Assignment 2: Recursive  
Function and Efficiency Analysis - Write a recursive function pseudocode and  
calculate the nth Fibonacci number and use Big O notation to analyze its  
efficiency. Compare this with an iterative approach and discuss the pros and  
cons in terms of space and time complexity.

**Pseudocode Recursive:**

Function fib(n):

If n is 0:

Return 0

Else if n is 1:

Return 1

Else:

Return fib(n-1) + fib(n-2)

**Pseudocode iterative:**

**function fibonacci(n):**

**if n equals 0:**

**return 0**

**else if n equals 1:**

**return 1**

**else:**

**a = 0**

**b = 1**

**for i from 2 to n:**

**temp = b**

**b = a + b**

**a = temp**

**return b**

Code Recursive:

#include <stdio.h>

int fib(int n)

{

if (n==0)

{

return 0;

}

else if (n==1)

{

return 1;

} else

{

return fib(n-1) + fib(n-2);

}

}

int main()

{

int n;

printf("Enter the value of n: ");

scanf("%d",&n);

printf("The %dth Fibonacci number: %d\n",n,fib(n));

return 0;

}

space complexity (Recursive): O(n)

Time complexity (Recursive): O(2^n)

space complexity (Iterative): O(1)

Time complexity (Iterative): O(n)