```
package collatz;

public class Collatz {
    public static final long TOO_BIG = (Long.MAX_VALUE - 1)
/ 3;

    public static long next(long n){
        // check for bad input
        if (n < 0){
            return 0;
        }
        if (n % 2 == 0){
            return n / 2;
        }
        // check for input number too big
        if (n > TOO_BIG){
            return 0;
        }
        return 3 * n + 1;
    }

    public static void test01(){
        System.out.println("Start test 01.");
        System.out.println("17 --> " + next(17));
        System.out.println("16 --> " + next(16));
        for(long i = 0; i < 11; ++i){
            System.out.println(i + " --> " + next(i));
        }
    }

    public static void test02(){
        System.out.println("\ntest big numbers\n");
        long bigEven = 4_000_000_000L;
        long bigOdd = 4_000_000_001L;
        long wayBig = Long.MAX_VALUE / 2;
        System.out.println(bigEven + " --> " +
next(bigEven));
        System.out.println(bigOdd + " --> " +
next(bigOdd));
```

```
        System.out.println(wayBig + " --> " +
next(wayBig));

    }

    public static void drawOrbit(long start){
        long current = start;
        long count = 0;
        System.out.print(start);
        while(current >1){
            ++count;
            current = next(current);
            System.out.print(" ---> " + current);
            if (count % 5 == 0){
                System.out.print("\n ");
            }
        }
    }

    public static void biggestSoFar(long first, long last){
        // header:
        System.out.print("\n\nTable of Collatz height and
width records\n");
        System.out.printf("\n%16s%16s%16s%16s%16s","start",
"this height", "this width",
                "max height", "max width");
        long [] answer = new long [2];
        long bigHSF = 0;
        long bigWSF = 0;
        for (long i = first; i <= last; ++i){
            answer = getData(i);
            if (answer[0] > bigHSF || answer[1] > bigWSF){
                if(answer[0] > bigHSF){
                    bigHSF = answer[0];
                }
                if(answer[1] > bigWSF){
                    bigWSF = answer[1];
                }
                System.out.printf("\n%16d%16d%16d%16d%16d",
i, answer[0], answer[1],
                        bigHSF, bigWSF);
            }
```

```
        }
    }
    public static long [] getData(long start){
        // 0: height, 1: width
        long [] retVal = new long[2];
        long current = start;
        long count = 0;
        long height = start;
        while(current >1){
            ++count;
            current = next(current);
            if (current > height){
                height = current;
            }
        }
        retVal[0] = height;
        retVal[1] = count;
        return retVal;
    }

    public static void testGetData(){
        long [] output = getData(52);
        System.out.println("height:" + output[0] + "
width: " + output[1]);
    }

    public static void testOrbit(){
        drawOrbit(9);
    }

    public static void main(String[] args) {
        System.out.println("\nCollatz Project\n");

//          test01();
//          test02();
//          testOrbit();
//          testGetData();
        biggestSoFar(1, 1_000);

        System.out.println("\n\nBye.\n\n");
    }
```

```
}
```