Alan Tort

7/2/2021

Reflection for Project 3a

Understanding:

This problem asked us to implement a solution that can only be solved by using a combination of loops and conditional statements. I knew the project reflected what was to be learned in the modules, so it was critical that I had to fully understand how to properly implement loops and conditional statements together to be successful. I studied the modules very closely to get a good feel of how to combine the two.

Testing Plan:

I sought out to make a comprehensive testing plan that covered all the of the edge cases I could think of. The amount of tests I wrote definitely enabled me to spot the biggest problem I had, which was related to the minimum and maximum valuables not accounting for negative integers. Negative numbers were indeed the hardest because my initial implementation was not properly comparing the next number to the previous.

Design:

The pseudo-code really just followed my own logic of the problem from start to finish. I modified the design several times because I knew the pseudo-code had to properly account for all the tests that I had written so I actually didn't finalize the pseudo-code until I had a working program. I wanted my pseudo-code to properly represent my program's flow of logic.

Implementation:

The biggest problem I encountered during the implementation was that I couldn't understand at first how to calculate both the maximum and minimum at the start. This caused all negative inputs to not be properly compared so I parsed through every line of code to identify the problem with Pycharm's debugger. Once I understood that the first number entered by the user will be set to both the maximum and the minimum, I understood that I needed to compare the next number to the previous, thus I needed a loop that started from the next number (and not the initial number) that sequentially checked each next number to the previous until the last number. This subsequently enabled me to fix the problem that I was running into with my negative input test.

Improvement:

If I can find a way to simplify the code in any way, I would be very happy. I have thought about how I would go about accomplishing that task that but unfortunately haven't come up with anything different. I believe that the idea that the problem when broken down to a pseudo-code level flows as well as it does, allows one to see the true simplicity of the problem. I will take a look back at this problem later in the summer session to see if my newfound knowledge can permit me to write a more efficient program.