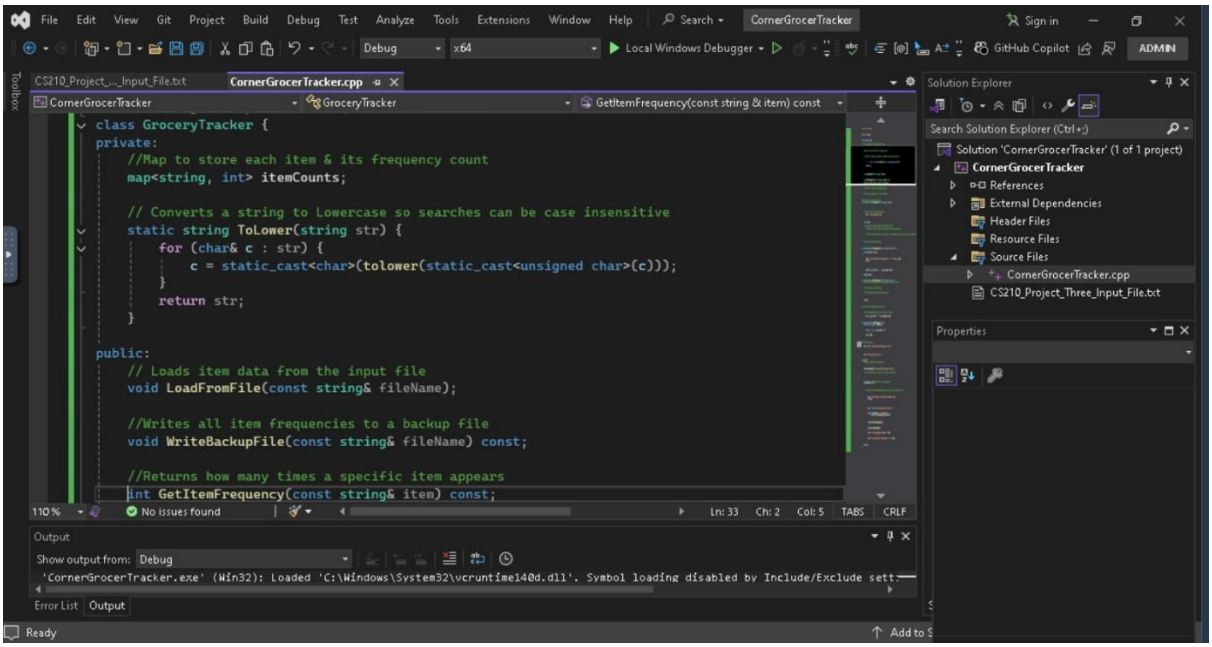
Corner Grocer

The Corner Grocer Item Tracking Program was developed to analyze daily grocery purchase records and determine how frequently each item was purchased. The program reads a text file containing item names listed in chronological order throughout the business day. It processes this data to calculate purchase frequencies and provides an interactive menu that allows the user to search for specific items, view all item frequencies, and display a text-based histogram. In addition to this, the program automatically generates a backup data file named frequency.dat at startup. This file stores each item and its corresponding purchase count to preserve the accumulated data for future reference.

Class Design & Structure

The program was designed using object-oriented principles. A single class named GroceryTracker was created to encapsulate all functionality related to file processing and frequency tracking. The class contains one private data member: map<string, int> itemCounts. This map stores each grocery item as a key & its frequency as the associated value. Using a map was an intentional design choice because it efficiently stores key value pairs and automatically handles unique keys, When reading items from the file, if an item does not yet exist in the map, it is automatically created with a value of zero then incremented. This makes frequency tracking both clean and efficient.

This class contains the following public member functions: LoadFromFile() which reads input data & builds frequency counts, WriteBackupFile() which creates the frequency.dat backup file, GetItemFrequency() which returns the count for a specific item, PrintAllFrequencies() which displays all items & counts, and PrintHistogram() which displays a visual representation using asterisks. Separating these responsibilities into member functions improves readability, maintainability, and organization of the code.

