Hunter Goodwin

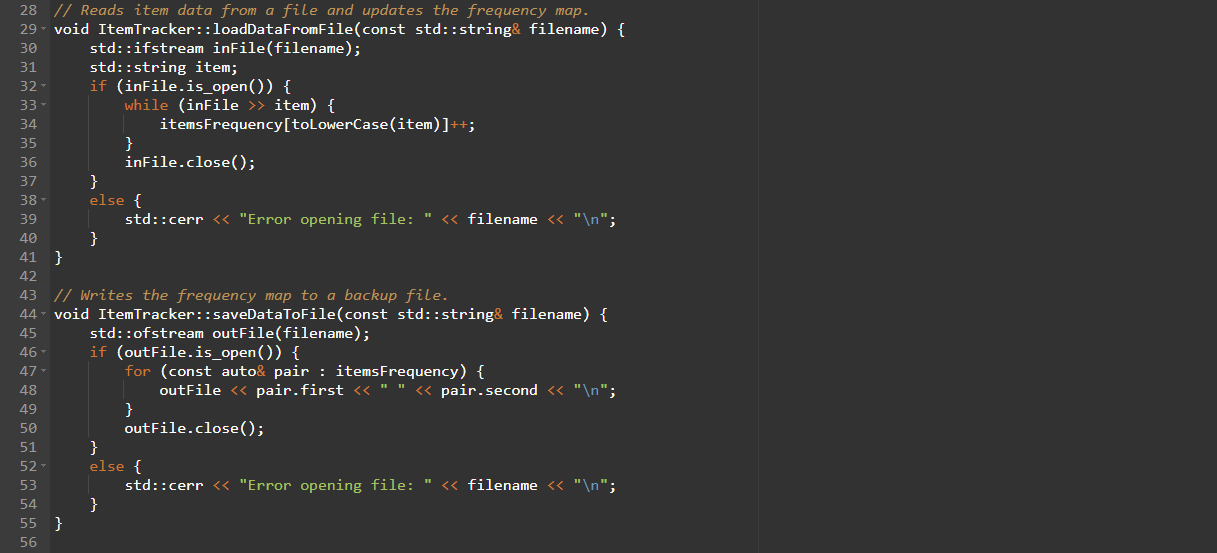
April 20, 2025

CS 210 7-3

**Corner Grocer Item Tracking Program**

My code’s design and functionality were focused on reading a given file and presenting that material in a clear and concise manner. This program is designed to help track the frequency of items purchased at a grocery store. It reads data from a text file, processes the information, and provides users with a menu to interact with the data. Users can search for specific items, see a list of all items and their frequencies, or view a histogram that visually represents the data.

The program uses a class called `ItemTracker` to handle all the logic for tracking and displaying item frequencies. It reads data from an input file (`CS210\_Project\_Three\_Input\_File.txt`) and saves the results to an output file (`frequency.dat`).

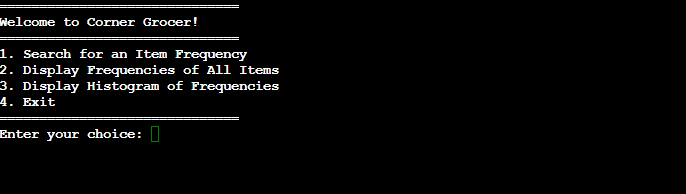


Our key features are:

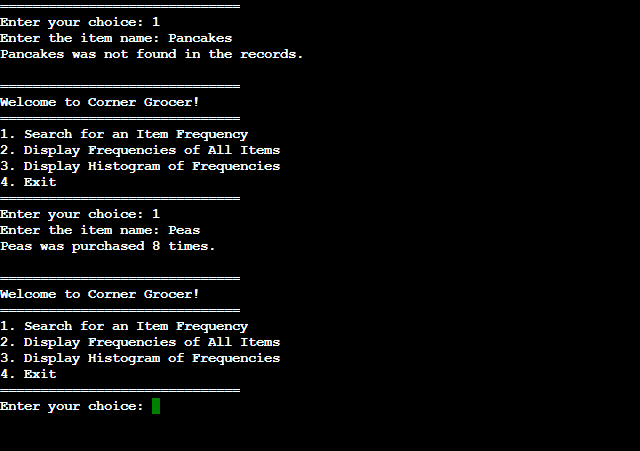
* Search for an Item: Users can input an item name, and the program will display how many times that item was purchased. If the item isn't found, it will let the user know.
* Display All Frequencies: The program can list all the items and how many times each one was purchased.
* Histogram: This feature creates a visual representation of the purchases using asterisks (`\*`). Each asterisk corresponds to one purchase of the item.

