

Hailstone.java

This program displays the Hailstone Sequence proposed by German mathematician Lothar Collatz. The sequence always ends in a 1, although it has yet to be proven. The sequence works this way: pick a number, if it is even then half it, if it is odd then multiply by 3 and add one. Continue this process until the number becomes one.

- In main, create an instance and use it to call findIt().
  - In findIt(), make a loop to call the other methods. The program will terminate when the user types in the number -1 (a sentinel value).
- The program will need methods for the following
- Accept the user input.
  - When printing the sequence, the program will use a loop until the sequence number 1.
    - the sequence will print 10 numbers on a line, then go to the next line.
  - Print a count for the number of times the loop had to run to make the sequence.
  - You need to decide on the use of field or local variables.

Here is a sample run:

Prompt (you need to do this part)

Enter a positive integer ( 1 - 10000 ). To quit, enter -1:     **20**

      20      10      5      16      8      4      2      1

The loop executed 8 times.

Enter a positive integer ( 1 - 10000 ). To quit, enter -1:     **30**

      30      15      46      23      70      35      106      53      160      80  
      40      20      10      5      16      8      4      2      1

The loop executed 19 times.

Enter a positive integer ( 1 - 10000 ). To quit, enter -1:     **-3**

Enter a value within the range, please!

Enter a positive integer ( 1 - 10000 ). To quit, enter -1:     **-1**

Thank you for playing Hailstone.

The following is a flowchart for the logic of the printing the sequence part (not the loop to determine if the program will terminate). This is another way to make your plan, i.e. your pseudocode.

