Problem Set 2 Exercise #09: Collatz Problem

Reference: Lecture 5 notes

Learning objective: Repetition statements

Estimated completion time: 25 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 your start with, you will *always* eventually reach 1.

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

You are not required to prove the Collatz conjecture, but to write a program **collatz.c** 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 → 10 → 5 → 16 → 8 → 4 → 2 → 1.

Your program should contain a function

```
int count_iterations(int n)
```

to compute 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
```