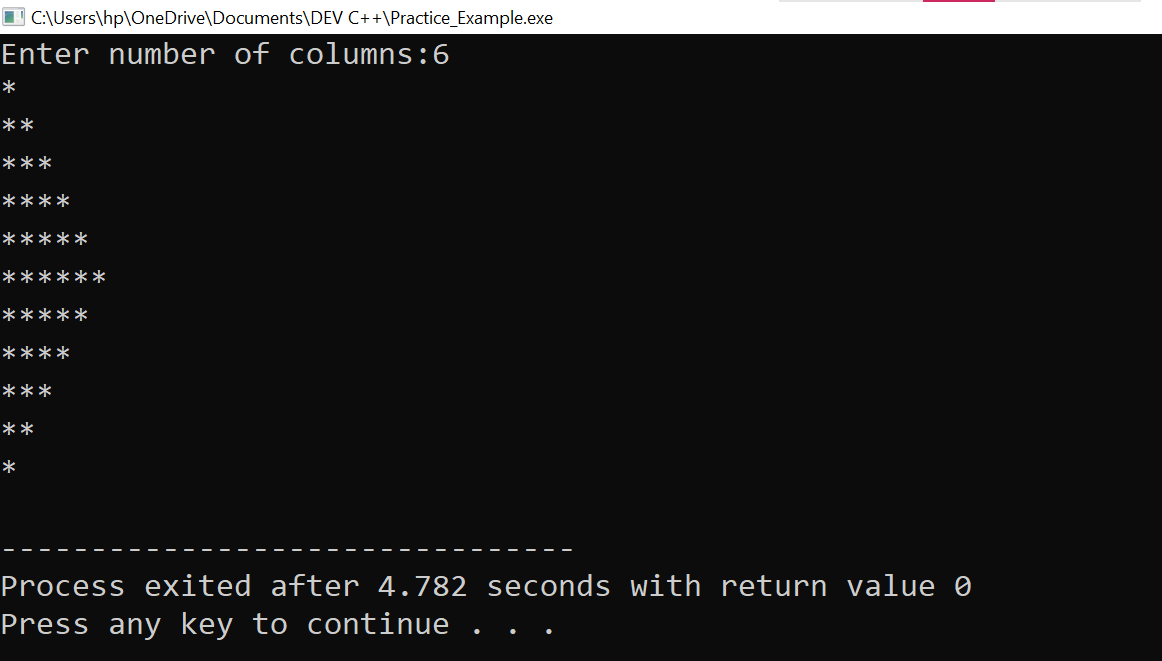


4.

## Practical:-

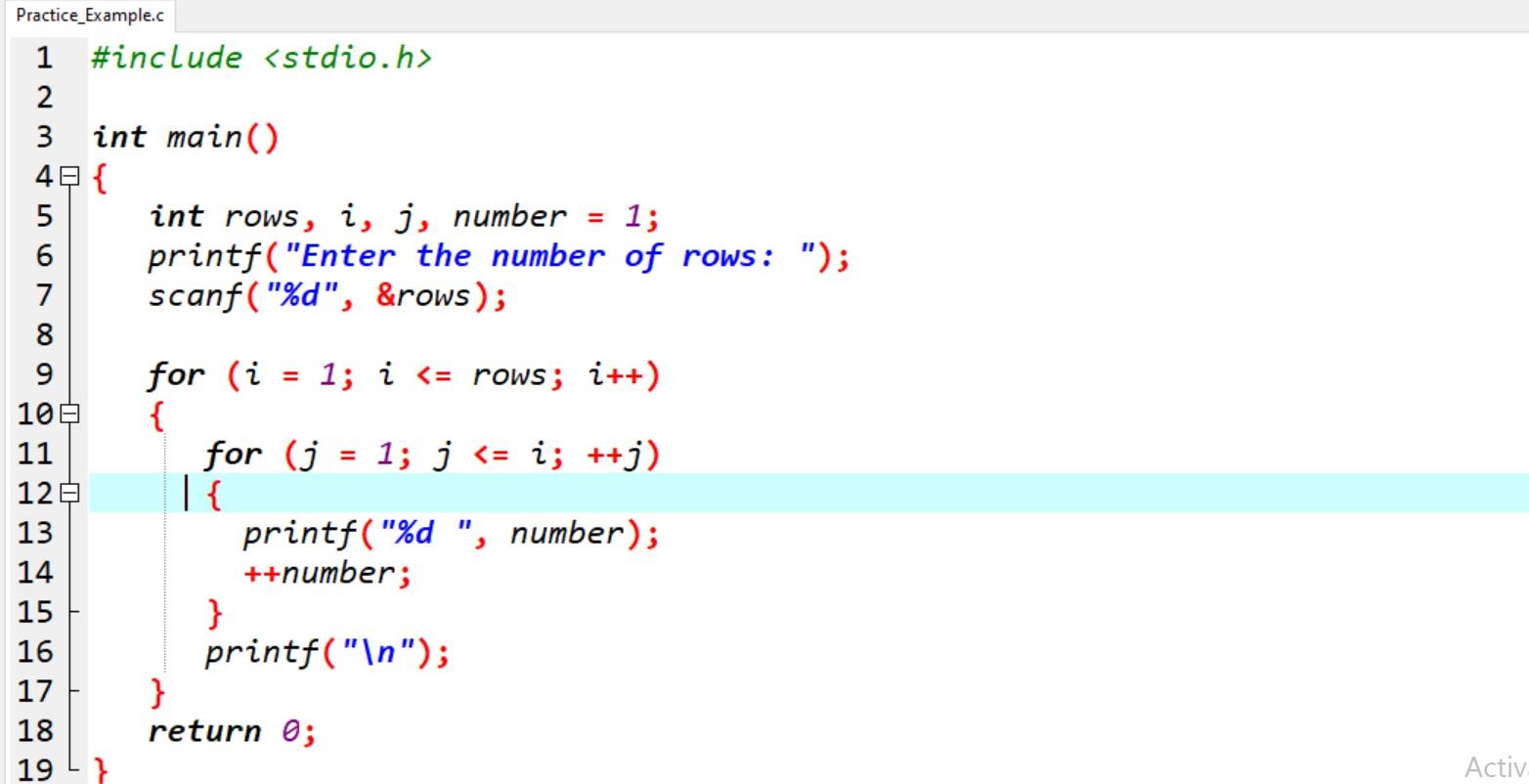


**Output:-**

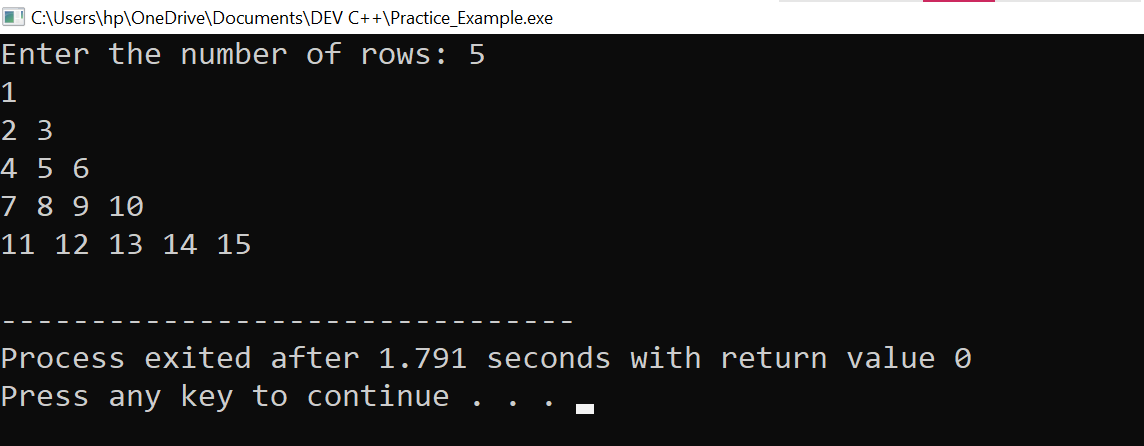


5.

