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 and append to length.dat
    - get the highest value of the collatz sequence and append to max.dat

for figure 3
- sort length.dat and find unique lengths and their frequencies and store in his.dat

plot the length graph with length.dat
plot the max value graph with max.dat
plot the histogram with his.dat

Notes about the pseudocode

- for i in range(1, 10000) specifies for us to use collatz sequences starting at starting points 1 to 10,000
- getting the length of the collatz sequence is as easy as
    - ./collatz -n $i | wc -l
        - i being the iterator from the for loop
    - to append we can append to the /temp/length.dat by doing
    - ./collatz -n $i | wc -l >> /tmp/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 >> /tmp/val.dat
- to get frequencies of lengths
    - we first sorted and then used | uniq -c | less
    - we used head -n to redirect the output into /tmp/his.dat, rather than appending value by value
        - we did not append, since the output of our uniq -c command was an entire list rather than simply a value or data point like before
- to plot the graphs
    - we used commands given to us in the plot.sh file
        - set terminal pdf
        - set output “name of file.pdf”
        - set title “title of graph”
        - set xlabel “xlabel”
        - set ylabel “ylabel”
        - set xrange or yrange if necessary
        - set zeroaxis
        - plot “data” with dots or impulses with title ””

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 a new language to me and his explanations bridged the gaps between learning the methods to learning from examples.
- Additionally, the bit about head in both the asgn1 doc and the lecture helped me in redirected output in my histogram graph.
- The format of my DESIGN.pdf and WRITEUP.pdf drew inspiration from Audrey Ostrom, as she shared her DESIGN.pdf and WRITEUP.pdf from a different quarter and unrelated assignments on discord.