

***Name***: ***KASHIF MAJID***

***PRN: 19070122084***

***Batch***: ***CS B1***

**ASSIGNMENT 11**

**Title: Factorial Function**

***Aim*:**

# Write a C program to find the factorial of a number using recursion.

# ***Theory*:**

# In C, we can divide a large program into the basic building blocks known as a function. The function contains the set of programming statements enclosed by {}. A function can be called multiple times to provide reusability and modularity to the C program. In other words, we can say that the collection of functions creates a program. The function is also known as procedure or subroutine in other programming languages.

# **Function declaration:** A function must be declared globally in a C program to tell the compiler about the function name, function parameters, and return type.

# **Function call:** Function can be called from anywhere in the program. The parameter list must not differ in function calling and function declaration. We must pass the same number of functions as it is declared in the function declaration.

# **Function definition:** It contains the actual statements which are to be executed. It is the most important aspect to which the control comes when the function is called.

# **Algorithm:**

# 1) Start

# 2) Input a no. ‘n’ from the user

# 3) Call the function factorial giving n as argument

# 4) If (n>=1)

# Recursively multiply n with (n-1)

# Return factorial

# Else

# Return 1

# 5) Display Factorial

# 6) Stop

# ***Flowchart*:**



# ***Source Code*:**

#include<stdio.h>

long int factorial(int n)

{

if (n>=1)

return n\*factorial(n-1);

else

return 1;

}

int main()

{

int n;

printf("Enter a positive integer: ");

scanf("%d",&n);

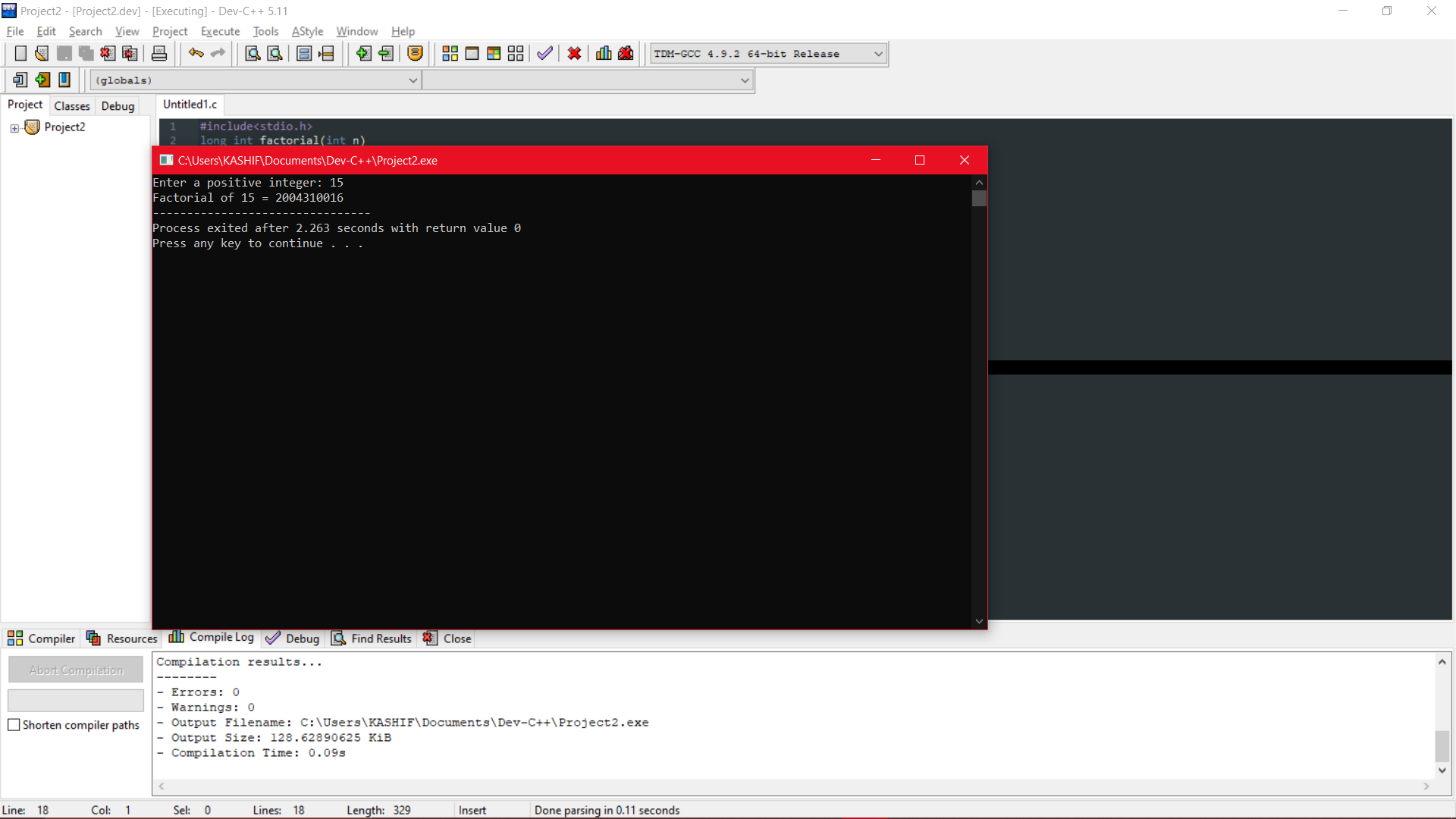
printf("Factorial of %d = %ld", n, factorial(n));

return 0;

long int factorial(int n);

}

# ***Output*:**



# ***Conclusion*:**

# Through this assignment, we have learnt about functions, their usability and how to implement functions for different operations like recursion.

# 