Tres Caballeros Collatz Conjecture Calculator

 No AI use

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

The hypothesis says that regardless of the initial value of n, 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. Outline the program with pseudocode and comments.
    a. I want to see your work and that you understand the program.

b. I want to see that you did not use AI.

2. Create a program in MATLAB named: **CollatzCalculator.m**
3. Create a function in a separate file named: **CalculateCollatz.m**
    a. The function will calculate the sequence and store it in a vector.
    b. The vector will be returned to the main method.
    c. This is the function signature that will return the sequence and iteration count.

```
function [sequence, count] = CollatzCalculator(number)
```

4. Input will take place in the main application.
    a. This input will be passed to the function.
    b. The function will return a vector and iteration count to the main application.

```
% Call the CollatzCalculator function to get the sequence
[sequence, count] = CollatzCalculator(number);
```

5. In the main application, use input validation to ensure the user enters only a non negative and non zero integer. Here are some built in functions to help validate input.
    a. **isnumeric(n)**
        i. **Purpose:** Checks if n is a numeric value.
        ii. **Explanation:** This function returns true if n is a number (e.g., integer, floating-point number) and false otherwise. This ensures that the input is a number and not a string or other data type.
    b. **floor(n) == n**
        i. Purpose: Checks if n is an integer.
        ii. Explanation: The floor function rounds n down to the nearest integer. If n is already an integer, floor(n) will be equal to n. This condition returns true if n is an integer and false if n is a floating-point number with a fractional part.
6. User interaction and display will take place in the main application.

7. Ask the user to enter a natural number.
8. Calculate the Collatz sequence.
9. Calculate how many steps it took.
10. Display the input and output as shown.

Example runs:

```
Welcome to the Collatz Sequence Calculator!
Please enter a positive integer: 15
The Collatz sequence is:
46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
Number of iterations: 17
```

```
Welcome to the Collatz Sequence Calculator!
Please enter a positive integer: 17
The Collatz sequence is:
52 26 13 40 20 10 5 16 8 4 2 1
Number of iterations: 12
```

```
Welcome to the Collatz Sequence Calculator!
Please enter a positive integer: 16
The Collatz sequence is:
8 4 2 1
Number of iterations: 4
```

I copied and pasted the results into Notepad, as the answer was to long to screenshot in MATLAB.

```
Welcome to the Collatz Sequence Calculator!
Please enter a positive integer: 1023
The Collatz sequence is:
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
Number of iterations: 62
```

Challenge

This has been solved by a previous student.

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

Assignment Submission

1. Attach the pseudocode.
2. Attach the program files.
3. Attach screenshots showing the successful operation of the program.
4. Submit in Blackboard.