The program is built around the `ItemTracker` class, which uses a map (`itemsFrequency`) to store the data. Here are some of the important methods:

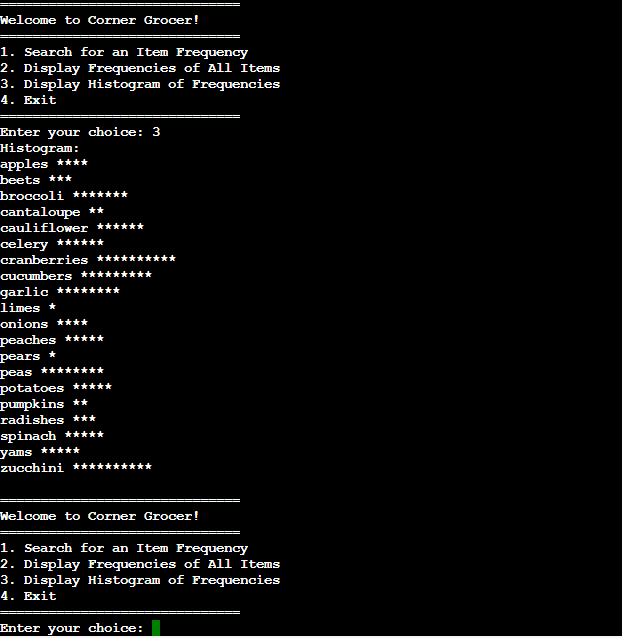
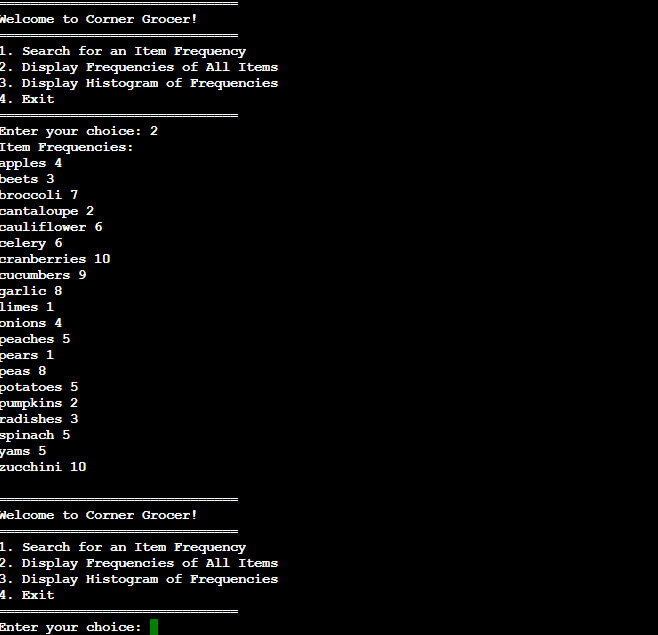
* loadDataFromFile(): Reads the input file, processes each line, and updates the itemsFrequency map. It ensures that capitalization differences (e.g., "Apple" vs "apple") don't cause duplicates.
* saveDataToFile(): Writes the data from `itemsFrequency` to an output file (`frequency.dat`) for backup or further analysis.
* getItemFrequency(): Lets the user search for a specific item and returns its frequency.
* displayAllFrequencies(): Prints all the items and their frequencies in a clean list format.
* displayHistogram(): Prints a histogram where each item is followed by a row of asterisks that represent its frequency.

Here is an example of how those outputs may look: 

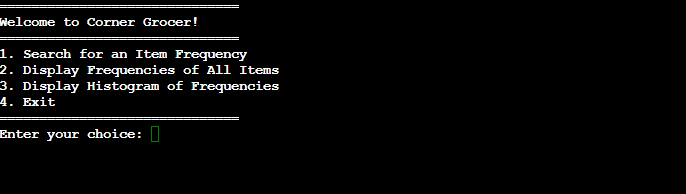
*Corner Grocer Main Menu*



*Choice 1 with an unlisted item followed by choice 1 with a listed item*



*Options 2 & 3 and their respective differences*



*Option 4 to leave our program*

Overall, this program should remain highly adaptable as it simply reads the items listed on the .txt file regardless of what their names are. Therefore, it should be very easy to incorporate new items into our list.