CSE 13S Winter Quarter 2022
Assignment 1: Getting Acquainted with UNIX and C

Description of the Program:

The goal of this assignment is to familiarize ourselves with bash and how to create shell scripts. Given a collatz sequence, our shell script is set out to plot three different graphs. The first graph plots the length of the collatz sequence given the starting value, while the second graph plots the highest value reached in the collatz sequence given the starting value. The last graph is different, as it plots a histogram by length of the collatz sequence and number of collatz sequences.

Files to be included in the “asgn1” directory

- plot.sh
    - This is the bash script that we are to make and contains the necessary code in order to display the three different graphs.
- collatz.c
    - This is the code given to us to simulate the collatz sequence in c
- Makefile
    - This is also given to us and is used to help compile our code.
- README.md
    - This is a file written in markdown format to explain the purpose of the code and how to build it, in addition to any errors we found.
- DESIGN.pdf
    - Gives insight into some of the processes of coming up with the code, along with the purpose of the assignment.

Pseudocode:

for figures 1 and 2
- for i in range(1, 10000):
    - get the length of the collatz sequence
    - get the highest value of the collatz sequence
- plot length on gnuplot
- plot highest value on gnuplot

for figure 3
- for in in range(1, 10000):
    - get the length of the collatz sequence
    - append to a length array
    - sort the length array
        - for loop through lengths

        - use wc -l for each to find the number of occurrences of each length
    - plot length and occurrences on gnuplot as a histogram

Notes about the pseudocode

- for i in range(1, 10000) specifies for us to use collatz sequences starting at starting points 1 to 10000
- getting the length of the collatz sequence is as easy as
    - ./collatz -n $i | wc -l
    - to append we can append to the /temp/length.dat by doing
    - ./collatz -n $i | wc -l >> /temp/length.dat
- to get the highest value
    - ./collatz -n $i | sort -n | tail -n 1
    - and to append like before
    - ./collatz -n $i | sort -n | tail -n 1 >> temp/val.dat
- we can use both of these ideas and methods to complete the third figure as well

Credit:

- I watched the recording of Eugene's section and drew a lot of inspiration from that
- I knew how I would have done it in python, however script was new language to me and his explanations bridged the gaps between learning the methods to learning from examples.