

### **Assignment 3.** Function Design and Modularization

#### 1. Factorial Function:

Input: Integer n

Output: Integer (factorial of n)

Pseudocode:

arduino

function factorial(n):

    if n == 0 or n == 1:

        return 1

    else:

        return n \* factorial(n - 1)

Explanation: This function calculates the factorial of a number n recursively. If n is 0 or 1, it returns 1. Otherwise, it multiplies n with the factorial of n - 1.

#### 2. Fibonacci Function:

Input: Integer n

Output: Integer (nth Fibonacci number)

Pseudocode:

arduino

Copy code

function fibonacci(n):

    if n <= 1:

        return n

    else:

        return fibonacci(n - 1) + fibonacci(n-2)

Explanation: This function calculates the nth Fibonacci number recursively. If  $n$  is 0 or 1, Then it returns  $n$ . Otherwise, it returns the sum of the  $(n-1)$ th and  $(n-2)$ th Fibonacci numbers.

Modularity in Programming:Modularity in programming refers to the concept of breaking a program into smaller, independent, and reusable modules or functions. This approach helps in organizing code, improving readability, and facilitating code reuse. By modularizing code, developers can focus on specific tasks or functionalities, making it easier to understand, maintain, and debug the codebase.

Advantages of Modularity:

Code Reusability: Modular functions can be reused in different parts of a program or in other programs, reducing redundancy and improving efficiency.

Ease of Maintenance: Modular code is easier to maintain and update as changes can be made to individual modules without affecting the entire codebase.

Improved Readability: Modular code is easier to read and understand, making it easier for developers to collaborate and work on different parts of a program.

Scalability: Modular design allows for easy scalability as new functionalities can be added by creating new modules or modifying existing ones.

Debugging: Modular code simplifies the debugging process as issues are isolated to specific modules, making it easier to identify and fix errors.

In conclusion, modular programming promotes code reusability, organization, and maintainability, leading to more efficient and reliable software development.