Planning Document – Nicolas Sanchez Assignment 0: Making Smoothies

Planning Phase

The planning phase for this program didn't really take as much time as the previous assignment mostly because it consisted of just reading the assignment pdf and really understanding the program and its function prototypes.

The function prototypes really helped when planning because when I first began writing the program I had written all the function headers and noted out the ones I hadn't gotten to. So my process was just writing each function in the order they were going to be called. It was like a to-do list. The input guide was also extremely helpful for understanding what each function did in a way where you could see the inputs specifically.

Assistance Received

I received some assistance from the TAs in the UCF CS discord and Caleb Curry on youtube. He has a few videos working with structures that were very helpful.

Debugging Phase

My debugging phase consisted of 3 specific things: Not understanding structures and their syntax, improper calculations, and a debug method causing bugs….Not understanding structures was quickly fixed by just doing some research on how to use them, which quickly sparked my memory. After this, the rest of my issues happened in calculateOrder. The first issue was within the second for loop. I was indexing my order list (amtOfEachItem) incorrectly. Instead of indexing by the recipe item list item id, I was indexing with the loop variable. This obviously didn't work because where is that even indexing to. After this fix and writing up the equation I set up some print statements to display my calculations and make sure everything was working correctly, and it was about half correct. Originally my program would print out the weight for each ingredient but wouldn't add the weight of matching ingredients, it would instead print something twice. For example, a smoothie recipe might use multiple of one ingredient. So if you took the output of my program and took the sum of items with matching id's, then you would get the correct numbers. Once I realized what was happening I just set the "=" to a "+=" in order to combine certain values instead of storing them again. This problem also caused a seg fault because the loop was eventually trying to access memory that didn't exist because my malloc ran out of space. It was filling up because repeated values weren't being stored together but instead separately. Once I fixed this I wanted to create a print statement that displayed the proper outputs. I did this by creating a for loop within the function, which caused a bunch of problems. Because I was trying to print the values within the function itself and not in main where the loop was, I was getting another seg fault. I was trying to access memory outside the malloc and also values that didn't even exist yet. I spent a lot of time trying to figure out if this seg fault was involved with my program. This wasted a lot of time but I then realized it was just the weird print. I ignored it hoping that when I created printOrder I wouldn't get a seg fault because the memory would be accessed properly, and it did so that was nice. After that, I finished up the last 3 functions, printOrder, and the frees, with no issues.

Testing Phase

My testing phase was fairly simple, nothing crazy. I just made up some ingredients and smoothie recipes and did the math myself for each one just to confirm my calculations were correct, and they were. I didn't make any weird test cases that targeted a specific thing because I really just couldn't think of any, so my test cases are very normal.

After making my code look nice and removing any nonsense whitespace, I went through my program like usual and traced it all to reinforce my understanding of my own code. I confused myself a few times because looking at some of the lines involving accessing certain members in a struct are very long and confusing, but I just took my time with it and it made sense. After this, I just did a final test to make sure everything was working fine.