## Practical:-

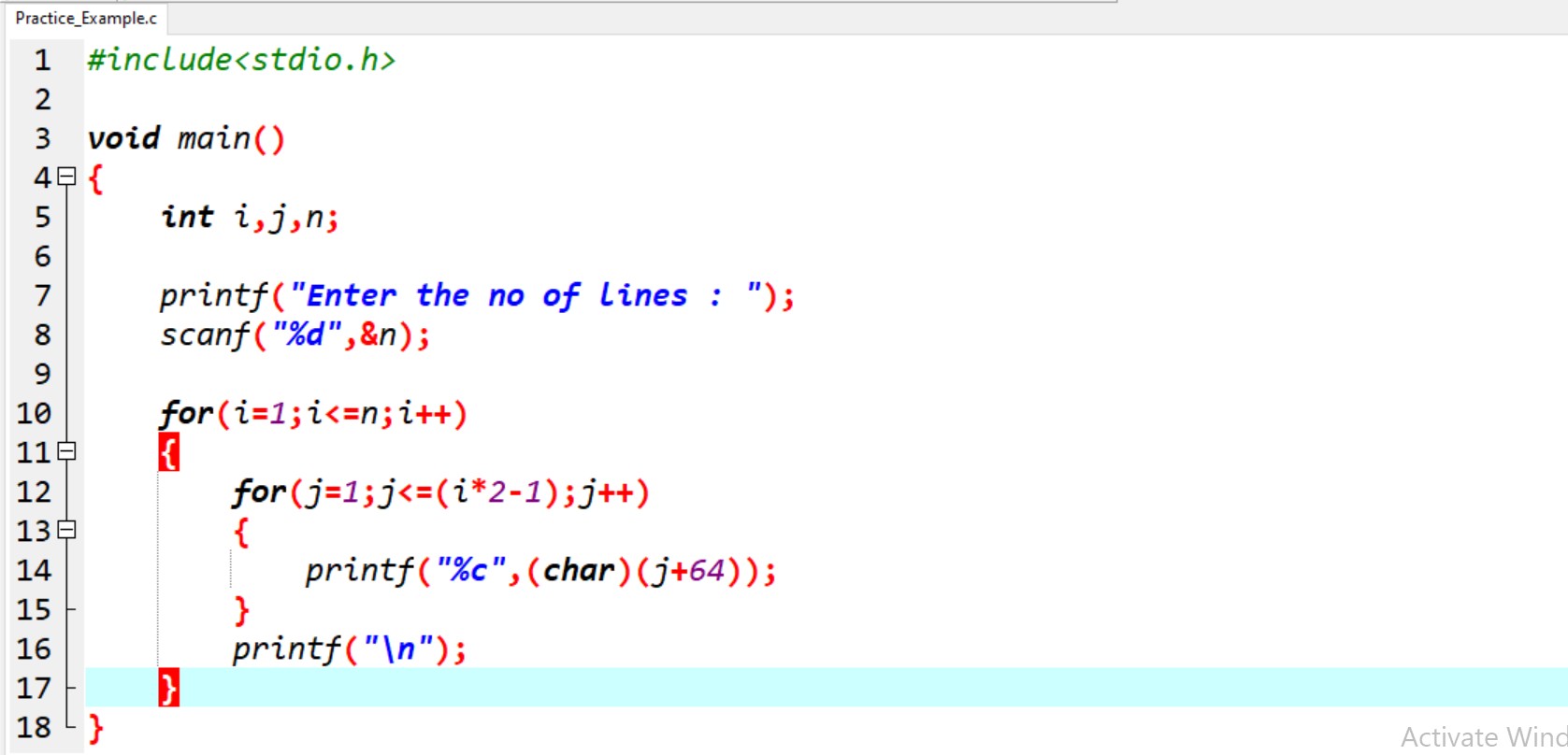


**Output:-**

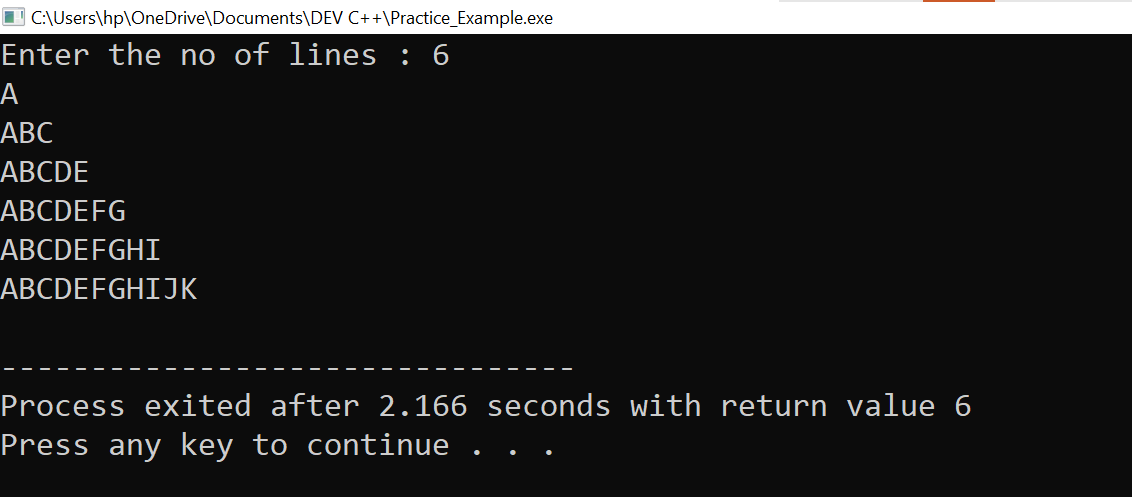


7.

## Practical:-



**Output:-**

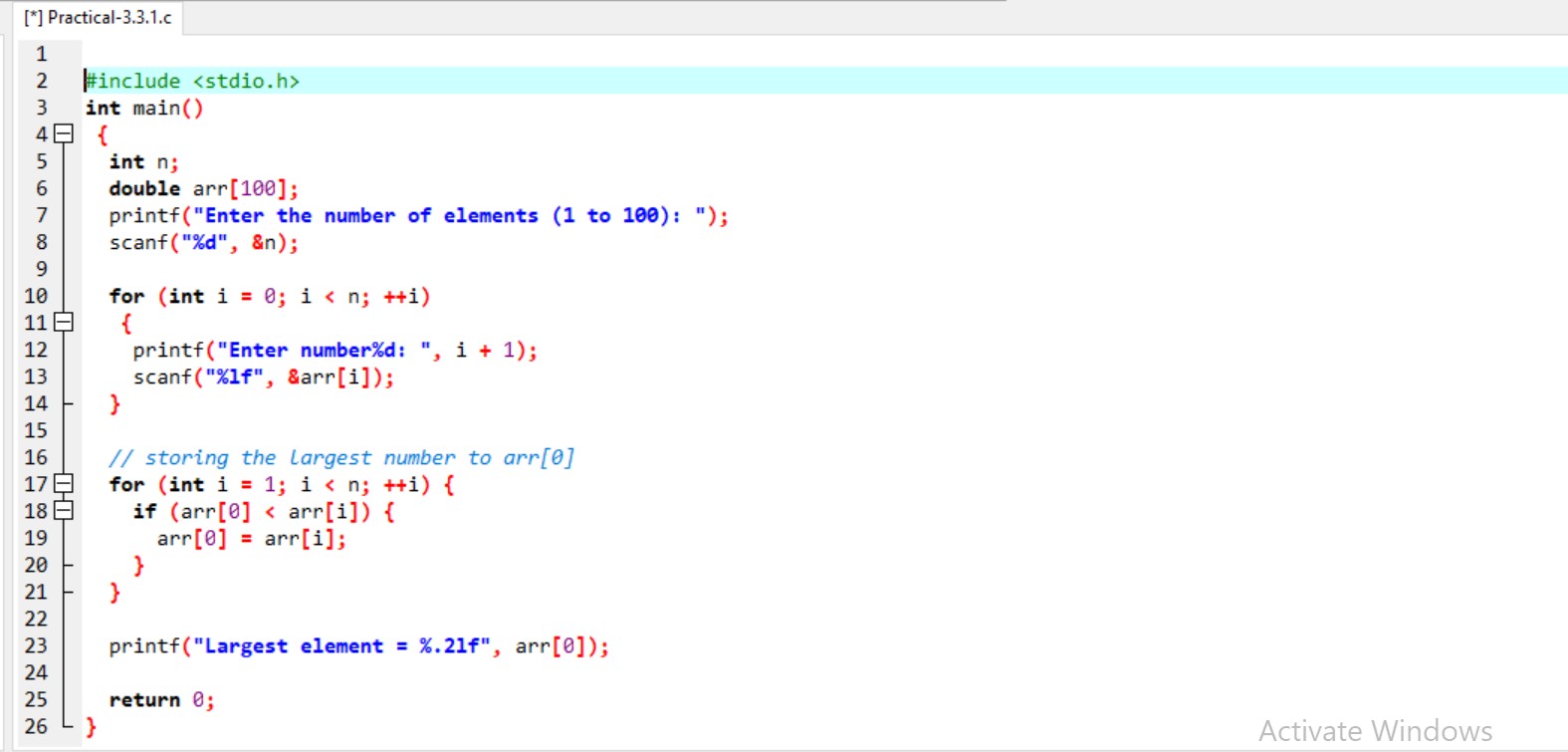


MODULE: 2.3 (File Handling and Debugging)

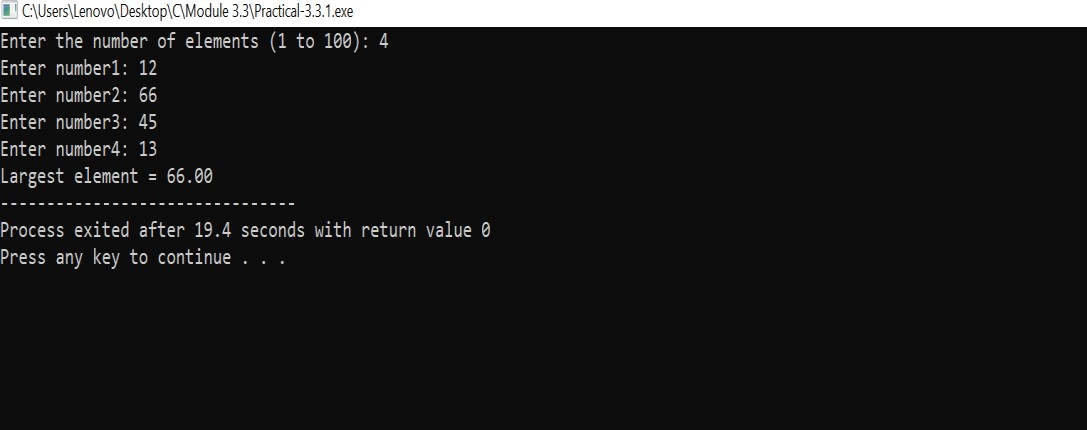
1. **Write a program to find out the max number from given array using function.**

