

Week 2 Lab Assignment: Building on Collatz

This is the assignment for the labs in Week 2 (January 14 – 18). This assignment will not be handed in or graded, but please take it seriously. It will help you with the rest of the course. If you already know Python, this assignment will be a useful review of the basics of the language. If Python is new to you, this assignment will give you a chance to work with some of the fundamentals.

Please don't skip this assignment if it seems too easy. Writing and testing a solution is always good practice.

If this assignment seems difficult, that's good! Working through it will improve your programming skills. Remember that the TA's and I are available to provide help if you need it.

Deadline: Please complete your solution by Monday January 21. If you fall behind, it's very hard to catch up. I will post my solution on January 21, but please don't look at it until you have worked through the lab and completed your own solution. In terms of learning, looking at my solution is a very poor substitute for creating your own.

Introduction:

```
Start with any positive integer n
while n > 1
    if n is even
        n = n/2
    else
        n = 3*n + 1
```

Collatz conjectured that this process always reaches a point where $n = 1$

We developed a Python program in class that reads an integer n from the user, then for each integer from 1 up to n , prints the sequence of numbers generated by the Collatz procedure. This program is posted on the course notes from January 11.

Your task is to add some features to the program.

Details:

1. Download the Python program and make sure it runs.
2. Modify the program so that it keeps track of the largest integer generated during its

execution. For example, if the user-supplied integer is 5, the sequences generated are

```
1
2, 1
3, 10, 5, 16, 8, 4, 2, 1
4, 2, 1
5, 16, 8, 4, 2, 1
```

The largest integer generated is 16

3. Modify the program so that it keeps track of the longest sequence generated during its execution. For example, if the user-supplied integer is 5, the longest sequence is 3, 10, 5, 16, 8, 4, 2, 1 which has length 8
4. Modify the program so that it gets an integer n from the user, then finds the smallest integer that produces a Collatz sequence that contains an integer $\geq n$. For example, if the user supplies the value 50, your program should print the value 7 because the sequence for 7 is 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1
5. Modify the program so that it gets an integer n from the user, then finds the smallest integer that produces a Collatz sequence of length $\geq n$. For example if the user supplies the value 6, your program should print the value 3.

Tasks 4 and 5 may make your program take a long time to complete, if you input a large value of n . You will probably discover that your program will run faster if you don't print the sequences that are generated. Output is one of the slowest things your program does.

Dealing with Errors: In the real world, the program would need to be robust enough to not crash if the user doesn't enter a numeric value. For this assignment you can assume that the user will not do this.

Style: Good programming style is essential to writing programs that others can understand. Good style includes (but is not limited to) effective use of comments, good layout on the page, good variable names, and clear flow of events within your program. Try to get into the habit of writing clear code.