Q. 1. Write a program to find the n^{th} Fibonacci number using recursion.

Algorithm: Fibonacci (n)

```
        Input: integer n
        Output: n^th Fibonacci number
        Assumption: First two Fibonacci Numbers are 0 and 1.
    IF n < 2 THEN
        RETURN n  // Fibonacci(0) = 0 and Fibonacci(1) = 1
    ELSE
        RETURN Fibonacci(n-1) + Fibonacci(n-2)
    END IF
END ALGORITHM
```

Algorithm Analysis:

$T(n) = C_1 * T(n-1) + C_2$

.....

.....

$T(n) = O(2^n)$

Program:

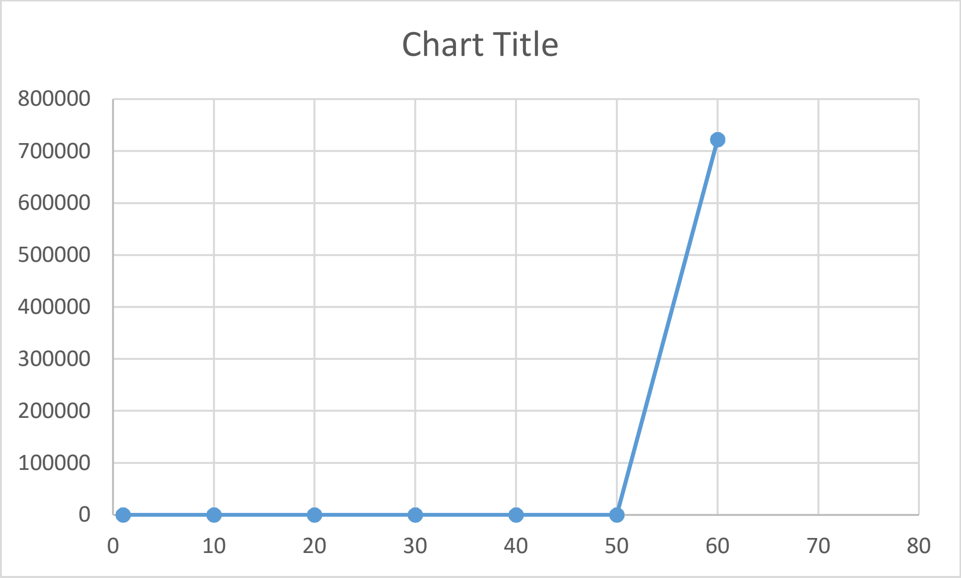
```
//Fibonacci Series using Recursion
#include<stdio.h>
int Fibonacci(int n)
{
   if (n < 2)
      return n;
   return Fibonacci (n-1) + Fibonacci (n-2);
}
int main ()
{
  int n = 9;
  printf("%d", Fibonacci (n));
  return 0;
}
```

Input/Output:

Input(n)	Output	Time
1	1	0
10	55	0
20	6765	0
30	832040	0.006
40	102334155	0.689
50	12586269025	165.019
60		722468.84
70		

80		
90		
100		

Complexity Graph:



Conclusion/Remark/Justification: