Java Tres Caballeros Collatz Conjecture

Time required: 90 minutes

Please read the directions carefully before beginning the assignment.

- 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 or TODO

1. Write pseudocode or TODO for the exercise.
2. Comment your code to show evidence of understanding.

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 c0;
2. If it's even, evaluate a new c0 as c0 // 2 (Integer division)
3. Otherwise, if it's odd, evaluate a new c0 as (3 × c0) + 1
4. If c0 ≠ 1, skip to step 2.

The hypothesis says that regardless of the initial value of c0, 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

1. Create a program in Java.
2. Create a separate OOP class file with a collatz calculation method.
    a. This class method will take in an int as an argument.

    b. The method will calculate the sequence and store it in an ArrayList<Integer>.

    c. The ArrayList<Integer> will be returned to the application.

3. Input will take place in the main application.
4. Display will take place in the main application.
5. Ask the user to enter a natural number.
6. Calculate the Collatz sequence.
7. Display the input and output as shown.
8. Ask the user if they wish to continue or exit.

Return an ArrayList from a Method

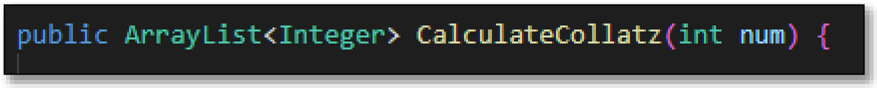
Returning an ArrayList from a method is just a little different than returning a regular variable. The difference is how you declare the return data type.

long is an integer data type with enough room to calculate big, big numbers.

Return data type: **ArrayList<long>**

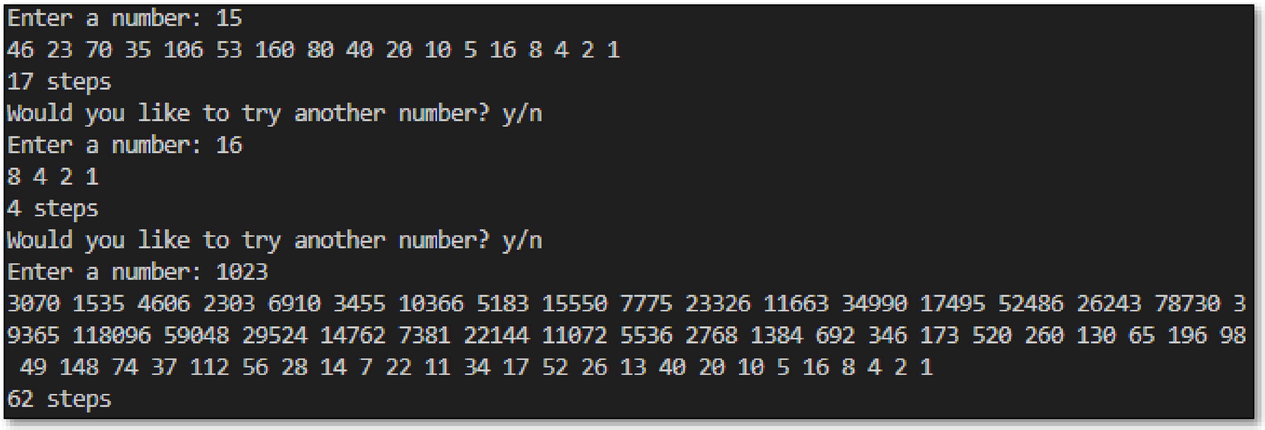
Method name: **CalculateCollatz()**

Parameter: **long num**



```
public ArrayList<Integer> CalculateCollatz(int num) {
```

Example runs:



```
Enter a number: 15
46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
17 steps
Would you like to try another number? y/n
Enter a number: 16
8 4 2 1
4 steps
Would you like to try another number? y/n
Enter a number: 1023
3070 1535 4606 2303 6910 3455 10366 5183 15550 7775 23326 11663 34990 17495 52486 26243 78730 3
9365 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
62 steps
```

Challenge

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

You will need to change your data types for this to work. Int doesn’t have enough for the calculation.

int = long

Integer = Long

Assignment Submission

1. Use pseudocode or TODO.
2. Comment your code to show evidence of understanding.
3. Attach the program files.
4. Attach screenshots showing the successful operation of the program.
5. Submit in Blackboard.