

```

// fib(n) = n if n == 0 or n == 1
// fib(n) = fib(n - 2) + fib(n - 1) if n >= 2

// if (n <= 1)
// return(n) ;
// lofib = 0;
// hifib = 1;
// for (i = 2; i <= n; i++) {
//     // x = lofib;
//     // lofib = hifib',
//     // hifib = x + lofib;
// } / * fim for */
// return(hifib);
public class Fibonacci {
    int contadorRecurs=0;
    int contadorIterativa=0;

    public int fibonacciRecursiva(int n) {
        if(n == 0 || n == 1) {
            // contadorRecurs++;
            return n;
        } else {
            contadorRecurs++;
            return fibonacciRecursiva(n - 2) + fibonacciRecursiva(n - 1);
        }
    }

    public int fibonacciIterativo(int n) {
        int x;
        int pequenoFib = 0;
        int grandeFib = 1;

        if(n <= 1) {
            // contadorIterativa++;
            return n;
        }

        for(int i =2; i <=n; i++) {
            x = pequenoFib;
            pequenoFib = grandeFib;
            grandeFib = x + pequenoFib;
            contadorIterativa++;
        }

        return grandeFib;
    }
}

```

