**ASSIGNMENT 3**

**Problem Statement:**

Function Design and Modularization - Create a document that describes the design of two modular functions: one that returns the factorial of a number, and another that calculates the nth Fibonacci number. Include pseudocode and a brief explanation of how modularity in programming helps with code reuse and organization.

**Source Code:**

**1. Factorial Function:**

- Input: An integer number (n)

- Output: The factorial of the input number (factorial)

Pseudocode:

// Function to compute the factorial of a non-negative integer

int factorial(int n) {

if (n == 0 || n == 1)

return 1; // Base case: 0! and 1! are both 1

else {

int result = 1;

for (int i = 2; i <= n; ++i)

result \*= i; // Multiply result by each integer from 2 to n

return result;

}

}

Explanation:

* The factorial function calculates the factorial of the input integer **n.**
* It uses a loop to multiply the result by each integer from **2** to**n.**
* Modularity: By encapsulating the factorial logic in a separate function, we promote code reuse and maintainability. Other parts of the program can call this function whenever factorial computation is needed.

**2**. **Fibonacci Function:**

The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones. The sequence starts with **0** and **1**, and subsequent numbers are obtained by adding the last two.

Pseudocode:

// Function to find the nth Fibonacci number

int Fibonacci(int n) {

if (n == 0)

return 0; // Base case: F(0) = 0

else if (n == 1)

return 1; // Base case: F(1) = 1

else {

int prev = 0, curr = 1, next;

for (int i = 2; i <= n; ++i) {

next = prev + curr;

prev = curr;

curr = next;

}

return curr;

}

}

Explanation:

* The Fibonacci function computes the **nth Fibonacci number**.
* It uses a loop to iteratively calculate the Fibonacci sequence.
* Modularity: Separating the Fibonacci logic into a function allows us to reuse it across different parts of the program. If we need Fibonacci numbers elsewhere, we can call this function.