RECURSIVE

```
#include <iostream>
int fibonacciRecursive(int n) {
    if (n <= 1)
        return n;
    return fibonacciRecursive(n - 1) + fibonacciRecursive(n - 2);
}

int main() {
    int n;
    std::cout << "Enter the value of n: ";
    std::cin >> n;

    int result = fibonacciRecursive(n);
    std::cout << "Fibonacci(" << n << ") = " << result << std::endl;

    return 0;
}
```

NON RECURSIVE

```
#include<iostream>
using namespace std;
int fibonacciNonRecursive(int n) {
   if (n <= 1)
      return n;
   int a = 0, b = 1, result;
   cout<<a<<","<<b<<",";
   for (int i = 2; i <= n; i++) {
      result = a + b;
      cout<<result<<",";
      a = b;
      b = result;
   }
   return result;
}
int main() {
   int n;
   std::cout << "Enter the value of n: ";
   std::cin >> n;

   int result = fibonacciNonRecursive(n);
   std::cout << "Fibonacci(" << n << ") = " << result << std::endl;

   return 0;
}
```

/*

Recursive fibbonacci:

Time Complexity: O(n*2n)

Auxiliary Space: O(n), For recursion call stack.

Iterative fibbonacci:

Time Complexity: O(n)

Auxiliary Space: O(1)

*/