CS 2302

Fall 2019

Lab Report #1

John Rodriguez

Due: September 6, 2019

Professor: Olac Fuentes

TA: Anindita Nath

Introduction

For lab 1 we were given the task to find the anagrams for a word in a list of words in a text file. To do this we must be able to read the text file into a list and then make a function that check if there are any anagrams for the user’s inputted word in that list. This lab’s purpose is to review recursion and become more familiar using python or making sure you can make your code compatible with python.

Proposed Solution Design and Implementation

I approached this lab by dividing it into 3 different modules. The first thing I focused on was to create a function that could read a text file with the collection of words and put that into a list in python. I named this function read\_words which takes in a string which is the name of the file I then created a list variable called words\_set that stored the words from the file into it. I also added line strip to remove everything besides the actual words and then returned the list at the end.

The next module of the program that I focused on was a function called anagrams that takes in a word with will be used to find the anagrams of and it also takes in a list of words to pull the anagrams from. The first thing I did inside the function was create a list to store the found anagrams which I called found\_anagrams. I then found a tool called Counter that stores the letters used in a string and how many times that letter is used which I used to find the anagrams of the word proficiently. I then created a variable that held this information for the search word. Then I created a loop that goes through the list of words and compares them the search word using the counter tool and if an anagram is found that word is then added to the list called found\_anagrams. When the loop ends, I then used a condition that removes the search word from the anagrams list then I sorted the list into alphabetical order. I then printed how many anagrams were fond for the search word and returned the found anagrams list.

The final module of my program focused on utilizing the functions that I created, getting a input from the user which would be used as the search word, outputting the questions/statements to the user, and finally timing how long the program took to find the anagrams of the search word. The first thing I did was to use the read\_words function with the file given to use called words\_alpha.txt. I then created a while loop to be used to continue asking the user to input a word that they want to find the anagrams of, and the loop would be exited when the user inputs a empty string. In this loop I first print “Enter a word or empty sting to finish” then the user input is read. I then created a conditional that if the user inputted a word would use the anagrams function to search the list for any anagrams and then output how many were found and print the list of word that were found. I also used a timer tool that can be used to see how long a program takes to run and used that to output how long the search for the anagrams took to the user. If the user inputs a empty string the program will output a goodbye message and the loop will be exited and the program will end.

Experimental Results

**Test Case #1**

The first test case I did was an edge case to see what would happen if the file used to store the list of anagrams wasn’t found when using the function read\_words. To test this, I removed the file from the directory that the program was being ran in. When I ran the program, I received a FileNotFoundError so I then implemented a try and except to my program so if the text file isn’t found the program will output the message “The file was not found, make sure the spelling is correct or if it is in the same directory as the program”. I attached a screenshot showing the error without the try and except and with it.

A screenshot of a cell phone

Description automatically generated

**Test Case #2**

The second test case I did was a base case to test 3 words that came to my head and then enter the blank string to end the program. I did this to see if the program is running the way it was designed to. The program did behave the way it was designed to.A screenshot of a cell phone

Description automatically generated

**Test Case #3**

The third test case was another base case but this time I used the words that were given as examples that would make the program run for a long time. When testing these words my program was able to run them all relatively fast.

A screenshot of a social media post

Description automatically generated

Test Case #4

My fourth test case was an edge case in which I input the search word with unexpected capitalization and extra spaces at the beginning and end of my input. My program should be able to handle this because I implemented .strip() which removes the white spaces from the input and .lower() which will make all the letters in the string lowercase. My program was able to handle a user input like this.

A screenshot of a social media post

Description automatically generated

Conclusion

This lab helped me become more comfortable when using python and how knowledge in one coding program translate to using a different program. I knew what I wanted to do when designing this program and knew how to do it in java. When I wanted to implement something into my program, I just needed to learn how it is done when using python like reading a file into a list. I also learned that python I much easier to use when compared to java you only need to learn the small differences when using python compared to java. I also have a better understanding on how the source code and report for future labs need to be organized and how they are supposed to be submitted.

Appendix

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class