**Assignment 3:**

**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.**

Ans: 1) Factorial of a number: The factorial of non-negative integer(n) is the product of all positive integers from 1 to n.

Ex: 4! = 4\*3\*2\*1 = 24

Pseudocode:

Start

function fact(n): if n equals 0:

return 1

else:

fact = 1

for i from 1 to n: fact = fact \* i

return fact

Stop

Explanation: Here we have function factorial(n). User gave input n for which the factorial has to be calculated.

Firstly, it checks if number n is 0, if number is zero then it will return 1. We use if else case in the function.

In case when number is nonzero positive integer. We initialize the variable fact to 0. It will store the product of numbers from 1 to n. Then we enter the loop from 1 to n. This loop iterates through each integer from 1 to n and multiply each number with the current value of fact. Also update the value of fact by multiplying it with current value of i.

After loop completion it gives us factorial value of number n as a fact.

2: Fibonacci: It is a mathematical sequence where each number is the sum of the two preceding ones.

**Example: 0,1,1,2,3,5,8,13,21,…..**

Pseudocode:

function fibo(n): if n equals 0:

return 0

else if n equals 1: return 1

else:

a = 0

b = 1

for i from 2 to n: temp = b

b = a + b a = temp

return b

Modularity in programming: It refers to breaking down of a program into smaller, independent modules or function. Each module or function performs a specific task and can be reused according to need in different part of the program or in other program.