## Ans:-

**Practical:-**



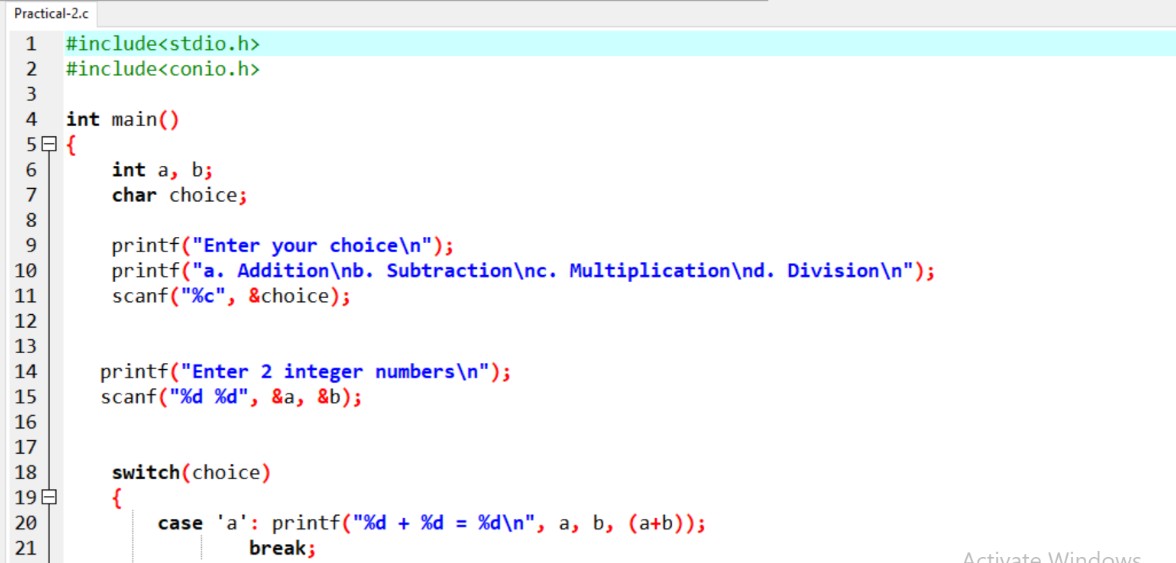
## Output:-



1. **WAP of Addition, Subtraction, Multiplication and Division using Switch case.(Must Be Menu Driven).**

## Ans:-

**Practical:-**





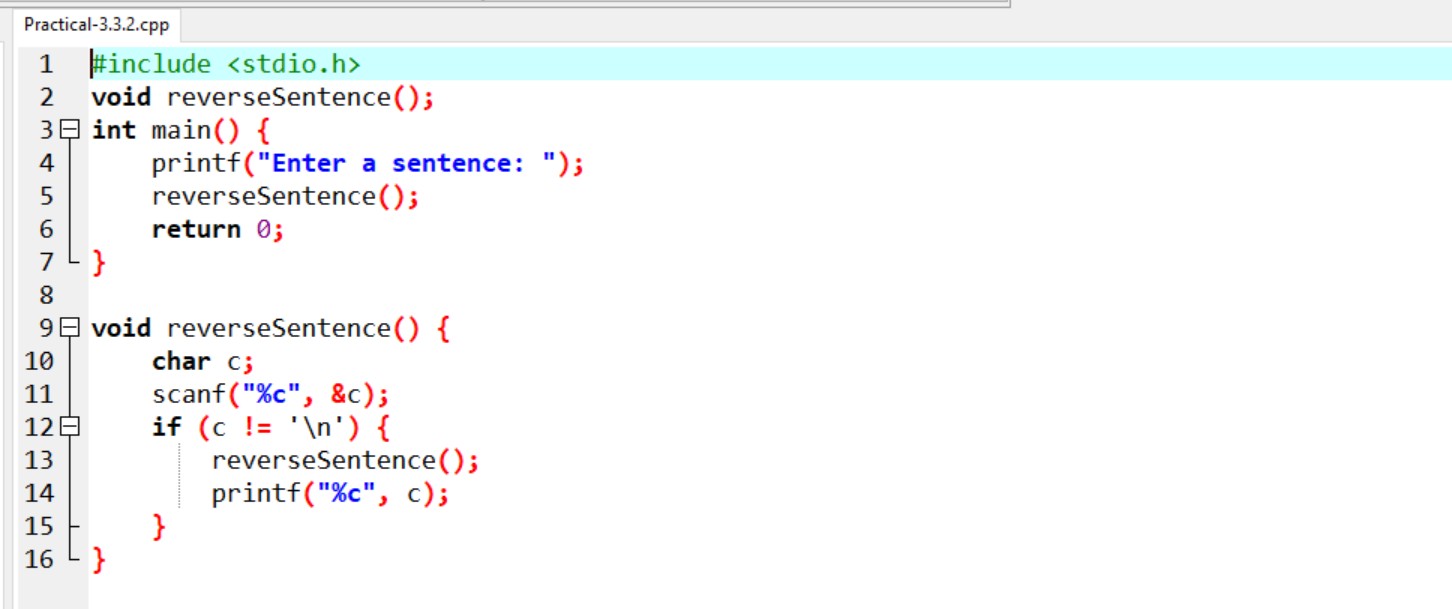
## Output:-



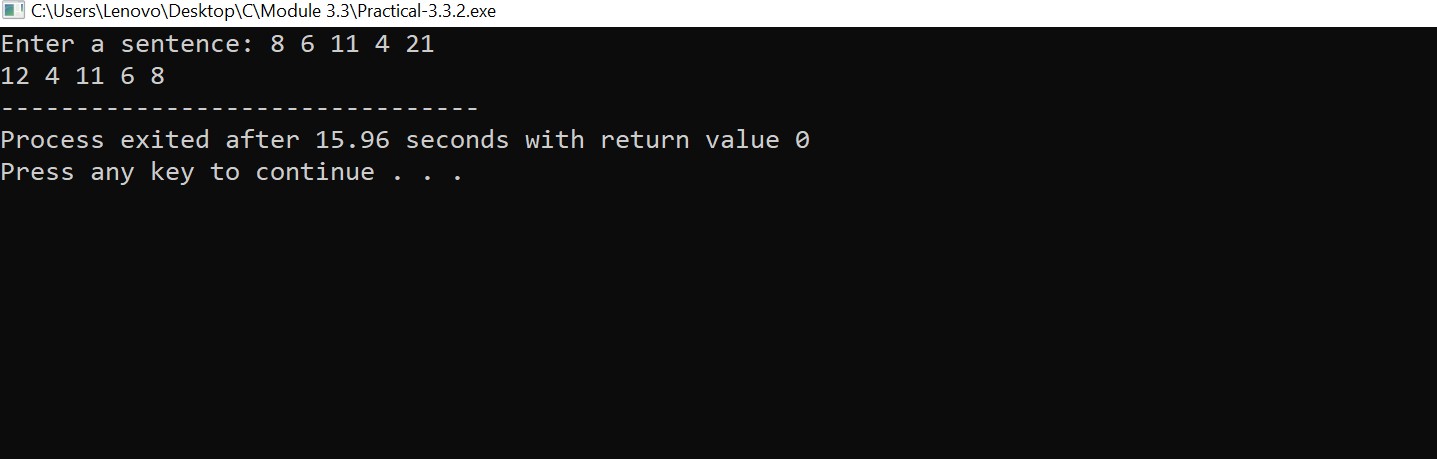
1. **WAP to find reverse of string using recursion.**

