Corner Grocer Item-Tracking Program Documentation

The Corner Grocer item-tracking program, developed in C++ using Visual Studio, analyzes grocery purchase records to assist in optimizing produce section layouts. The program reads items from CS210\_Project\_Three\_Input\_File.txt, processes them, and provides frequency analysis through a user-friendly menu interface.

The program’s core is the GroceryTracker class, which uses a std:: map to store item names and their frequencies, ensuring O(log n) lookup efficiency. Private methods handle file operations: loadInputFile reads the input file and normalizes item names to lowercase for case-insensitive processing, while createBackupFile generates frequency.dat for data backup. Public methods implement the menu options: searchItemFrequency for item lookup, printFrequencyList for displaying all frequencies, and printHistogram for a text-based histogram. The main function drives the program with a menu loop, supporting four options: search, list frequencies, show histogram, and exit. Input validation ensures robust handling of user inputs, such as alphabetic item names and numeric menu choices.

Industry-standard practices enhance code quality. In-line comments clarify functionality (e.g., // Reads input file and populates itemFrequencies map), and camelCase naming (e.g., itemFrequencies) ensures readability. The program efficiently processes 20 unique items from a 100-item input file, producing outputs like frequency lists and histograms (Figures 1–3). The backup file frequency.dat stores results for future use (Figure 4). Challenges included ensuring case-insensitive searches and robust file handling, addressed through std:: transform and error checks.

The design leverages C++’s performance and standard library. The std::map was chosen for its efficiency, and input validation improves user experience. This implementation meets all functional requirements while maintaining clarity and maintainability.

Figure 1: Menu Prompt  
A screenshot of a computer

AI-generated content may be incorrect.

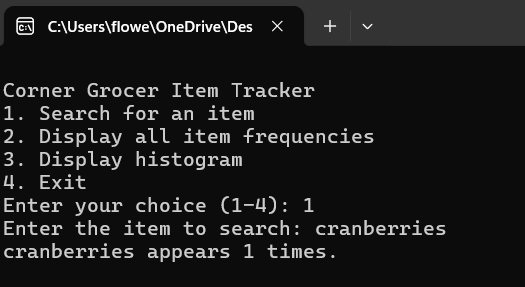
Figure 2: Item Search Output  


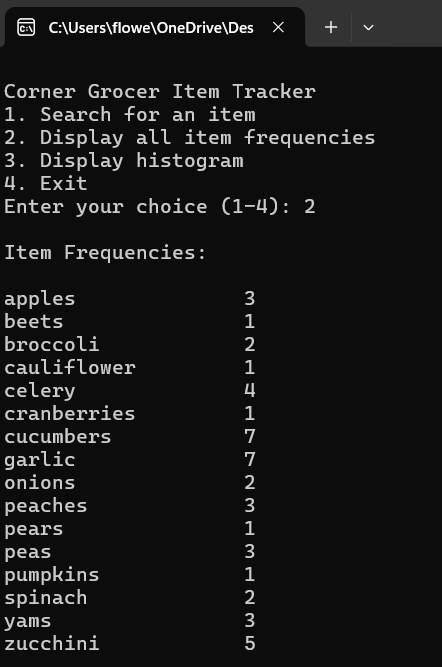
Figure 3: Display All Item Frequencies  


Figure 4: Display Histogram  
