# Python Tres Caballeros Collatz Conjecture

Time required: 120 minutes

- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

## Pseudocode

1. Write pseudocode or TODO for the exercise
2. Submit with the assignment

## Scenario

In 1937, a German mathematician named Lothar Collatz formulated an intriguing hypothesis (it remains unproven) which can be described in the following way:

1. Take any non-negative and non-zero integer number and name it num;
2. If it's even, evaluate a new num as num // 2 (Integer division)
3. Otherwise, if it's odd, evaluate a new num as (3 × num) + 1
4. If num ≠ 1, skip to step 2.

The hypothesis says that regardless of the initial value of num, it will always go to 1.

It's an extremely complex task to use a computer to prove the hypothesis for any natural number (it may even require artificial intelligence), but you can use Python, Java, and C++ to check some individual numbers. Maybe you'll even find the one which would disprove the hypothesis.

## Requirements

You can complete this program with OOP or functions.

1. Create a program in Python named **collatz_main.py**
2. Create a separate OOP class file with a collatz calculation method called **collatz_class.py**

   a. This class method will take in an integer as an argument.

   b. The method will calculate the sequence and store it in a list.

   c. The list will be returned to the main application.

3. Input will take place in the main application.

4. Display will take place in the main application.

## Program Operation

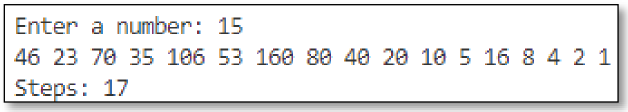
1. Ask the user to enter a natural number.

2. Calculate the Collatz sequence.

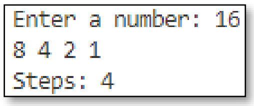
3. Display the input and output as shown.

4. Ask the user if they wish to continue or exit.

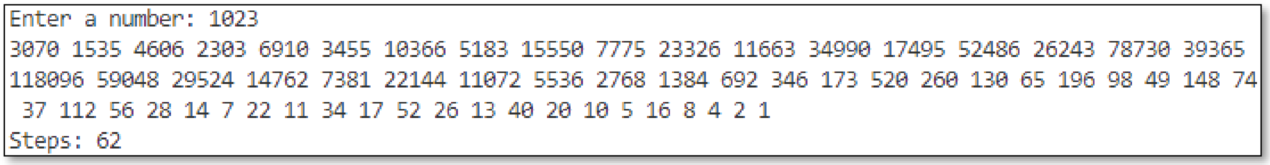
Example runs:



```
Enter a number: 15
46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
Steps: 17
```



```
Enter a number: 16
8 4 2 1
Steps: 4
```



```
Enter a number: 1023
3070 1535 4606 2303 6910 3455 10366 5183 15550 7775 23326 11663 34990 17495 52486 26243 78730 39365
118096 59048 29524 14762 7381 22144 11072 5536 2768 1384 692 346 173 520 260 130 65 196 98 49 148 74
 37 112 56 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Steps: 62
```

## Challenge

Which starting number, under one million, produces the longest chain?

## Assignment Submission

1. Attach the pseudocode or use TODO’s.

2. Attach the program files.

3. Attach screenshots showing the successful operation of the program.

4. Submit in Blackboard.