## Ans:-

**Practical:-**



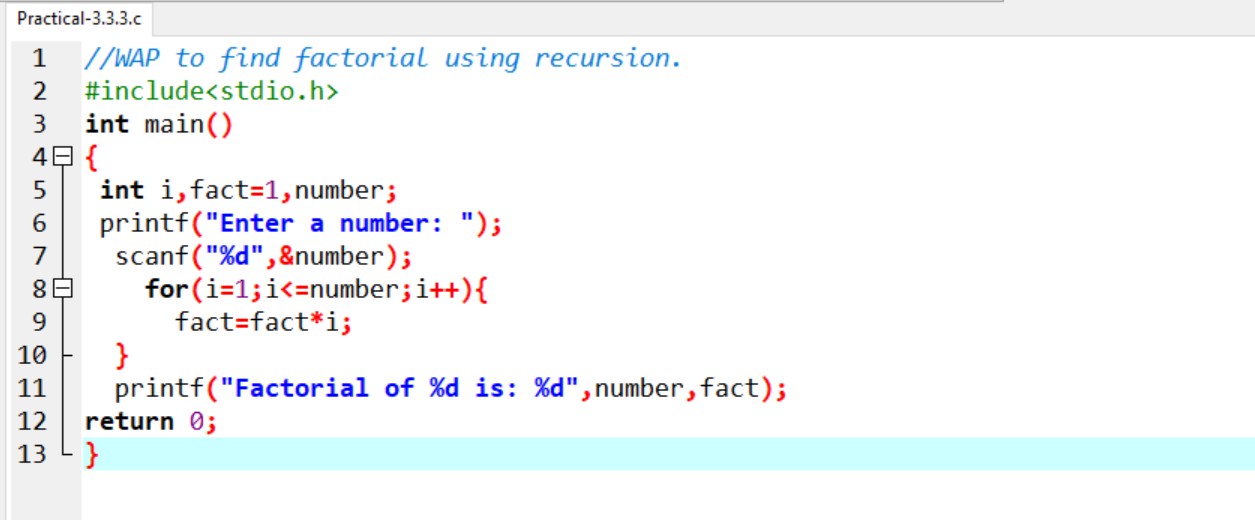
## Output:-



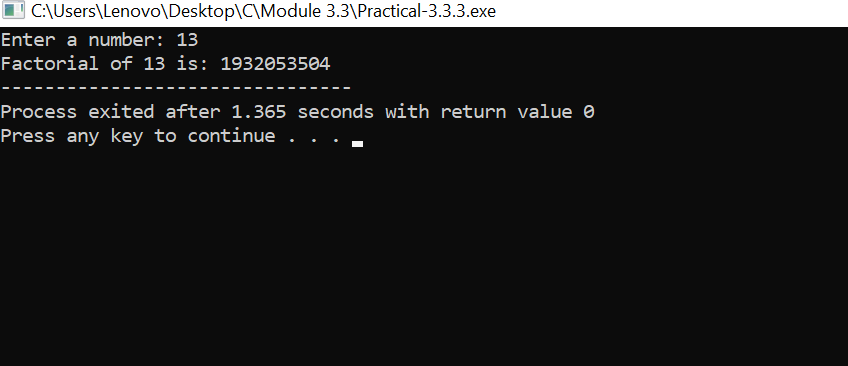
1. **WAP to find factorial using recursion.**

## Ans:-

**Practical:-**



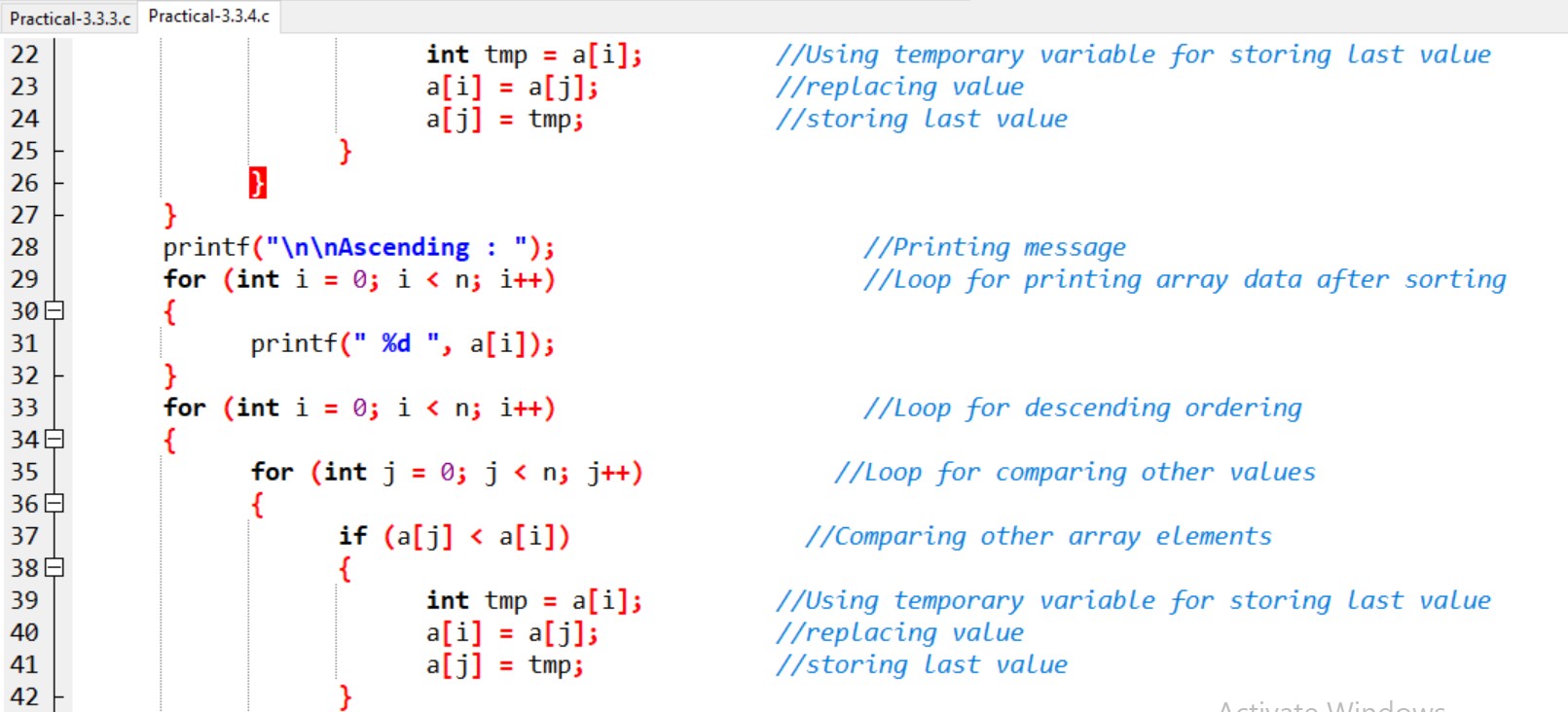
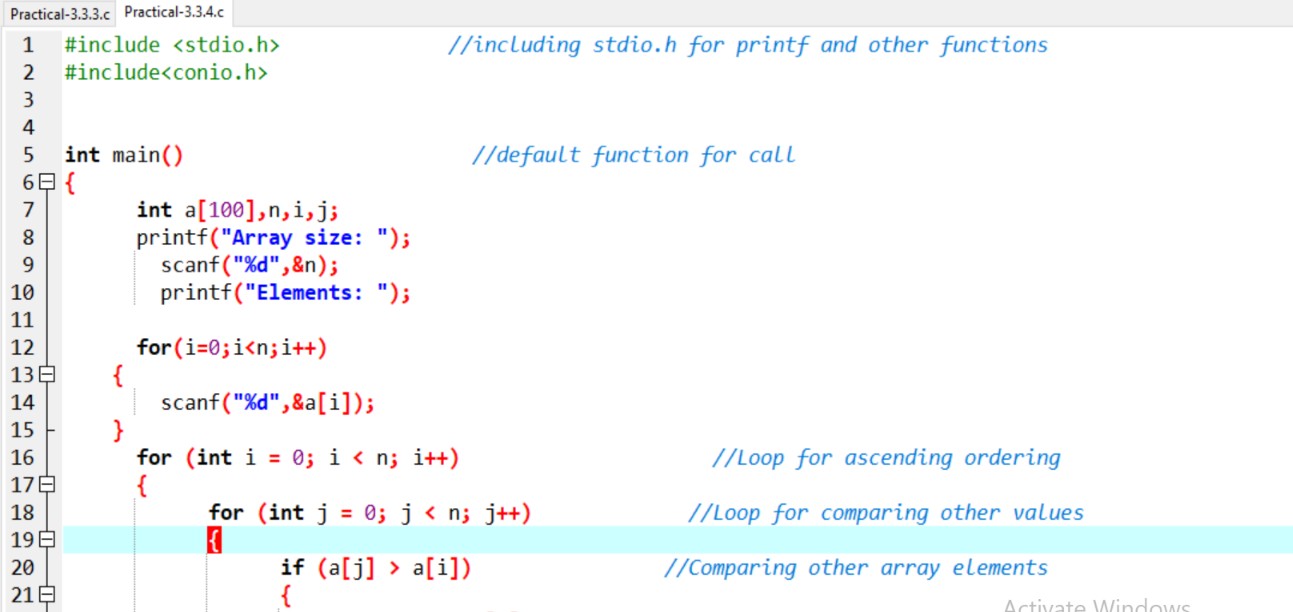
## Output:- -

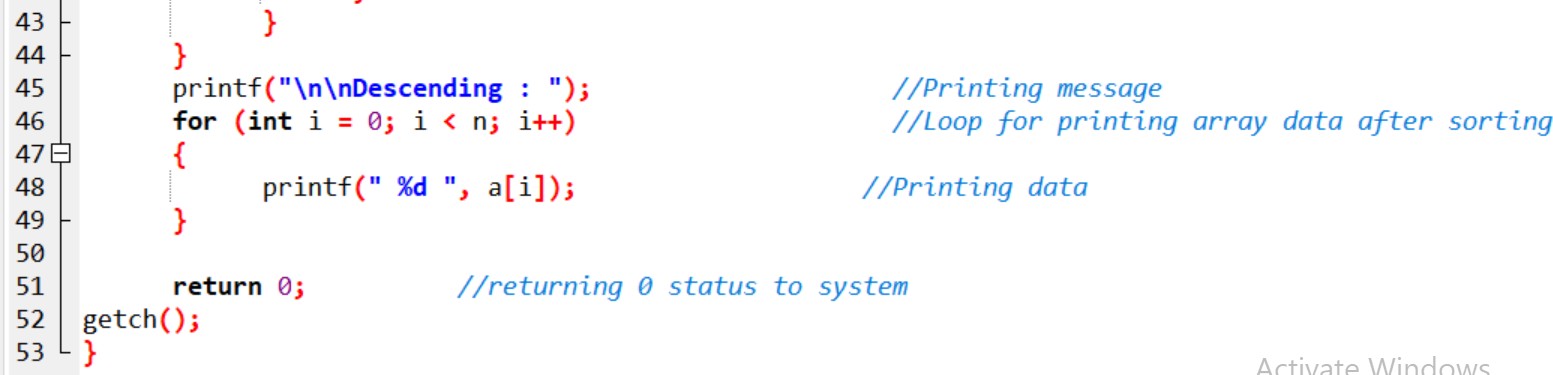


1. **WAP to take two Array input from user and sort them in ascending or descending order as per user’s choice.**

