

Problem Set 2 Exercise #10: Collatz Problem

Reference: Lecture 4 notes

Learning objective: Repetition statements

Estimated completion time: 20 minutes

Problem statement:

The Collatz problem is named after Lothar Collatz who first proposed it in 1937. It states the following:

Take any natural number n . If n is even, divide it by 2 to get $n/2$; if n is odd, triple it and add 1 to obtain $3*n + 1$ (see mathematical expression below). Repeat the process indefinitely. No matter what number you start with, you will *always* eventually reach 1.

$$f(n) = \begin{cases} \frac{n}{2}, & \text{if } n \text{ is even} \\ 3 * n + 1, & \text{if } n \text{ is odd} \end{cases}$$

You are not required to prove the Collatz conjecture, but to write a program **PS2_Ex10_Collatz.java** that reads in a positive integer and determines how many iterations it takes to reach 1.

For example, if n is 3, then the answer would be 7 (iterations), as

$$3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1.$$

Your program should contain a method

```
int countIterations(int n)
```

to count the number of iterations required for the value n to reach 1.

Sample run #1:

```
Enter a natural number: 3
Number of iterations = 7
```

Sample run #2:

```
Enter a natural number: 1
Number of iterations = 0
```