Recording

Date : 29/06/2025

Problem Specification : Find the nth Fibonacci number using a recursive function.

Assumption : None

Limitation : This program is designed for non negative integer inputs.

Input : An integer (n)

Processing :

Calculate the nth Fibonacci number using a recursive function based on the definition: F(0) = 0, F(1) = 1, and F(n) = F(n-1) + F(n-2) for n > 1.

Output : The (n)th Fibonacci number.

Algorithm :

Step 1: Prompt the user to enter a number n.
Step 2: Read the input n.
Step 3: Call the recursive function fibonacci(n).
Step 4: If n is 0, return 0.
Step 5: If n is 1, return 1.
Step 6: Else (if n > 1), return fibonacci(n - 1) + fibonacci(n - 2).
Step 7: Display the result.
Step 8: End the program.

Programme listing : Programme file attached

Test data and expected output :

1. Test data: n = 0 Expected output: 0
2. Test data: n = 1 Expected output: 1
3. Test data: n = 5 Expected output: 5
4. Test data: n = 10 Expected output: 55

Output obtained for test data :

1. Test data: n = 0 Obtained output: Fibonacci number at position 0 is: 0
2. Test data: n = 1 Obtained output: Fibonacci number at position 1 is: 1

3. Test data: n = 5 Obtained output: Fibonacci number at position 5 is: 5
4. Test data: n = 10 Obtained output: Fibonacci number at position 10 is: 55

Analysis : The numbers of operation required in performing the algorithm.

	+,-	/,*	%	</>/<=/>=
For calculation	-	*N* times	-	-
Do while loop	*N* times	-	-	*N* times

Conclusion :

This program calculates and prints the nth Fibonacci number using a recursive approach as per the standard definition.

Discussion :

The recursive implementation of Fibonacci has a time complexity of O(2n) due to its tree-like structure and re-computation of subproblems. For larger values of n, this approach can be very inefficient. To handle negative values, additional checks would be needed at the beginning of the fibonacci function, possibly displaying an error message for invalid input.