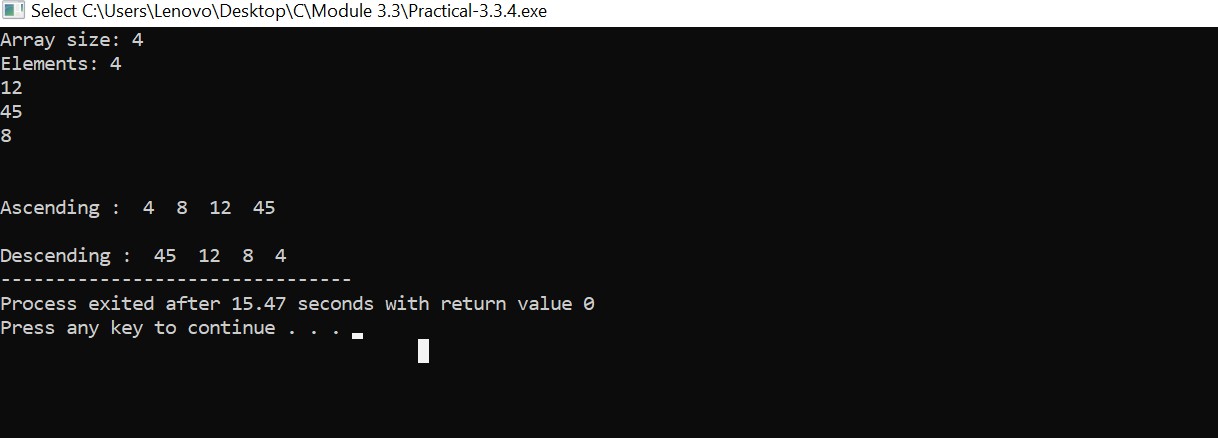
## Ans:-

**Practical:-**





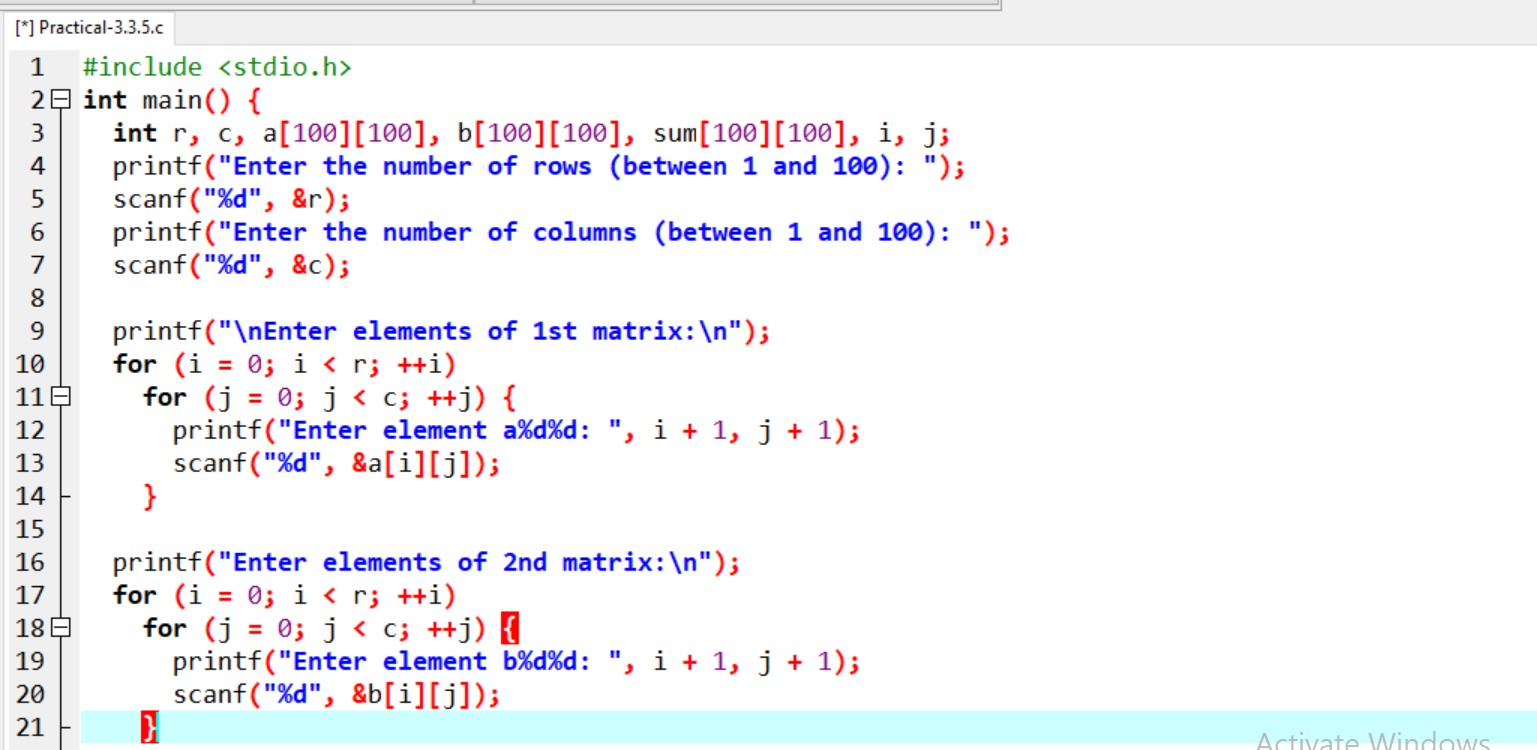
## Output:-

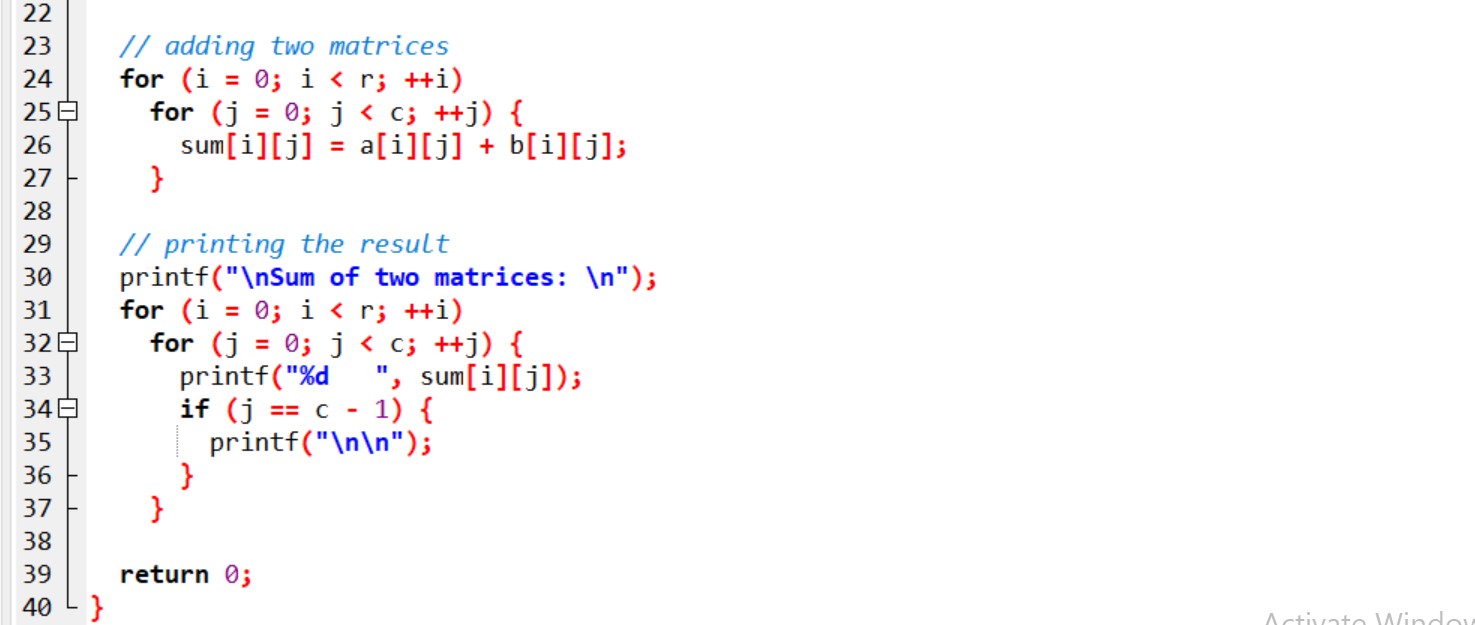


1. **WAP to make addition, Subtraction and multiplication of two matrix using 2-D Array.**

