Ethan Dunham

Dunhamet

10/10/16

Reflection 3.a

Understanding: What did you learn about the problem as you went? Why or how did you learn it?

The understanding of the program came while reading through the problem and thinking about where I would need loops and how I would have the program save the results of the loop. I also had to think about never ending loops and think about how to stop them from happening. I learned about most of these issues while reading through the text book and thinking about the problem I was asked to solve.

Testing plan: What tests didn't work out the way you expected? What alterations did you have to make to your program due to failed tests? How could your planned tests have been more complete?

Tests where the user asked to input a negative or 0 integers caused an endless loop in my program. I had not foreseen this happening and did not find the issue until I built and tried to run the program. I had to put in a verification loop at the beginning to verify that the user stated that they wanted to put in 1 or more integers. My planned tests could have been more complete if I had thought of this test prior to writing the program and testing it. I did not think about testing negative or 0 inputs because it seemed crazy that a user would want to put in negative or 0 numbers. I went back and added the 0 and negative inputs into a test after noticing the issue and attempting a patch.

Design: What was missing or needed to be altered from your initial design, and why?

I had to add a verification loop to make sure that the user didn't try to put in 0 or negative integers for the "How many integers would you like to enter." While negative integers would be absurd, 0 integers is a feasible possibility that a user could enter. I chose to use a loop to solve both issues at the same time.

Implementation: What problems did you encounter during implementation? How were you able to solve those problems? What outside sources (sites, books, or other materials) did you find helpful?

I only ran into the problem of the "how many integers?" as previously stated. I used the textbook sections on verifying user input to fix that issue. I find that every issue/question I run into, the textbook has an answer somewhere as to why it is happening.

Improvement: How can you generalize any parts of your problem solving experience in a way that might help you on future assignments?

Make sure that I take a few minutes to think like a "dumb" user and patch issues that shouldn't be an issue. It is the programmers job to anticipate issues with the program and patch them. I'm just glad that I found the issue before believing the program was complete.