Homework Turnin

Account:	6G_06 (rgalanos@fcps.edu)
Section:	6G
Course:	TJHSST APCS 2016-17
Assignment:	08-06
Receipt ID:	80f7e5e7a7ad6a62b36a32d9418094cc

Turnin Successful!

The following file(s) were received:

Hailstone_Driver.java (4518 bytes)

```
1. //name:     date:
2. import java.util.*;
3. public class Hailstone_Driver
4. {
5.    public static void main(String[] args)
6.    {
7.       Hailstone hs = new Hailstone();
8.
9.       Scanner keyboard = new Scanner(System.in);
10.      System.out.print("Enter Hailstone starting number:  ");
11.      int startNum = keyboard.nextInt();
12.      do{
13.         long startTime = System.nanoTime();
14.         hs.hailstoneMaps(startNum);
15.         int time = (int)(System.nanoTime() - startTime);
16.         System.out.println(Hailstone.steps.get(startNum) +" steps.");
17.         System.out.println(Hailstone.sequence.get(startNum));
18.         System.out.println("Nanoseconds:  "+ time);
19.         System.out.println();
20.
21.         System.out.print("Enter Hailstone starting number:  ");
22.         startNum = keyboard.nextInt();
23.      }while( startNum != -1);
24.      System.out.println("Goodbye.");
25.   }
26. }
27.
28. class Hailstone
29. {
30.   public static HashMap<Integer, Integer> steps = new HashMap<Integer, Integer>();
31.   public static HashMap<Integer, String> sequence = new HashMap<Integer, String>();
32.
33.   public int hailstoneMaps(int k)
34.   {
35.      if(sequence.containsKey(k))
36.         return steps.get(k);
37.      else
38.      {
39.         if(k==1)
40.         {
41.            steps.put(1, 0);
42.            sequence.put(1, "1");
43.            return steps.get(k);
44.         }
45.         else
46.         {
47.            if(k%2==0)
48.            {
49.               steps.put(k, 1+hailstoneMaps(k/2));
50.               sequence.put(k, k+", "+sequence.get(k/2));
```

```
            return steps.get(k);
         }
         else
         {
            steps.put(k, 1+hailstoneMaps(3*k+1));
            sequence.put(k, k+", "+sequence.get(3*k+1));
            return steps.get(k);
         }
      }
   }
}

/*********************************

 ----jGRASP exec: java Hailstone_teacher
Enter Hailstone starting number:  16
4 steps.
16, 8, 4, 2, 1
Nanoseconds:  726799

Enter Hailstone starting number:  16
4 steps.
16, 8, 4, 2, 1
Nanoseconds:  12661

Enter Hailstone starting number:  16
4 steps.
16, 8, 4, 2, 1
Nanoseconds:  6843

Enter Hailstone starting number:  8
3 steps.
8, 4, 2, 1
Nanoseconds:  6843

Enter Hailstone starting number:  5
5 steps.
5, 16, 8, 4, 2, 1
Nanoseconds:  23611

Enter Hailstone starting number:  5
5 steps.
5, 16, 8, 4, 2, 1
Nanoseconds:  6501

Enter Hailstone starting number:  199
119 steps.
199, 598, 299, 898, 449, 1348, 674, 337, 1012, 506, 253, 760, 380, 190, 95, 286, 143, 430, 215, 646, 323, 970,
Nanoseconds:  1348206

Enter Hailstone starting number:  598
118 steps.
598, 299, 898, 449, 1348, 674, 337, 1012, 506, 253, 760, 380, 190, 95, 286, 143, 430, 215, 646, 323, 970, 485,
Nanoseconds:  8212

Enter Hailstone starting number:  299
117 steps.
299, 898, 449, 1348, 674, 337, 1012, 506, 253, 760, 380, 190, 95, 286, 143, 430, 215, 646, 323, 970, 485, 1456,
Nanoseconds:  7870

Enter Hailstone starting number:  -1
Goodbye.
 ----jGRASP: operation complete.
 *********************************************/
```