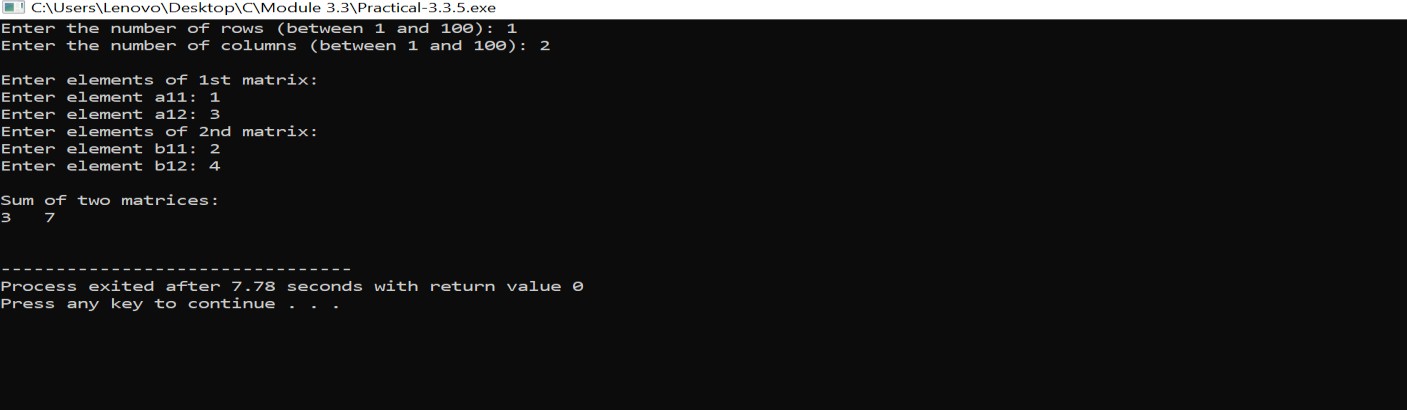
## Ans:-

**Practical:-**





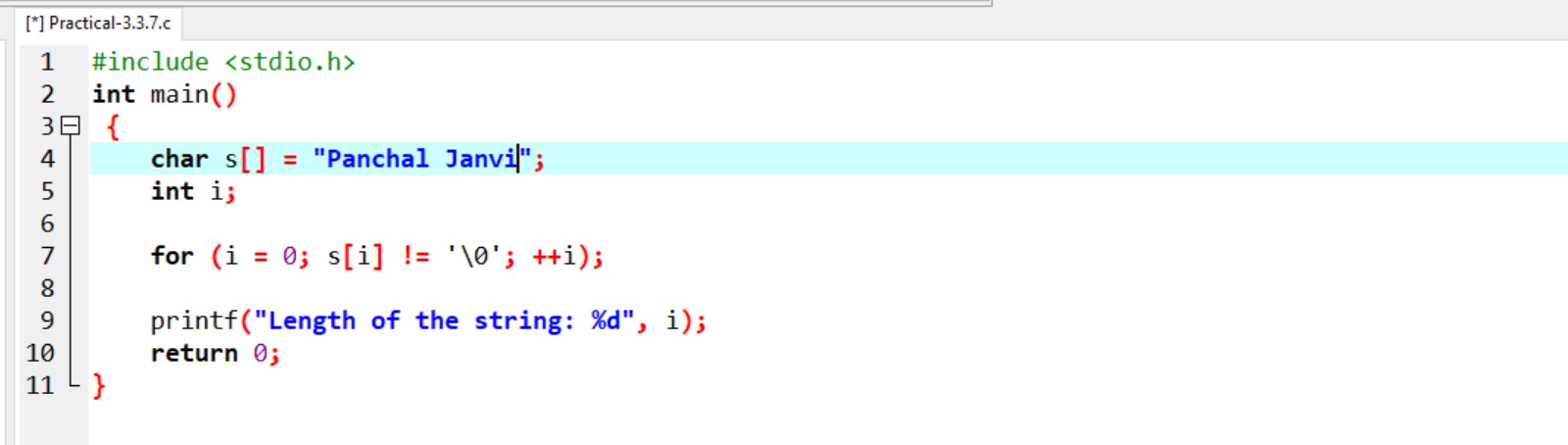
## Output:-



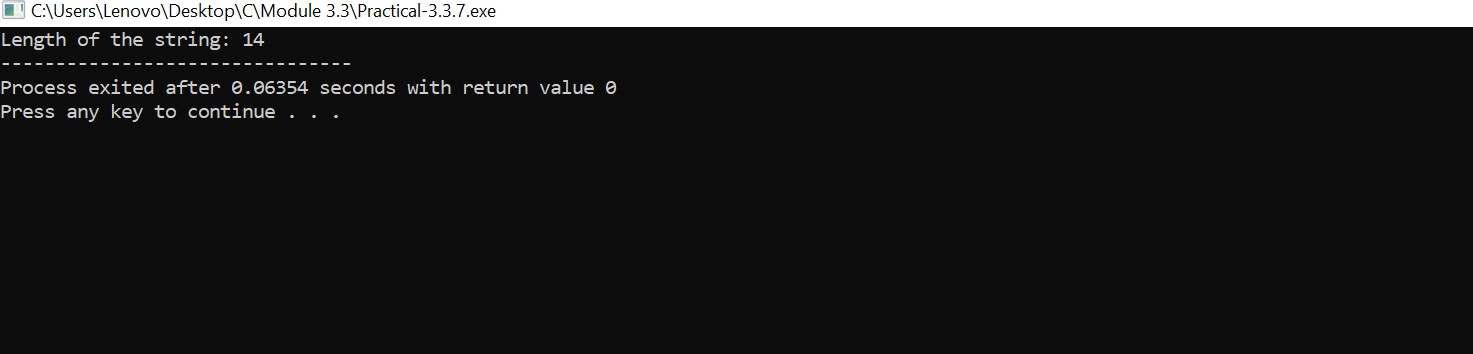
1. **WAP Find out length of string without using inbuilt function.**

## Ans:-

**Practical:-**



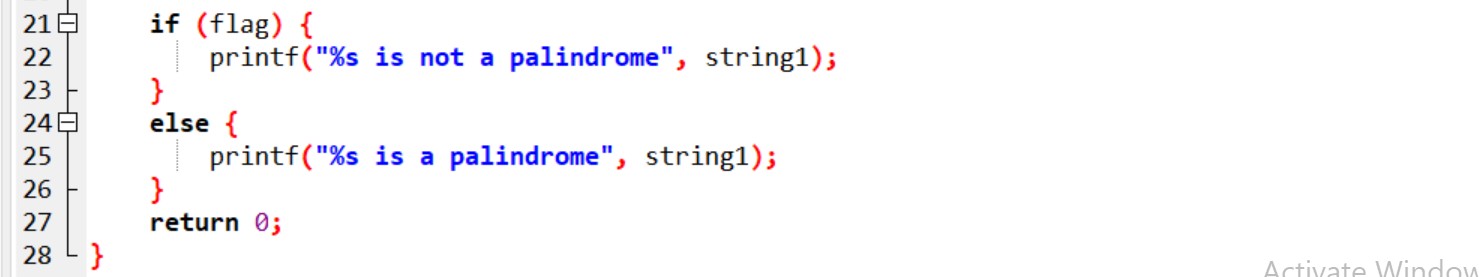
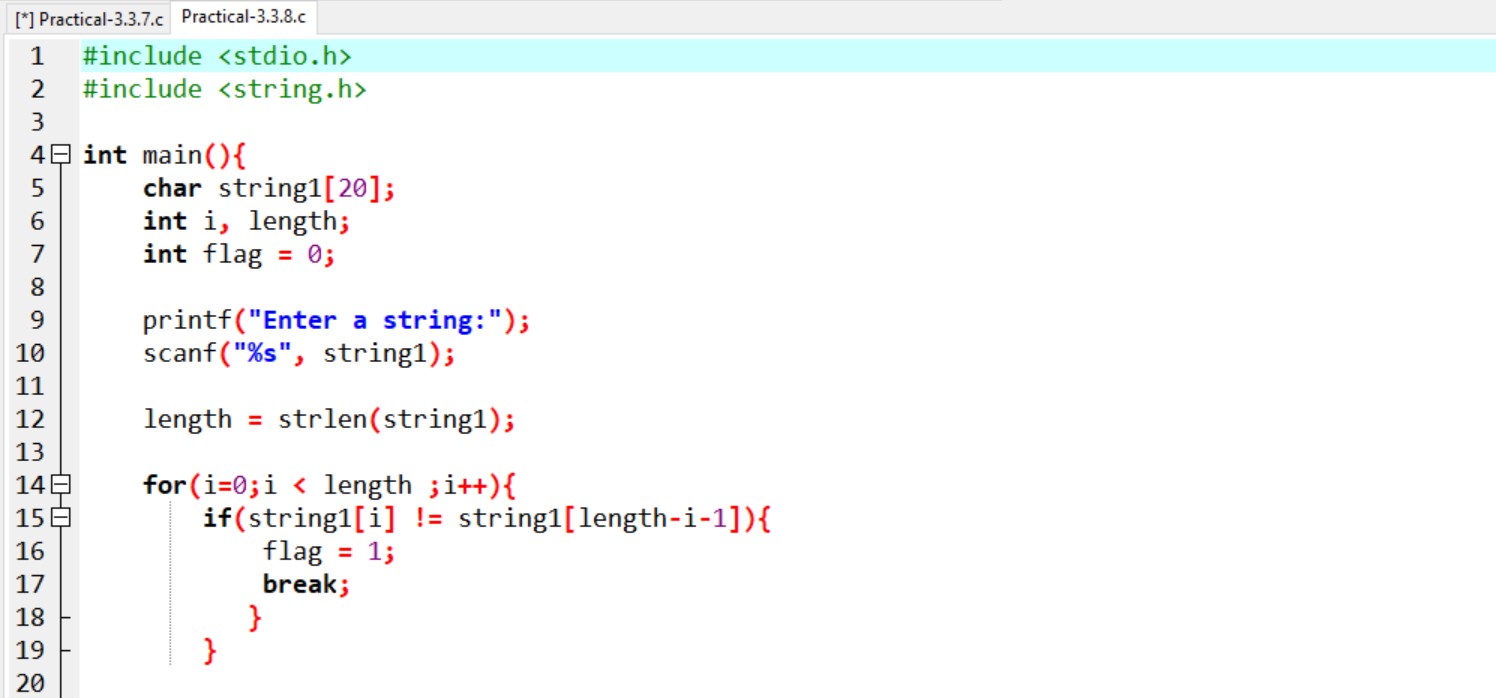
## Output:-