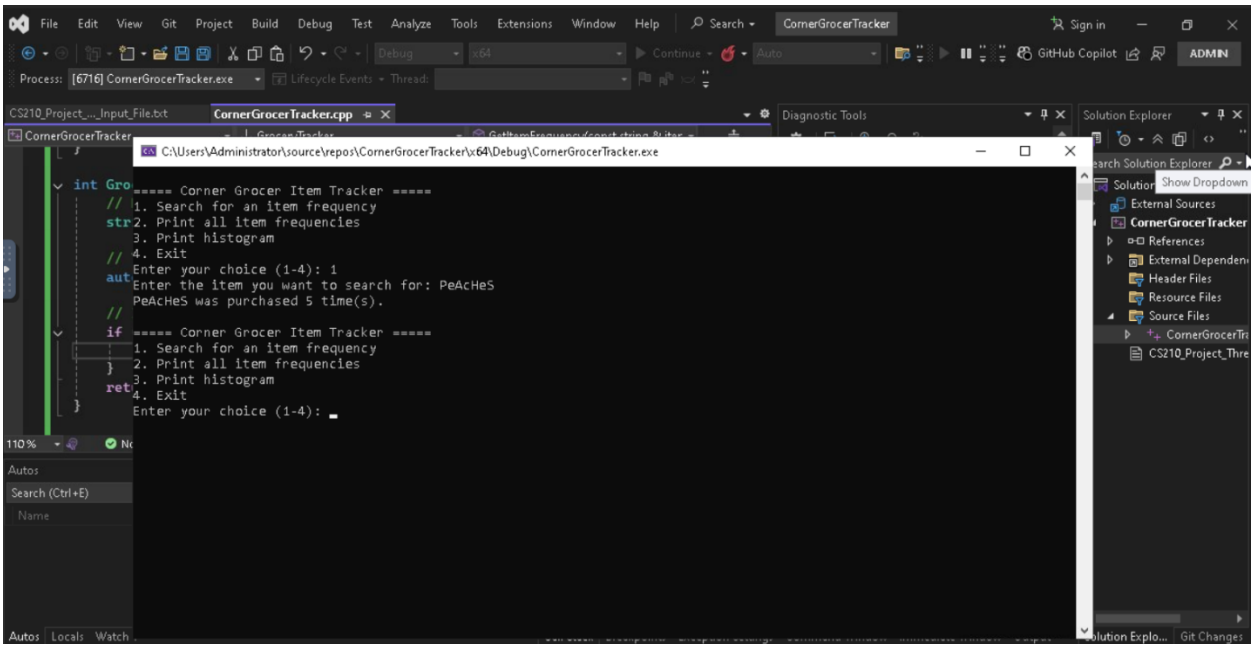


File Handling & Data Processing

The program begins by calling LoadFromFile(), which attempts to open the file CS210_Project_Three_Input_File.txt. File validation is performed using is_open() to ensure the file loads successfully before processing. Items are read using a while (inFile >> item) loop, which continues until the end of the file is reached. Each item is normalized to lowercase before being stored in the map to ensure consistent frequency tracking. Immediately after loading the file, the program calls WriteBackupFile() to create frequency.dat. This function writes each item & its frequency to the backup file without requiring user input. This ensures data persistence and satisfies the assignment requirement for automatic backup creation.

