Assignment 1: Pseudo code Development - Task: Write a detailed pseudo code 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.

Start

Input a num

If (num%2==0):

Result = num\*num

Else:

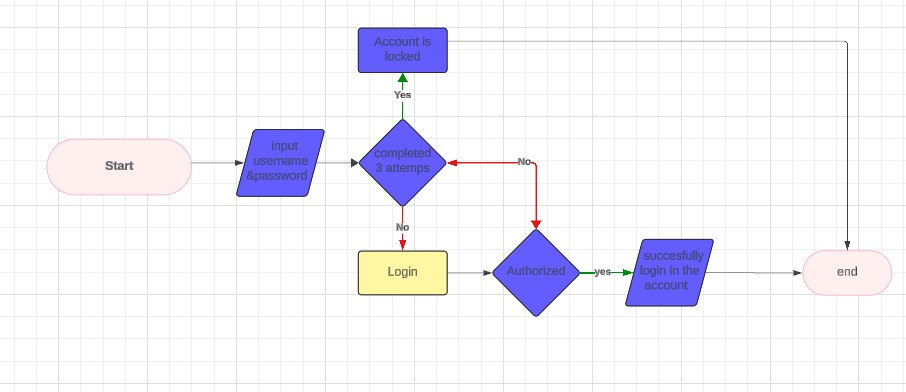
result= num\*num\*num

Display the result to the user

End

Assignment 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.



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.

1. Factorial using Function

start

input num

factorial=Call Factorial(num)

display factorial

end

function Factorial(num):

if (num==0)

return 1

else:

result = 1

for (i=1,i<=num):

result = result \* i

i=i+1

return result

End function

2.fibonacci series

Start

input n

result=call fibonacci(n)

display result

end

function fibonacci(n):

if n==0:

return 0

else if n==1:

return 1

else:

prev = 0

current = 1

for (i=2, i<=n):

temp = current

current = current + prev

prev = temp

i=i+1

return current

end function

Modularity:

Modularity means breaking down a program in smaller. Manageable, and reusable modules or functions. it helps to organize the code.

Benefits of modularity:

1. Ease of Maintenance: we can easily maintain the code if it is easily understandable.

2. Code Reusability: we can reuse the code when we are able break the code into smaller.

3. Improved Readability: modularity enhances the readability of code. We can easily read the code by breaking it into smaller.

4. Scalability: if we need to add any additional functionality to code we can easily do it.

5. Collaboration: if we use modularity then multiple programmers can support or collab in developing one code because it is easily understandable and readable.

It helps in code organization and promotes reusability by encapsulating specific functionalities within separate modules.