CS222 Assignment 5 - Arrays and Cstrings

Objective:

The purpose of this assignment is to introduce you to the manipulation of arrays and cstrings.

Description:

You are to develop a program that will search for a word as well as count the number of characters in a text file. Specifically, it will ask the user to input a file name (cstring) and a word (cstring). Then, it will decide whether the word exists in the file, and count the number of characters (A-Z lower or upper case alphabets, excluding all other symbols or special characters) in the file. After that, it will report the results: whether the word is contained in the file and how many characters are in the file.

Instructions:

Here are some basic components for your program design.

First, your main program gets the file name (string) from the keyboard input, make sure that the file exists while opening the file. Specifically, your program asks "Please input the text file name:" Then it uses scanf to read the string from the keyboard, which is the file name for fopen. If the file does not exist, which means that fopen will return NULL instead of the pointer to the file, your program notify the user and quit.

Second, if the file exists, the main program asks for a word "Please input the **word** to search:". After receive the word from the keyboard, it calls **search_and_count()** to search the word and count the characters in the file. So **search_and_count()** will have the file pointer and the word as input, and an indicator (0 or 1) (whether the word is contained in the file) and the total number of characters ('A'-'Z' or 'a'-'z' that are in the file) as output. Your main program then reports whether the word exists in the file, and how many characters are in the file.

search_and_count() will read a string from the file, count the characters in the string, compare the word with the string (to see if the word exists in the file), and repeat until the end of the file. Specifically, you will need to build two more functions:

- **search_and_count()** reads a string from the file into a variable, call function **character_count()** to count the number of characters in the string, and then call function **word_search()** to compare the word with the string to decide whether it is a match. This will go on in a loop until the end of file. If there is a match already, it doesn't need the comaprison any more. In other words, word_search() will be called only if your program hasn't found a match.
- **character_count()** should have an input, which is the current string, and a return value, which is the count of the number of characters in the string.
- **word_search()** should have two input, the word and the string, and a return value 0 or 1 as an indicator whether the word is the same as the string. The program should allow the string has an ending punctuation. That is, you may want to put a NULL in the string to make it the same length as the word, so that you can use strcmp to compare the word with the string.

Finally, in the main program, close your file and you report whether there is a match and how many characters are in the file.

As you have previously done, create a typescript that prints your program to the screen, confirms the system you are on, compiles it using gcc, runs it using various input, and submit *both* the typescript and the source file containing your program to Blackboard as HW 5.

Submission:

You will submit two files: (1) the .c file containing your source code; and (2) a script file containing a listing of your code and a sample run.