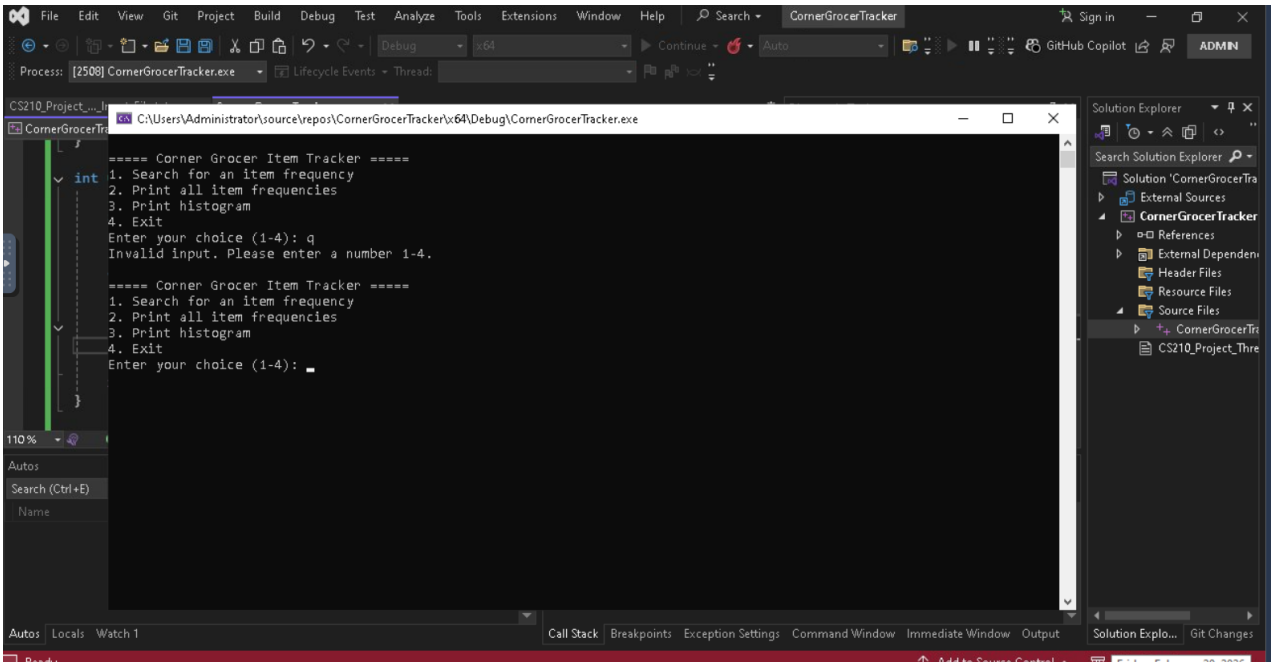
Case insensitive Search Enhancement

To improve usability, the program implements case-insensitive searching. Without this enhancement, searching for “Peaches”, “peaches”, or “PEACHES” would produce different results. A helper function named ToLower() was created to convert strings to lowercase. This function is used when reading items from the input file, and also when processing user search input. By standardizing all stored data and search terms to lowercase, the program ensures accurate & consistent matching regardless of how the user types the item name.



