Katrine Chow
CS161 Fall 2017

Assignment 3a Reflection

For Assignment 3a, I might have underestimated the logical flow required to correctly compare and sort user-entered values. I originally thought I can run a simple loop with a pair of if-else statement, but the actual coding showed me I needed to place more control in the sorting process.

Example: In my pseudocode I accounted for the scenario where user would only input 1 number. I immediately assign the value to both my min and max variables. However, during the actual coding I realized that the min and max variables were automatically initialized to zero. This means that when I entered all negative integers, the max is always 0 even if 0 was never entered by the user. The same happened for the min value if I entered all positive integers > 0. This took me awhile to fix, but I understand how that I need to be more careful with initializing my variables.

I also thought I could get by with only two int variables: min and max. However, I ended up adding a temp variable to not overwrite correct sorts during my loop. Not doing this caused the program to display the most recent min/max values instead of the correct min/max values.

I ended up using an if-if-else if combo to control for user inputting only 1 integer, and explicitly sorting into min and max variables after each comparison.

I also had some trouble coming with the logic when I used a while loop. After a few hours I decided to start over and tried a for loop instead. For some reason this helped me see the logic flow more clearly.

Another testing technique I learned is to make sure to test edge cases. I originally didn't think about this as I saw the assignment and thought it'd be a very simple program. However, I was reminded that it is good practice to test close to the max and min values of an int. My test values are all very small, which cannot accurately depict what the program would do as it reaches certain boundary conditions.

Lessons Learned:

- Be explicit with what I want the computer to do. Even though C++ is a high-level language, computers cannot read my mind… yet.
- Be careful with initializing variables. If I don't initialize, ints will likely be zero. This can affect test cases that involve all negative or all positive inputs.
- Test edge cases. Don't assume user will only input small values or values close to specs.

- Don't underestimate assignments. Ever.