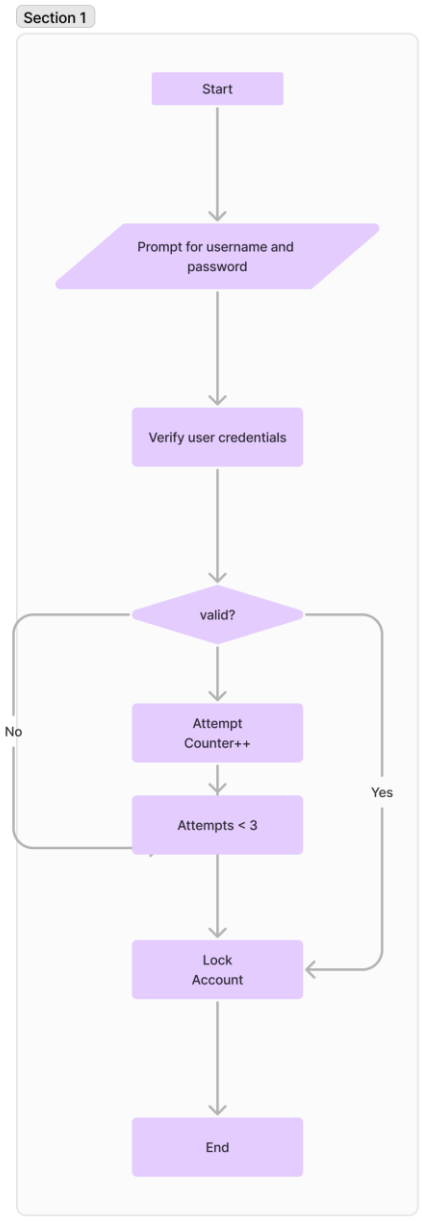
1.Flowchart Creation - Design a flowchart that outlines the logic for a user login process. It should include conditional paths for successful and unsuccessful login attempts, and a loop that allows a user three attempts before locking the account.



2. 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.

Factorial Function:

The factorial of a non-negative integer n is the product of all positive integers less than or equal to n. For example, the factorial of 5 (denoted as 5!) is: 5 * 4 * 3 * 2 * 1 = 120.

Pseudocode:

```
Function calculateFactorial(n):
  If n is less than 0:
    Return an error message

  If n is 0:
    Return 1

  Else:
    result = 1
    For i from 1 to n:
      result = result * i
    Return result
```

Fibonacci Number Function:

The Fibonacci sequence is a series of numbers in which each number is the sum of the two preceding ones. The first two numbers in the sequence are 0 and 1, and each subsequent number is the sum of the previous two. For example, the first few Fibonacci numbers are: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

```
Function calculateFibonacci(n):
  If n is less than 0:
    Return an error message

  If n is 0:
    Return 0

  If n is 1:
    Return 1

  Else:
    a = 0
    b = 1
    For i from 2 to n:
      temp = b
      b = a + b
      a = temp
    Return b
```

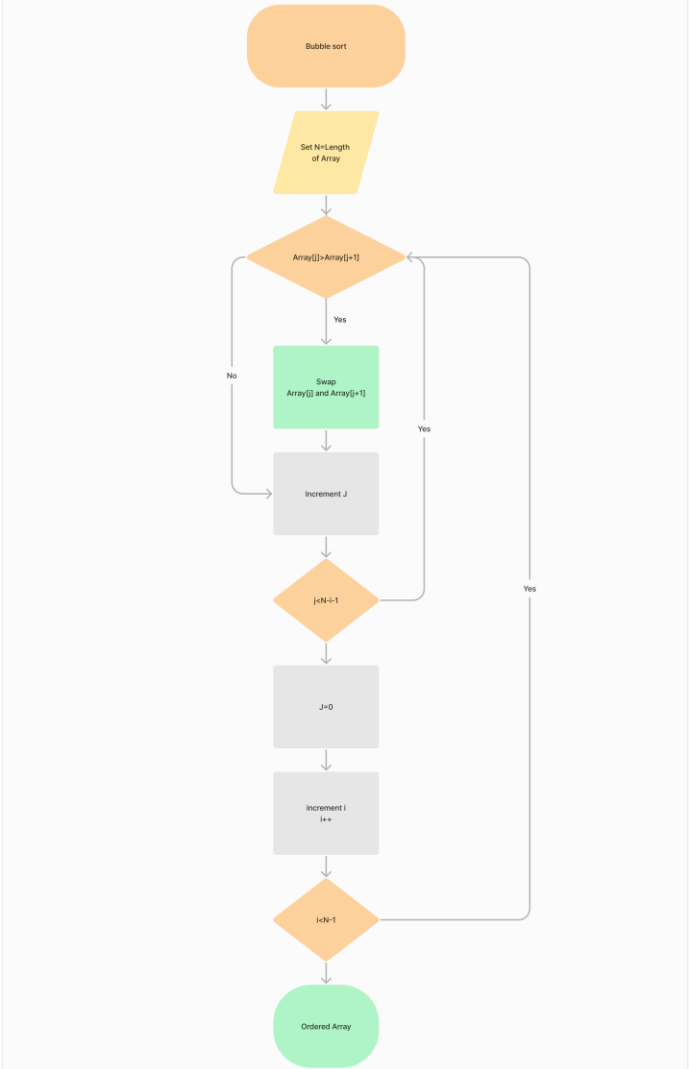
Importance of Modularity in Programming:

Modularity in programming refers to breaking down a complex task into smaller, independent modules or functions. Each module or function is designed to perform a specific task or set of related tasks. Modularity offers several benefits, including:

Code Reuse, Organization, Debugging and Testing, Scalability, Parallel Development.

3.Pseudocode and Flowchart for Sorting Algorithm - Write pseudocode and create a flowchart for a bubble sort algorithm. Provide a brief explanation of how the algorithm works and a simple array of integers to demonstrate a dry run of your algorithm.

Section 2


Bubble sort
Set N=Length of Array
Array[j]>Array[j+1]
Yes
No
Swap Array[j] and Array[j+1]
Increment J
j<N-i-1
Yes
Yes
J=0
increment i
i++
i<N-1
Ordered Array