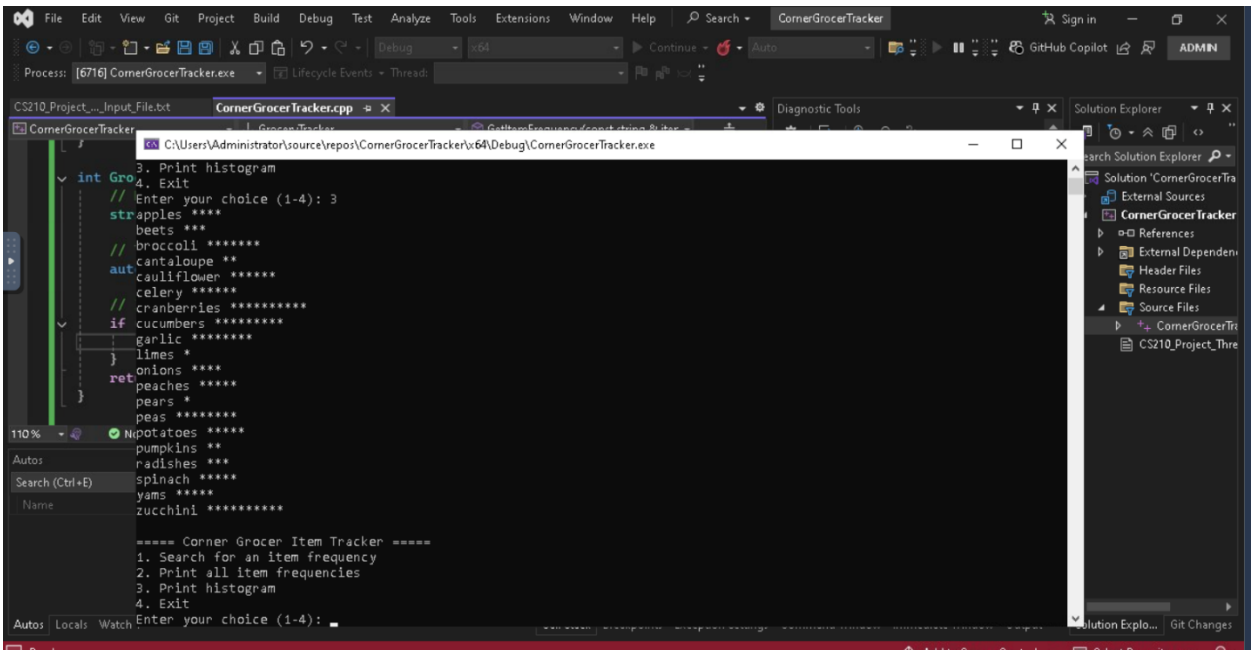
Menu System & Input Validation

The program includes an interactive menu that allows the user to select from four options. 1. Search for an item frequency. 2. Print all item frequencies. 3. Print a histogram. 4. Exit the program. The menu runs inside a loop that continues until the user selects option 4. Basic input validation is implemented using cin.fail() to prevent the program from crashing if a user enters invalid input (such as letters instead of numbers). If invalid input is detected the input stream is cleared and the user is prompted again. This improves readability and user experience.



Histogram Representation

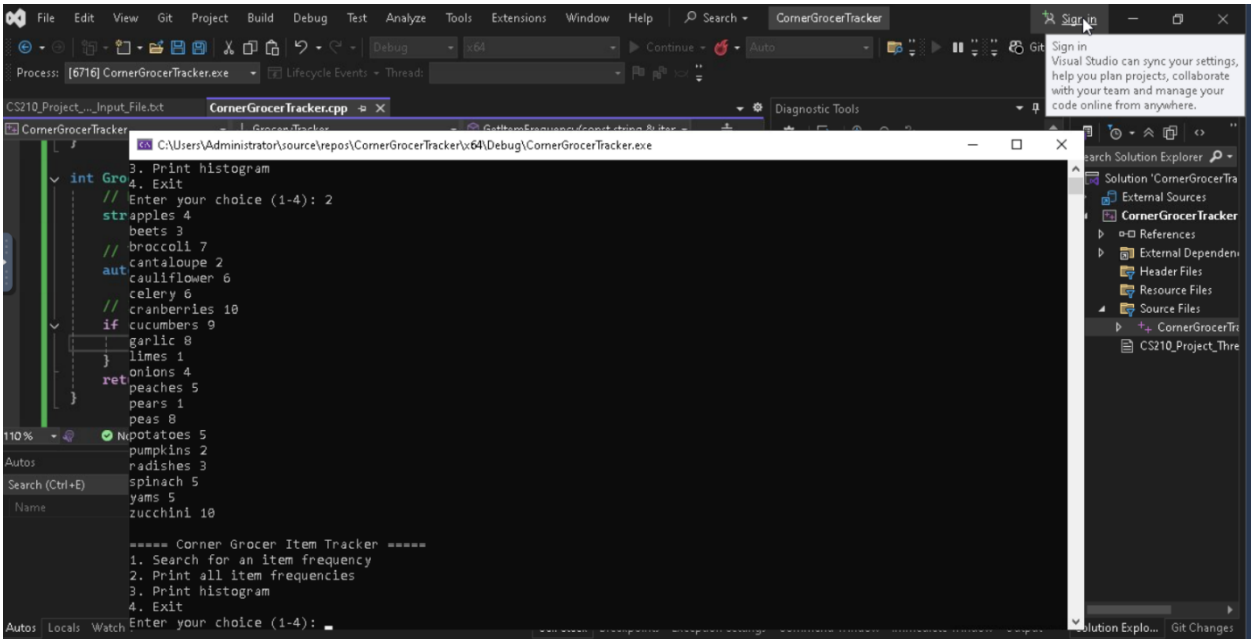
The histogram feature visually represents item frequency by printing one asterisk (*) per record purchase. This provides a quick and intuitive visual comparison of purchase trends, helping the store determine how to reorganize its produce section effectively.



Program Execution & Testing

The program was tested using multiple search inputs, including variations in capitalization,

to verify that case-insensitive searching functions correctly. All menu options were tested to ensure accurate frequency reporting and correct histogram output. The backup file frequency.dat was also verified to confirm that it contains the correct item counts immediately upon program execution.



In conclusion, the Corner Grocer Item Tracking Program successfully meets all functional requirements outlined. It demonstrates proper use of file input and output, object oriented design, associative containers (maps), user interaction, input validation, and data normalization. The program is organized, readable, and maintainable, following industry standard best practices such as inline comments, clear naming conventions, and structured logic flow.

References

Vahid, F., & Lysecky, R. (2019). CS 210: Programming languages. zyBooks. http://www.zybooks.com