**Description**

This project is designed as the primary basis of a word cloud. A word cloud is an image that represents the frequency of a word by showing that word to be bigger in size than words that are not as prominent. In this project, I coded for the first half of the word cloud where the code would determine the prominence of every word in a given text file. This project will use functions to ensure that all the words from a given text file are taken in and accounted for. It will make sure that there are no errors in counting by implementing code that removes any capitalization and punctuation in the word so that words such as “independent”, “InDePeNdEnT” and “i.n.d.e.p.e.n.d.e.n.t” are all counted as the same word. It asks the user for the number of words that they want the count for and using this info and the input file, it will take in all the words and return in a separate file the top frequency words according to the number the user input. It returns a file with the word and their frequencies and also a total count of how many words there were total.

**Structures Used**

For this project, the three main data structures I used were vectors, strings, and structs. Ints were implemented as a way to keep count of everything. A string was used because it allows for easy parsing character by character to ensure that there are no extraneous characters included. A vector was used because the size of a structure needed was unsure. With a vector I could easily add or remove data. Finally a struct was used because it allowed me to keep track of multiple pieces of data at the same time. The struct in my case was utilized to ensure that I could store a word and with that word I could store its frequency.

**Instructions**

For this code, ensure that you have a file stored in a location where the complier can access the file or that you know the path to the file. When you run the program, it will ask you for the name/path of the file. Enter the information correctly (a file titled “Test\_File.txt” was provided with the project for testing purposes). If the information you entered does not lead to a file, the program will continue to ask for the name/path of file until the file could be found and opened. After this, the program will return the number of distinct words that were found in the file. It will then ask for a number, enter the number of words you would like the count of (note this value has to be between 1 and the number of distinct words). If you mistakenly enter something wrong, the program will continue to ask for a number until you enter something valid. After this, the program will accomplish its job and let you know when its done and prompt you with a message letting you know the name of the file with the information you requested.

**IDEs and OS**

The IDE I used to program and test this project was Xcode. All the coding was done on Xcode on a Mac computer. In addition to the testing on Xcode, I also tested the code using CodeBlocks on a Mac computer.