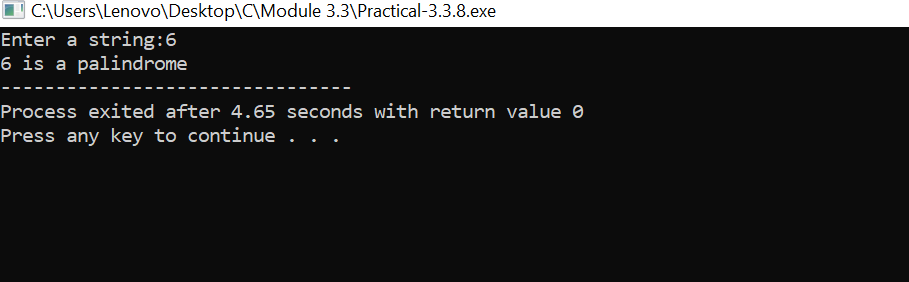
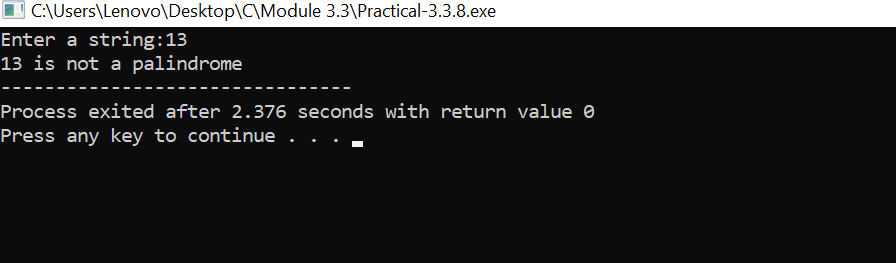
1. **WAP to reverse a string and check that the string is palindrome or not.**

## Ans:-

**Practical:-**



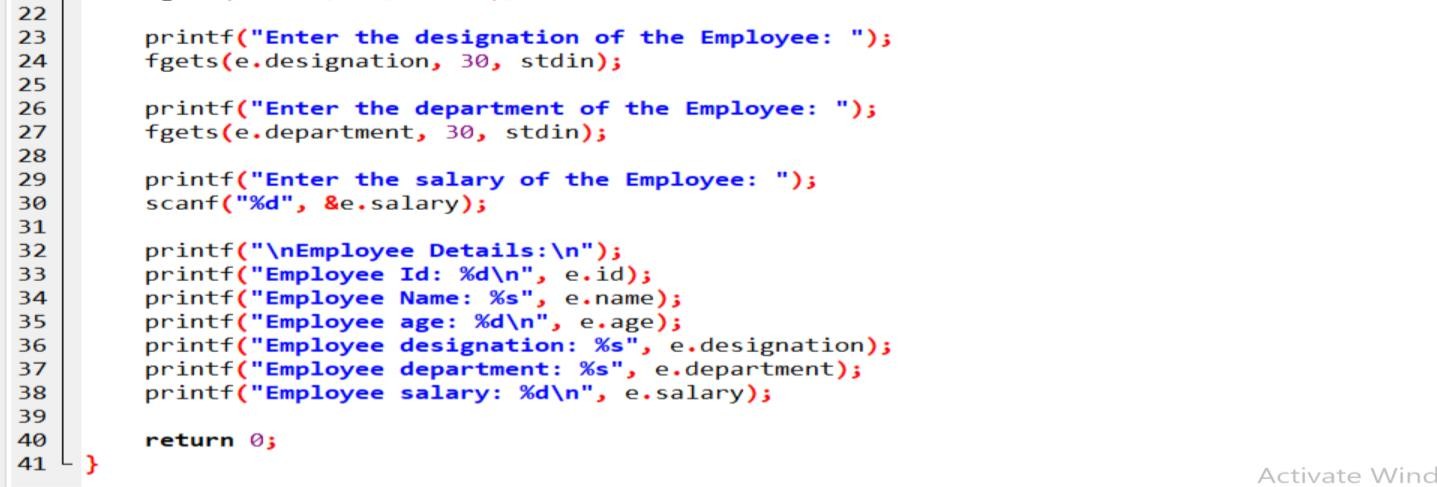
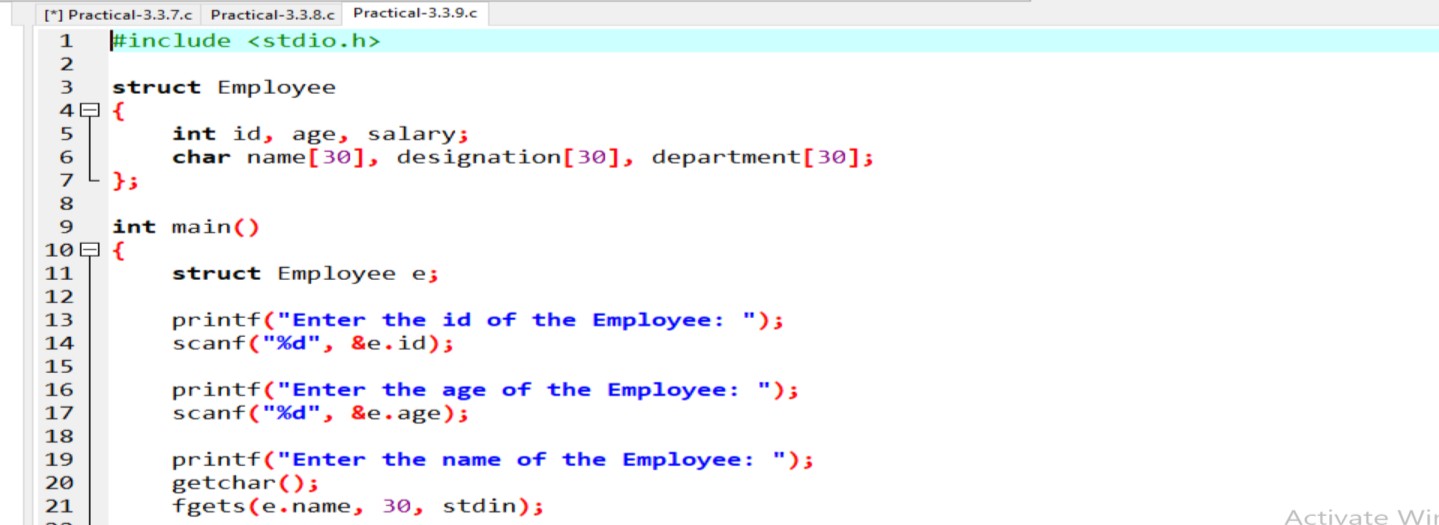
## Output:-



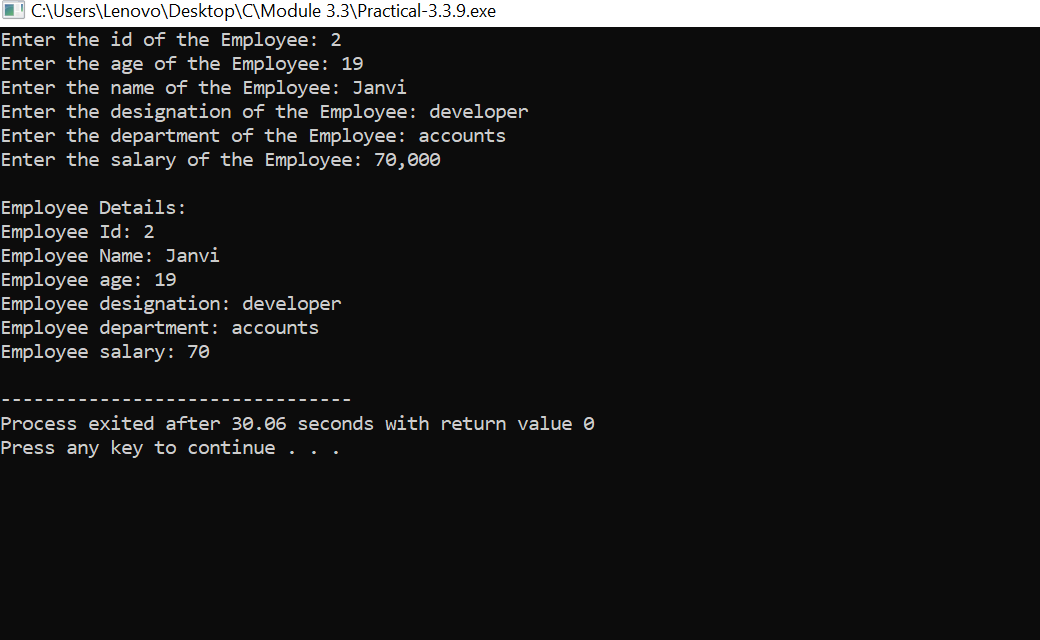
1. **Write a program of structure employee that provides the following information -print and display empno, empname, address and age.**

