CSCE 110: Programming I

Lab #4 – Extra Credit (15 points)

Due: Sunday, September 25th by 11:59pm

Directions. Take your time and make sure you understand everything in this lab before getting started. Also, make sure your programs match the output EXACTLY as given for each question. This is important as one of the keys to being a good programmer is attention to details.

1 Please make sure you understand the following.

For this assignment, you are only allowed to use what we have discussed in the first 4 weeks of class. If we haven't discussed it in class, you cannot use it in your program. Have fun!
You must attend your lab on Wednesday (Sept. 21st) or Thursday (Sept. 22nd) to receive extra credit.

2 Extra Credit Problem

The $3n + 1$ *problem.* Write a Python program (called q4.py) that generates a sequence of numbers based on the following procedure. Start with an integer $n \geq 1$. If n is even, divide by 2. If n is odd, multiply by 3 and add 1. Repeat this process with the new value of n, terminating when $n = 1$. For example, the following sequence of numbers will be generated for $n = 22$.

$$22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1$$

Your program will output (i) the sequence of numbers generated, (ii) the number of steps to generate the sequence, and (iii) the largest number in the sequence. For $n = 22$, there are 15 steps and 52 is the largest number in the sequence.

Programming tips. Consider writing your program to generate the appropriate sequence for a given integer n without worrying about making your output follow as shown in the examples below. Once your program generates the correct sequence, then add the necessary code to make your output match exactly as shown in the examples.

Example #1. At the prompt, the user enters the number 22 (line 1). The program then outputs the numbered steps of the procedure (lines 3–17). For example, line 3 shows that in the first step, 22 is even and divided in half to get 11. Step 2 then takes 11, which is odd, and multiplies it by 3 and adds 1 to get 34 (line 4). The remaining steps of the procedure follow similarly. After the last step of the procedure (line 17), the total number of steps and the largest number in the sequence are printed (lines 19–20).

```
1  Enter a number: 22
2
3  1. 22 is even, divide in half: 11
4  2. 11 is odd, multiply by 3 and add 1: 34
5  3. 34 is even, divide in half: 17
6  4. 17 is odd, multiply by 3 and add 1: 52
7  5. 52 is even, divide in half: 26
8  6. 26 is even, divide in half: 13
9  7. 13 is odd, multiply by 3 and add 1: 40
10 8. 40 is even, divide in half: 20
11 9. 20 is even, divide in half: 10
12 10. 10 is even, divide in half: 5
13 11. 5 is odd, multiply by 3 and add 1: 16
14 12. 16 is even, divide in half: 8
15 13. 8 is even, divide in half: 4
16 14. 4 is even, divide in half: 2
17 15. 2 is even, divide in half: 1
18
19 Number of steps: 15
20 Largest number in sequence: 52
```

Example #2. Here, the user tries the procedure on the number 304.

```
1  Enter a number: 304
2
3  1. 304 is even, divide in half: 152
4  2. 152 is even, divide in half: 76
5  3. 76 is even, divide in half: 38
6  4. 38 is even, divide in half: 19
7  5. 19 is odd, multiply by 3 and add 1: 58
8  6. 58 is even, divide in half: 29
9  7. 29 is odd, multiply by 3 and add 1: 88
10 8. 88 is even, divide in half: 44
11 9. 44 is even, divide in half: 22
12 10. 22 is even, divide in half: 11
13 11. 11 is odd, multiply by 3 and add 1: 34
14 12. 34 is even, divide in half: 17
15 13. 17 is odd, multiply by 3 and add 1: 52
16 14. 52 is even, divide in half: 26
```

```
15. 26 is even, divide in half: 13
16. 13 is odd, multiply by 3 and add 1: 40
17. 40 is even, divide in half: 20
18. 20 is even, divide in half: 10
19. 10 is even, divide in half: 5
20. 5 is odd, multiply by 3 and add 1: 16
21. 16 is even, divide in half: 8
22. 8 is even, divide in half: 4
23. 4 is even, divide in half: 2
24. 2 is even, divide in half: 1

Number of steps: 24
Largest number in sequence: 304
```

Example #3. The user tests our program's behavior on the number 1, the smallest possible legal input for this program.

```
Enter a number: 1

Number of steps: 0
Largest number in sequence: 1
```

Final thoughts. If you find this sequence interesting, don't be afraid to experiment. Can you find any numbers that won't eventually lead to 1?