PF2 Homework 3 – Jessie’s Coronavirus Database

Furqaan Indori 010852950

CSCE 2014-001

Summer 2021

Dr. Lora Streeter

**Problem Statement:** The goal of this programming project was to implement a program that would help people like Jessie easily access information about the numbers of COVID-19 cases in various counties. A csv file was provided with the county names, state names, and case numbers from three distinct months. There were also instructions that specified the way specific methods were to be implemented (mostly recursively for this project). The main inputs were multiple token strings, one being the county name and the other being the state code. The main output was all the relevant information on the county stored in the database and the number of counties where the mask mandate was effective. Simple error handling was required, such as making sure the recursive binary search stopped itself if a matching county name and associated state were not found in the database.

**Design:** There were a few design decisions made in this project, but most of them were purely cosmetic. An ASCII art menu title was generated using [the internet](https://patorjk.com/software/taag/#p=display&h=2&v=0&f=Big&t=Coronavirus%20Database) because the main menu felt a bit underwhelming. Also, horizontal rules were added to the print method in to separate the data from the initial user prompts. The decisions that influenced the functionality of the program were made in the two recursive methods. For the mask mandate method, instead of having a return value, a variable was passed by reference to store the result because it seemed more efficient than passing the counter variable through every recursive function call. For the recursive binary search, variables were made to store the results of the compare method on both the states and the counties so that it would be more readable.

**Implementation:** The implementation process began with creating the Coronavirus class and adding the associated variables, getters, and setters. Then, the main menu and the readFile method were created and debugged. Finally, the print method and the two recursive methods were implemented. Most of the work went into the two recursive methods because of repeated testing. At one point in implementation, the recursive binary search method was completely rewritten. This occurrence made it apparent that when implementing complex methods, writing pseudocode in comments before the actual code is extremely helpful.

**Testing:** The project was tested multiple times during each method implementation, usually after implementing a new statement or altering a condition, to ensure that it was working correctly. Normal inputs were multiple token strings that were read using the getline(cin, variableName) method. There were no special cases in this project because all input was read in as strings and the user was not prompted for any numerical data. Special characters, such as those with accents or those from other languages will not be parsed correctly because cin has a limited range of ASCII values which it can recognize. Methods for the vector class head to be researched before they were implemented, but they worked as expected. Vectors are much more user-friendly than linked lists.

*(Sample Inputs/Outputs included in the typescript file)*

**Conclusion:** This programming project was a success in that I learned a lot from writing and debugging it. Next time, I will write pseudocode first for more complicated methods and then begin implementation. It helps to get an outline down and then fill in the blanks instead of trying to keep track of everything while writing. I will also use the sample output provided as a reference sooner so that I do not have to fix mistakes at the end of implementation. This project took about 5 hours to complete because vectors already had built in methods for inserting, accessing, and deleting information.