## Ans:-

**Practical:-**



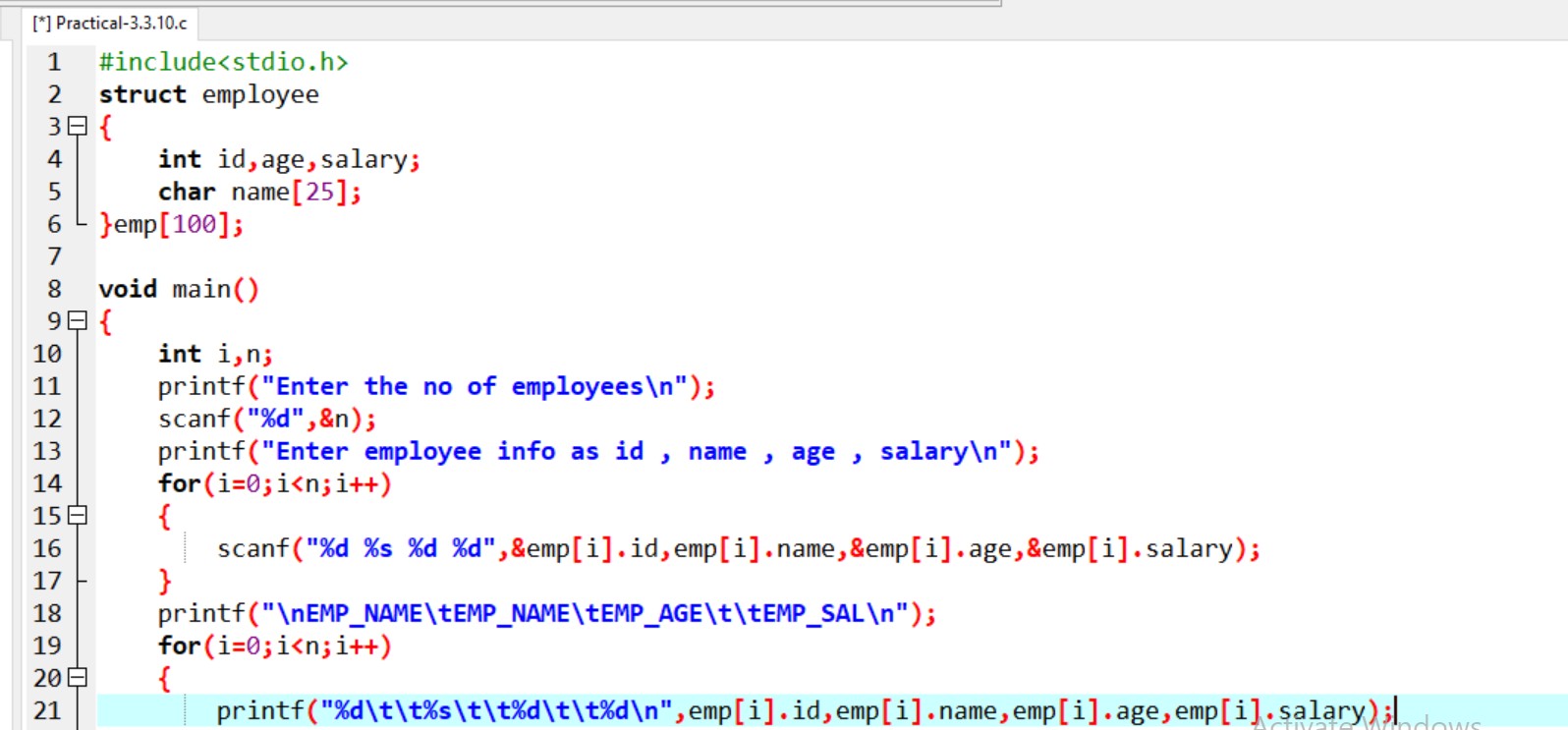
## Output:-

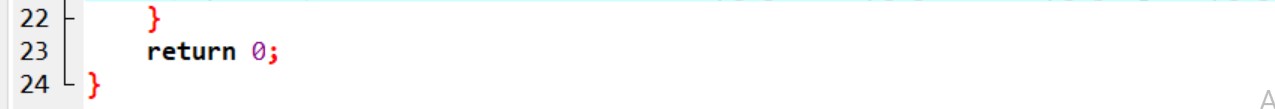


1. **Write a program of structure for five employee that provides the following information -print and display empno, empname, address and age.**

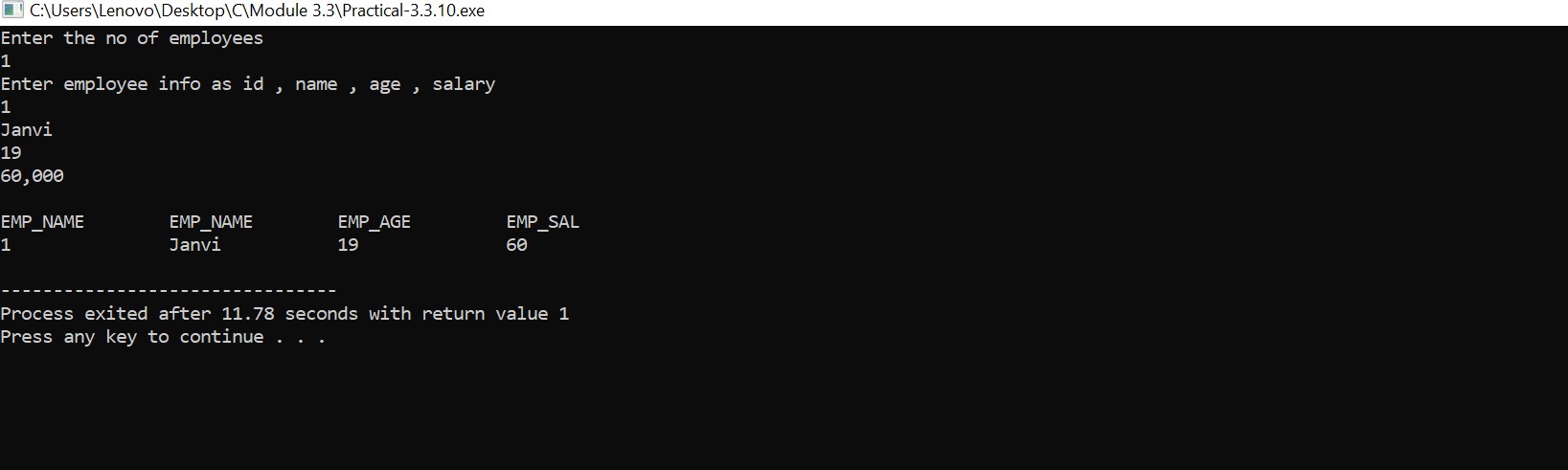
## Ans:-

**Practical:-**





## Output:-



1. **WAP to show difference between Structure and Union.**

## Ans:-

**Definition of Structure**

A structure is a custom data type in the C language. Structures can hold multiple members of different data types under a single unit. The elements of a structure are stored in contiguous memory locations and can be retrieved and accessed at any time. Every data object in a structure is a member or field.

**How to Define a Structure?**

A structure is defined using the struct statement. The struct keyword defines a new data type with more than one member.

**Syntax of Declaring a Structure**

**struct [structure name]**

**{**

**type member\_1; type member\_2;**

**. . .**

**type member\_n;**

**};**

**Example of Structure**

struct student

{

int rollno;

char name[50]; string phone;

};

**Definition of Union**

A Union is a user-defined data type. It is like the structure, except that all its members start at the exact location in memory. The union combines objects of different data types in the exact memory location. A user can define a union with many members, but only one member can hold a value at any given time. The storage space allocated for the union variable is equal to the total space required by the most prominent data member of the union.

Union provides such variables that can be accessed in several ways and the exact memory location simultaneously. A union provides an efficient way of using a single memory location for various tasks.

**How to Define a Union?**

The union statement is used for defining a union. It defines a new data type that can store multiple member variables of different data types in the exact memory location. The syntax to define a union using the union keyword is similar to defining a structure.

**Syntax of Declaring a Union**

union [union name]

{

type member\_1; type member\_2;

. . .

type member\_n;

};

**Example of Union**

union Student

{

char name[32]; int age;

string email;

};

**Similarities between Structure and Union**

The following are the similarities between structure and union:

* **Both structure and union are the custom data types that store different types of data together as a single entity**
* **The structure and union members can be objects of any type, such as other structures, unions, or arrays.**
* **Both structures or unions can be passed by value to a function and returned to the value by functions. The argument will need to have the same type as the function parameter**
* **To access members, we use the ‘.’ operator.**

