ASSIGNMENT 1

1.Pseudocode Development - Task: Write a detailed pseudocode for

a simple program that takes a number as input, calculates the square if it's

even or the cube if it's odd, and then outputs the result. Incorporate

conditional and looping constructs.

Pseudocode

OUTPUT "Please enter the number"

INPUT Num

IF NUM%2==0

THEN

OUTPUT NUM \*\* 2

ELSE

THEN

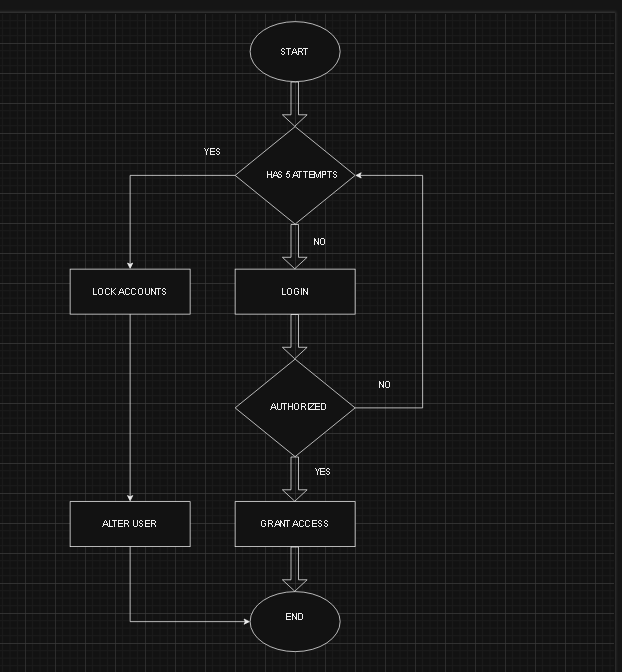
OUTPUT NUM \*\* 3

END IF

2. 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.

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.

Factorial of the number  
The factorial of a number, denoted as 'n!', is the product of all positive integers up to n. Essential in permutations, combinations, and probability, it simplifies complex calculations, like determining possible seating arrangements  
  
Using recursion  
  
An example  
Here the input is 5. At first, our i is set to 1. We are using a while loop at each step of which i is multiplied by our fact variable and incremented by 1.

\*When, i = 1, fact = 1, i becomes 2.

* When i = 2, fact = 1 X 2 = 2, i becomes 3.
* When i = 3, fact = 2 X 3 = 6, i becomes 4.
* When i = 4, fact = 6 X 4 = 24, i becomes 5.
* When i = 5, fact = 24 X 5 = 120, i becomes 6.
* i is no longer less than equal to 5. Thus, we come out of the while loop and print 120 to the screen.

In this way, we are calculating our factorial using a while loop.

If the given number is n, our loop will execute n times. Thus, the time complexity is O(N).  
  
 Pseudocode   
Print "Enter a positive integer: "

Read num

fact = Factorial( num )

CASE OF ( num )

0 : Print "The factorial of, 0, is: 1"

other:

IF ( num > 0 ) THEN

Print "The factorial of, ", num, ", is: ", fact

ELSE

Print "The factorial of a negative number is: UNDEFINED"

ENDIF

ENDCASE

END

nth Fibonacci number  
The Fibonacci Algorithm Is A Numerical Series Where Each Number Is The Sum Of The Two Preceding Ones, Starting From 0 And 1. It's A Simple And Significant Concept In Computer Science With Base Cases F(0) = 0, F(1) = 1, And Recursive Case F(N) = F(N-1) + F(N-2).

In terms of time complexity, the first implementation has a poor time complexity of O(2power of n)

, while the optimized version using dynamic programming has a better time complexity of

O(n)

Pseudocode

procedure fibonacci : fib\_num

IF fib\_num less than 1

DISPLAY 0

IF fib\_num equals to 1

DISPLAY 1

IF fib\_num equals to 2

DISPLAY 1, 1

IF fib\_num greater than 2

Pre = 1,

Post = 1,

DISPLAY Pre, Post

FOR 0 to fib\_num-2

Fib = Pre + Post

DISPLAY Fib

Pre = Post

Post = Fib

END FOR

END IF