**Function Design and Modularization**

**Function 1: Factorial of a Number**

The factorial of a number is the product of all positive integers less than or equal to that number. It’s denoted by n!.

**Pseudocode**

FUNCTION factorial(n)

IF n EQUALS 0 THEN

RETURN 1

ELSE

RETURN n MULTIPLIED BY factorial(n MINUS 1)

ENDIF

ENDFUNCTION

**Function 2: Nth Fibonacci Number**

The Fibonacci sequence is a series of numbers where a number is the sum of the two preceding ones, usually starting with 0 and 1.

Pseudocode

FUNCTION fibonacci(n)

IF n EQUALS 0 THEN

RETURN 0

ELSE IF n EQUALS 1 THEN

RETURN 1

ELSE

RETURN fibonacci(n MINUS 1) PLUS fibonacci(n MINUS 2)

ENDIF

ENDFUNCTION

**Benefits of Modularity in Programming**

Modularity in programming refers to the process of breaking a large into separate sub-modules, where each has a specific job. The benefits of modularity include:

Code Reuse: Modules can be reused across multiple parts of an application, reducing redundancy and the amount of code you need to write.

Organization: Modules help keep related code grouped together, which makes the code easier to understand and maintain.

Separation of Concerns: Each module should have a single responsibility. This separation of concerns makes it easier to develop, test, and debug each part of your application.

Ease of Maintenance: Changes to one module are less likely to impact other parts of the program, making it easier to update or add new features.

Faster Development: Multiple developers can work on separate modules at the same time, speeding up the development process.