**Programming Project Report**

Name: Gage Underwood

Date: 4/17/20

**Academic Integrity Statement:** I pledge that I have neither given nor received unauthorized help on this programming assignment.

**Problem Statement:**

The goal of this programming assignment is to gain experience with sorting algorithms and document analysis. We gain experience using vectors and learn how they are similar and different from linked lists and arrays. This program takes in one input from the user and has no outputs to the screen however it does have the ability to output up to three files. There was very little error handling required other than making sure my sort didn’t miss any data in my vectors.

**Design:**

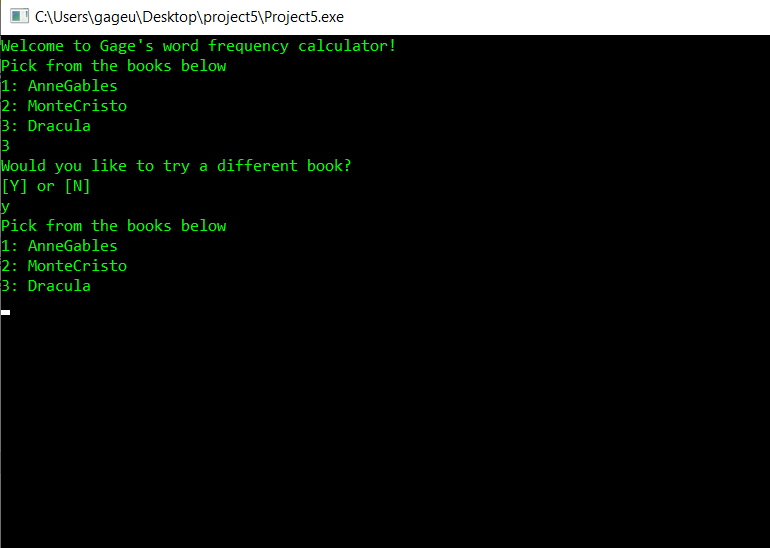
The design decisions I made for this project was using a method from project 3 called fix\_file and adapting it to work with vectors. I also borrowed read\_book from project three and was able to read in and fix all the words with very little additional code. The bulk of the program is vectors of strings with a few integer counters throughout to keep everything in line. The only algorithm I used was recursion with the merge\_sort method. The Pros of making these design choices was a great start on the project but I had no idea what to do after I read the book in and fixed the strings. The largest hurdle I had when designing this project was converting merge\_sort to work with vectors.

**Implementation:**

My implementation process began with skeleton methods that I thought I might need to complete the task at hand. After I had all the methods I started with read\_book and fix\_file as we already implemented the methods in project 3. Once these two methods were up and running I began working on the merge\_sort which now only takes in two integer values for high and low instead of previously taking in an array as well. Once I had merge\_sort compiling I quickly wrote a print function to test and confirm all of these functions were working. I then started working on the calculate function that cycles through the sorted array and counts each repetition of each individual word. The last thing to make was the PrintResults function which actually doesn’t print anything. It writes each book to its own result file that has them in alphabetical order with their count displayed in front of the word. My development timeline for this project was 7 hours total.

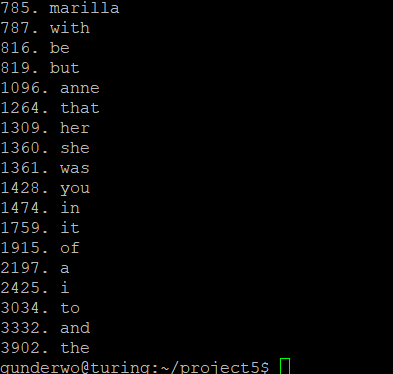
**Testing:**

To test this program I used sample1.txt from the project 3 testing folder as it is a much shorter document to test things with. The actually testing process was running the program and checking the outputs and making changes until I got the desired output. I did not test any special cases. A sample input.

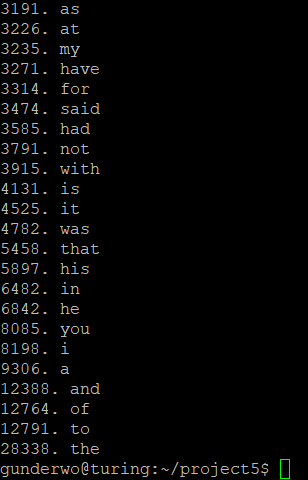


Outputs:

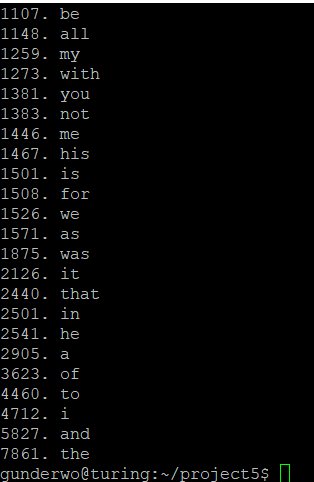
AnneGables



Monte Cristo



Dracula



**Conclusions:**

The overall result of this assignment was a successful frequency calculator that reads words from a file, sorts them alphabetically, and writes them to a file. The only thing I would do is start on it earlier to make the programming project less stressful. This project to me 7 hours total to complete around four spent coding two spent de bugging and 1 spent testing.