

Collatz.java

```
package euler;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashMap;

/**
 * Given input n, finds the term less than or equal to n that generates the longest Collatz
sequence
 * Project Euler Problem Number 14
 * @author sulliadfd
 */
public class Collatz {
    private HashMap<Long, Long> hash;
    public long N, solution;

    public Collatz(long n) {
        N=n;
        hash = new HashMap<Long, Long>();
        solution=greatestSequence();
    }

    /**
     * Finds the solution of the problem with the value N attributed to this Collatz
     * @param
     * @return long sol
     */
    private long greatestSequence() {
        long current=N, temp=1, sol=1;
        while (current!=1) {
            temp=sequence(current);
            if (sol<temp) {
                sol=temp;
                N=current;
            }
            current--;
        }
        return sol;
    }

    /**
     * Given a term n, returns the number of terms in its Collatz sequence
     * @param long n
     * @return long count
     */
    private long sequence(long n) {
        long current=n, count=1;
        while (current!=1) {
            if (!hash.containsKey(current)) {
                count++;
                if (current%2==0) current=current/2;
                else current=(3*current)+1;
            } else {
                count = count+hash.get(current);
                break;
            }
        }
    }
}
```

Collatz.java

```
    }
    add(current, count);
    return count;
}

/**
 * Adds a single element to the HashMap that drives the Collatz
 * @param long n
 * @param long count
 * @return
 */
private void add(long n, long count) {
    if(!hash.containsKey(n)) hash.put(n, count);
}

/**
 *
 * @param args
 * @return
 */
public static void main(String args[]) {
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    System.out.println("I will find the greatest sequence generated by the Collatz sequence
of all number less than your input (N):\nN=");
    long n;
    Collatz collatz;
    while(true) {
        try {
            n =Long.valueOf(br.readLine());
            break;
        } catch (IOException ioe) {
            System.out.println("Incorrect Input, try again:");
        } catch (NumberFormatException nfe) {
            System.out.println("Incorrect Input, try again:");
        }
    }
    collatz=new Collatz(n);
    System.out.println("The solution is n="+collatz.N+", which had "+collatz.solution+"
terms");
}
}
```