**Assignment 3**

**Function to calculate factorial**

function factorial(n)

if (n = 0 or n = 1)

return 1

else

return n \* factorial (n - 1)

**Algorithm:**

1. Start

2. Define a function factorial(n)

3. If n equals 0 or 1

4. Return 1

6. Return n \* factorial(n - 1)

8. End

**Explanation:**

Above function checks if the input number is 0 or 1, in both the case factorial is 1, for other number it recessively multiplies the number with the factorial of (n-1) until reach 1.

**Function to calculate Fibonacci**

function fibonacci(n)

if (n <= 1)

return n

else

return fibonacci(n - 1) + fibonacci(n - 2)

**Explanation:**

Check if n is less than or equal to 1, in this case the number is itself a Fibonacci and for other numbers it will calculate Fibonacci number at n-1 and n-2 positions and will add them together.

**Benefits of Modularity in programming:**

1. Improve readability and organization
2. Easy for testing and debugging
3. Code reusability
4. Easy for maintenance