**Q1. Pseudocode: Square or Cube Calculator**

BEGIN

PRINT "Enter a number: "

READ number

IF

number MOD 2 = 0 THEN

result = number \* number

PRINT "Square = " + result

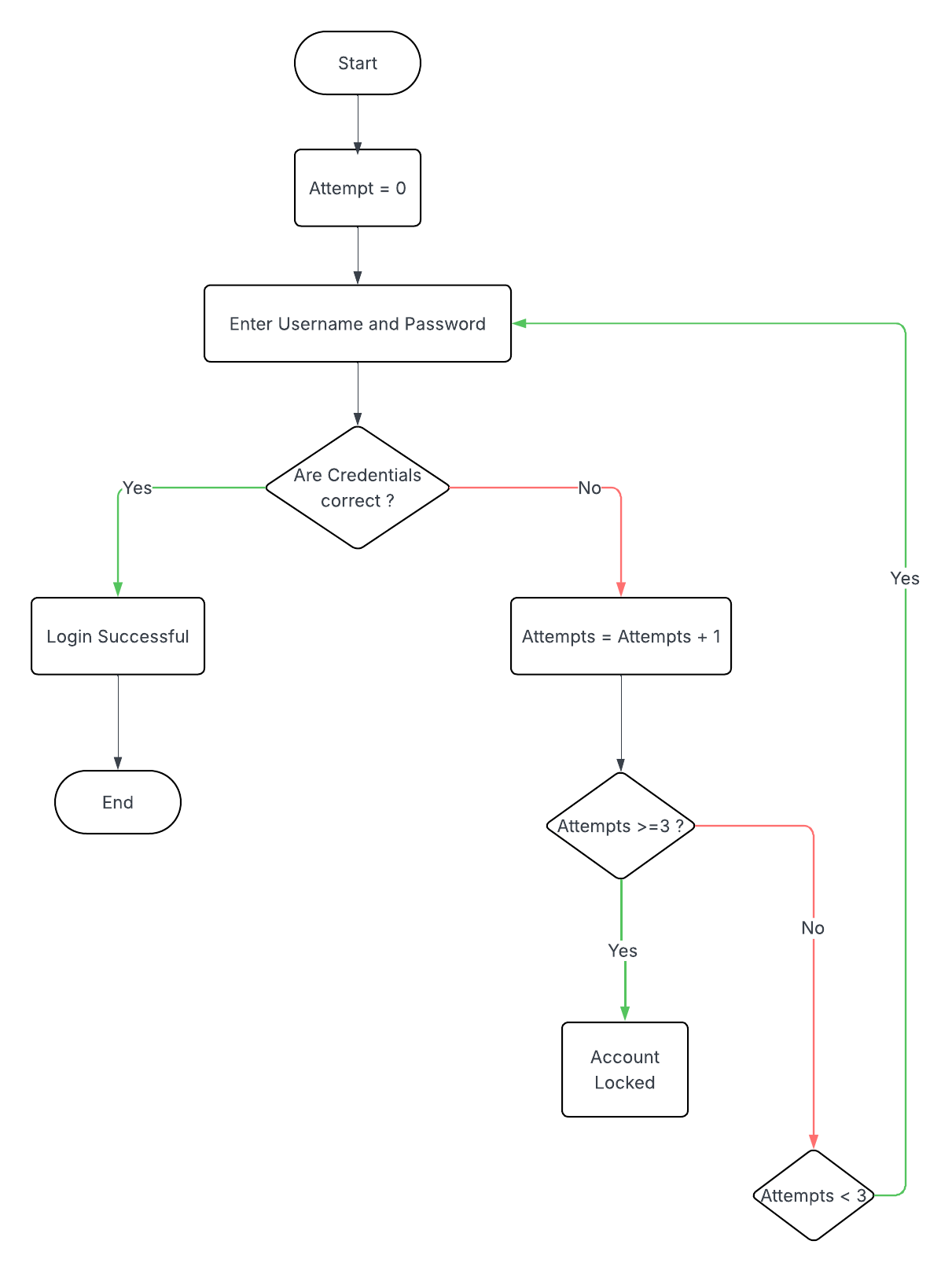
ELSE

result = number \* number \* number

PRINT "Cube = " + result

END IF

END

**Q2.**

**Q3.** **This document outlines the modular design of two functions:**

1. Factorial(n) – Returns the factorial of a number.
2. Fibonacci(n) – Returns the nth Fibonacci number.

**Function 1: Factorial**

To calculate the factorial of a number n.  
Factorial of n (written as n!) = n × (n-1) × (n-2) × ... × 1

Pseudocode:

FUNCTION Factorial(n)

IF n == 0 OR n == 1 THEN

RETURN 1

ELSE

RETURN n \* Factorial(n - 1)

END IF

END FUNCTION

**Function 2: Fibonacci**

To compute the nth Fibonacci number.  
*Fibonacci sequence:* 0, 1, 1, 2, 3, 5, 8, ...

Pseudocode:

FUNCTION Fibonacci(n)

IF n == 0 THEN

RETURN 0

ELSE IF n == 1 THEN

RETURN 1

END IF

SET a = 0

SET b = 1

FOR i FROM 2 TO n

temp = a + b

a = b

b = temp

END FOR

RETURN b

END FUNCTION

**Why Modularity Matters ?**

Using modular design for mathematical operations like factorial and Fibonacci improves the structure, clarity, and flexibility of the code. It also demonstrates best practices for building scalable and maintainable software.