AP Computer Science A Homework 6

Due date: Friday, October 14, 2016
Instructor: Mr. Alwin Tareen

Part A: The Hailstone Sequence

- In some respects, this process is reminiscent of the formation of hailstones, which get carried upward by the winds over and over again, before they finally descend to the ground. Because of this analogy, this sequence of numbers is usually called the Hailstone sequence.
- The Hailstone sequence(also known as the Collatz sequence) can be expressed as follows:
  - Pick some positive integer and call it `n`.
  - If `n` is even, divide it by two.
  - If `n` is odd, multiply it by three and add one.
  - Continue this process until `n` is equal to one.
- Mathematically, the Hailstone sequence can be expressed as:

$$n_{i+1} = \begin{cases} \frac{n_i}{2} & \text{if } n_i \text{ is even} \\ 3n_i + 1 & \text{if } n_i \text{ is odd} \end{cases}$$

- If $n_i = 1$, we stop iterating and have found the full sequence.
- For example, if we start with $n_0 = 12$, we obtain the sequence:

```
12 6 3 10 5 16 8 4 2 1
```

- This particular sequence required `9` steps to reach its termination point.

Part B: Calculating the Number of Steps Generated by each Hailstone Sequence

- Write a `Java` program in the file `Hailstone.java` that generates the Hailstone sequence for a given integer `n`, and counts the number of steps required for that sequence to reach `1`.
- The variable `n` is being read in as a parameter, which means it is available for use throughout your program. This is the integer from which your Hailstone sequence begins, and you must use this variable with your conditional statements.
- The variable `steps` has been declared as type `int`, and the number of steps required for the sequence to reach `1` must be computed, and placed in this variable.
- Verify that the program displays the following output when you run the `void main(String[] args)` method:

```
Sequence n = 387, steps = 120
Sequence n = 7316, steps = 132
Sequence n = 553, steps = 136
Sequence n = 1725, steps = 42
Sequence n = 8174, steps = 158
Sequence n = 4948, steps = 28
Sequence n = 2218, steps = 19
Sequence n = 490, steps = 22
Sequence n = 6567, steps = 75
Sequence n = 1512, steps = 109
```

- You will write your solution in a `method` called `sequenceSteps()`, right below the place where it says: `YOUR CODE HERE`.
- Compile and run the `Hailstone.java` file, and make sure that you get a correct value on the terminal output. If you have made an error, correct the mistake, and then compile and run the `Hailstone.java` file again, until you get the right answer.
- If you are confident you have a correct answer, clicked on the button labelled `Run Tests` to activate the `JUnit` test bench.
- If you see a green bar in the `BlueJ: Test Results` window, then your code is verified as correct. Otherwise, if a red bar appears, then you need to perform some additional debugging.

Part C: Submission

- Submit your `Java` program by uploading it to the `Web-CAT` automated grading platform.