Assignment 1; Intro, C and Bash
Nathaniel Brown
9/24/2023

I. Implementation Success:

A. C program

a. The factorial function and validity function’s were implemented as planned with minimal issues.
b. Bubble sort method was a simple and effective way to handle ordering the output in ascending order.

B. Bash script

a. Setting the file name, and compiling the C script using the CLAs from the bash script were simple and quick to get rolling using the information from the previous week’s lab.

II. Implementation Changes and Rationale:

A. C program

a. This 53 year old language does not have built-in boolean's. Rather than returning a true or false with my validity check, it was done using 0/1. Which is effectively the same thing, but not initially planned for.
b. Originally I intended to print the errors to standard output, however this is not the appropriate way to handle errors in C. Instead I shifted to using fprintf to print the errors directly to standard error. This fulfilled the specifications of the autograder.

B. Bash script

a. Similar to part II.A.b from above; initially I had been using echo to print the errors into the output of the bash script. However, I eventually discover that you need to use >&2 to have the errors sent to standard error. This helped satisfy the demands of the autograder.

b. Contorting the error messages in bash to properly account for the filename being the first argument was initially tricky. I had first attempted to rework the C file to create an alternative path of running if initiated from Bash. Ultimately this was scrapped after realizing I have monkey brain. Instead within the for loop that iterates through the arguments to validate them, I simply adjusted the starting position to start at i = 2, instead of i = 1. Effectively skipping the first element (the filename). Followed by simply adjusting the error echos themselves to properly reflect ‘second’ instead of ‘first’, and ‘third’ instead of ‘second’, and so on. This was the pinnacle of all issues in development.

III. Research Progress

A. C program

a. Not much research was needed for the C file.

b. Ultimately came to the realization that C didn’t have booleans, but was handled with a simple workaround using 0 and 1 for false/true conditionals. (Basically the same thing).

B. Bash script

a. The syntax of $, #, @, -le, -lt. Were all initially foreign to me. They seem very unique of bash, as I haven’t seen them much at all in Rust, C++, or Python. This was the largest barrier to overcome at first. After some simple reading it started to make sense and writing the simple conditionals I had planned became much more achievable.

b. Lastly, simply researching how to properly echo error messages, not traditional echos was vital to the completion of the project. This was done using the >&2 signifier after the echo statements.