```java
import components.simplereader.SimpleReader;

/**
 * Put a short phrase describing the program here.
 *
 * @author Put your name here
 *
 */
public final class Hailstone4 {

    /**
     * Private constructor so this utility class cannot be instantiated.
     */
    private Hailstone4() {
    }

    /**
     * Generates and outputs the Hailstone series starting with the given
     * integer.
     *
     * @param n
     *            the starting integer
     * @param out
     *            the output stream
     */
    private static void generateSeries(int n, SimpleWriter out) {
        int count = 0;
        int max = n;
        int x = n;
        out.print(n + ", ");
        while (x != 1) {
            if (x != 1) {
                if (x % 2 == 0) { // when even
                    x /= 2;
                } else { // when odd
                    x = 3 * x + 1;
                }
                if (x == 1) {
                    out.println(x);
                } else {
                    out.print(x + ", ");
                }
            }
            count++; // length of series
            if (x > max) {
                max = x;
            }
        }
        out.println("Length of series: " + count);
        out.println("The maximum value is: " + max);
    }

    /**
     * Main method.
     *
     * @param args
     *            the command line arguments
```

```
61      */
62     public static void main(String[] args) {
63         SimpleReader in = new SimpleReader1L();
64         SimpleWriter out = new SimpleWriter1L();
65         /*
66          * Put your main program code here; it may call myMethod as shown
67          */
68         char repeat = 'y';
69         int input;
70
71         while (repeat == 'y') {
72             out.println("Enter a positive number: ");
73
74             input = in.nextInteger();
75             generateSeries(input, out);
76
77             out.println(
78                     "Would you like to calculate a new series? If so, please enter y.");
79             repeat = in.read();
80         }
81
82         /*
83          * Close input and output streams
84          */
85         in.close();
86         out.close();
87     }
88 }
89
```