Vending Machine Algorithm

CPT 104

Anthony Frazier

Being tasked with designing an algorithm to dispense change, in the least amount of coins possible, proved to be more difficult than I originally thought. I needed several variables including : itemCost (cost of the item being purchased), givenCost (money provided by the user), change (total amount of change to be provided, if any), changeNum (equal to change, but a separate variable to perform math on so that I can retain my change variable as an ouput at the end of the algorithm), and coins (quarters, dimes, nickels). After obtaining my non-zero, non-negative number inputs from the user, and insuring that the givenCost (money provided) was at least enough to equal the itemCost, I decided to go with a nested loop solution to make change.

I initially wanted dispense change exactly, but changed the algorithm upon teacher request to not dispense pennies as change. I decided to round any change to be provided under $0.05 to the nearest nickel. I had some issues with Raptor when I attempted to start rounding (0.04 and 0.03 being rounded up, while 0.02 and 0.01 being rounded down). The flowchart program got stuck in an infinite loop when checking my 0.05 and 0.04 conditions, so I was forced to make adjustments to all my change conditions. I figured out that if I used a check condition of 0.049 instead of 0.05, Raptor would no longer get stuck in a loop. After changing all of my rounding conditions, I decided to complete the same changes to all of my conditions, with necessary documentation, for the sake of program integrity.

This project would have greatly benefited from being a team assignment. Having someone to come to when encountering a problem would have saved a lot of time, and the overall flowchart of the program could have been done much faster with several members brainstorming at the same time. If someone was paying a team to do this, I believe a talented team of individuals could easily finish this program in half or even one-fourth of the time that it would take one person. Cost-wise, if someone was paying a team of coders to do this job, it would be done much quicker than paying a single person to do the task.

If I were to personally code this program, I would use the C++ programming language. I have some experience with the language, so I would not have to learn quite as much as with some other languages. The bulk of the program is computational, only requiring a few inputs from the user. Designing the input boxes would be cumbersome in C++ as opposed to an object-oriented language like Visual Basic, but I believe the efficiency and lower memory footprint of C++ more than make up for this flaw.