CS 133U: C Programming

Assignment 6: Word Frequencies

Academic Integrity

You may NOT, under any circumstances, begin a programming assignment by looking for completed code on StackOverflow or Chegg or any such website, which you can claim as your own. Please check out the Student Code of Conduct at PCC.

The only way to learn to code is to do it yourself. The assignments will be built from examples during the lectures, so ask for clarification during class if something seems confusing. If you start with code from another source and just change the variable names or other content to make it look original, you will receive a zero on the assignment.

I may ask you to explain your assignment verbally. If you cannot satisfactorily explain what your code does, and answer questions about why you wrote it in a particular way, then you should also expect a zero.

Have you ever encountered a word frequency counter? Now you can write one yourself! A word frequency counter allows you to count the frequency of usage of each word in your text. In other words, a program that counts how many times each word appears in a string or text.

Purpose

In this assignment, you will be writing a program to count the occurrence of each word in a string of text.

After completing this assignment you will be able to:

- Work with an array of cstrings (2-dimensional char arrays)
- Use nested for loops and cstring functions
- Work with both int and char arrays

Task

- ❏ Open the Algorithmic Design Document, make a copy, and follow the steps to create your algorithm.
- ❏ You must express your algorithm as **pseudocode** or a **flowchart.**
- ❏ Write a program that reads a list of words. Then, the program outputs those words and their frequencies.
- ❏ The input begins with an integer indicating the number of words that follow. Assume that the list will always contain less than 20 words. Each word will always contain less than 10 characters and no spaces.
- ❏ Ex: If the input is:

```
5 hey hi Mark hi mark
```

- ❏ Then, the output is:

```
hey 1
hi 2
Mark 1
hi 2
mark 1
```

- ❏ **Hint:** Use two arrays, one array for the c-strings and one array for the frequencies.
- ❏ You may not use any temporary arrays to help you solve this problem. (But you may declare as many simple variables as you like, such as ints.) You also may not use any other data structures or complex types such as strings, or other data structures such as Vector.
- ❏ Because C strings are stored using arrays of characters, an array of strings can be created using a two-dimensional char array. Refer to zyBooks 6. C-Strings or Character Arrays Section 6.9 Arrays and Strings for more information on two-dimensional char arrays.
- ❏ Print a goodbye message.

❑ **Use only the concepts we have learned so far.**

Criteria for Success

❑ Test your program using the following sample runs, making sure you get the same output when using the given inputs (in **`blue`**).

❑ Your program should not create any new arrays.

❑ Make sure to test your program with different words and different counts of the words.

```
Welcome to my Word Frequency Counter!!

This frequency will count the number of occurrences of
each word. The number of words in your list must be
entered first followed by the list of words separated by
space. These are the rules of this frequency counter!

Enter the count of words first (as a whole number) and the list of
words separated by space:

8 Hey Hi Hey Priya How are you Priya

The frequency counts are as below:

Hey 2
Hi 1
Priya 2
How 1
are 1
you 1

Thank you for using my frequency counter!
```

❑ Complete zyBooks section **CS133U 6. C-Strings or Character Arrays** activities.

❑ Refer to zyBooks 6. C-Strings or Character Arrays Section 6.9 Arrays and Strings for more information on two-dimensional char arrays.

❑ Complete all sections of your Algorithmic Design Document.

❑ Include **pseudocode** or a **flowchart** in part d of the design document.

❑ Please open and compare your work with the grading rubric before submitting.

❑ Remember to follow all style guidelines.

❑ Download your Algorithmic Design Document as a PDF (File -> Download -> PDF), rename it to `a06.pdf`.

❑ Name your C source file `a06.c` and upload with `a06.pdf` to the D2L assignment.

- ❏ Do your own work. Consult the syllabus for more information about academic integrity.

Additional Support

- ❏ Post a question for the instructor in the Ask Questions! area of the Course Lobby.