```
// Write a  non-recursive and recursive program to calculate Fibonacci
// numbers and analyze their time and space complexity.

// iterativ
// TC -O(n)
// SC -O(1)
#include <iostream>
using namespace std;

void printFibonacciIterative(int n) {
    int a = 0, b = 1, c;
    if (n >= 1) cout << a << " ";
    if (n >= 2) cout << b << " ";

    for (int i = 3; i <= n; ++i) {
        c = a + b;
        cout << c << " ";
        a = b;
        b = c;
    }
    cout << endl;
}

int main() {
    int n;
    cout << "Enter the value of n: ";
    cin >> n;
    cout << "Fibonacci sequence up to " << n << " terms (iterative): ";
    printFibonacciIterative(n);
    return 0;
}
```

```
// rec
// TC -O(2^n)
// SC -O(n)  - rec stack depth
// #include <iostream>
// using namespace std;

// int fibonacciRecursive(int n) {
//     if (n <= 1)
//         return n;
//     return fibonacciRecursive(n - 1) + fibonacciRecursive(n - 2);
// }

// void printFibonacciRecursive(int n) {
//     for (int i = 0; i < n; ++i) {
//         cout << fibonacciRecursive(i) << " ";
//     }
//     cout << endl;
// }

// int main() {
//     int n;
//     cout << "Enter the value of n: ";
//     cin >> n;
//     cout << "Fibonacci sequence up to " << n << " terms (recursive): ";
//     printFibonacciRecursive(n);
//     return 0;
// }
```