

CS1XC3 - Computer Science Practice and Experience: Development Basics

Assignment #3 – C Arrays and Strings

Weight

7% of total course grade

Due date

Sunday February 14th at 11:59pm

Primary Learning Objectives

- Write C programs using arrays and strings.

Requirements

Create a program that will count the occurrences of letters in a user-input string and create a report.

Your program should begin by asking the user to enter text for analysis. The user should be able to type up to 1023 characters and hit enter, and your program should be able to store these characters as a C string. Your program should have a buffer (i.e. char array) for storing this string that can handle up to the maximum length needed to hold this amount of characters (but not more than the amount of space required either).

Your program should then analyze the string to count the number of characters of each letter from 'A' to 'Z' in the string. Your program should count both uppercase and lowercase instances of the characters as an occurrence of the character. In other words, the string "AaaAbB" would have a count of 4 'A' characters and 2 'B' characters. Your program should keep track of the count of each A to Z character using an array, not with individual variables.

Your program should output the total occurrences of each character in a table, along with the percentage of the string that is made up of each character from A-Z. The table should have three columns:

- Letter - 10 characters, left-aligned
- Occurrences - 15 characters, left-aligned
- Percentage – 15 characters, left-aligned, output two decimal digits of precision

Beneath the table, your program should also output the most frequently occurring character from A-Z, and the least frequently occurring character from A-Z. If two or more characters are “tied” for either most frequently occurring character or least frequently occurring character, then you can output any of those characters as being the most frequently or least frequently occurring character.

Hint: Remember that characters in C are really just an int number. It’s possible to write a very simple and short version of this program if you keep this in mind: <http://www.asciitable.com/>.

Your program should run either identically or near identically to this example, in particular the occurrences and percentages should be identical:

Enter text for analysis: The quick brown fox jumps over the lazy dog.

Letter Analysis Complete!

Letter	Occurrences	Percentage

A	1	2.27
B	1	2.27
C	1	2.27
D	1	2.27
E	3	6.82
F	1	2.27
G	1	2.27
H	2	4.55
I	1	2.27
J	1	2.27
K	1	2.27
L	1	2.27
M	1	2.27
N	1	2.27
O	4	9.09
P	1	2.27
Q	1	2.27
R	2	4.55
S	1	2.27
T	2	4.55
U	2	4.55
V	1	2.27
W	1	2.27
X	1	2.27
Y	1	2.27
Z	1	2.27

The most frequently occurring character is O.
The least frequently occurring character is A.

Here is another example of the program running:

Enter text for analysis: aAAa zzzzzZZ yxwv

Letter Analysis Complete!

Letter	Occurrences	Percentage

A	4	15.38
B	0	0.00
C	0	0.00
D	0	0.00
E	0	0.00
F	0	0.00
G	0	0.00
H	0	0.00
I	0	0.00
J	0	0.00
K	0	0.00
L	0	0.00
M	0	0.00
N	0	0.00
O	0	0.00
P	0	0.00
Q	0	0.00
R	0	0.00
S	0	0.00
T	0	0.00
U	0	0.00
V	1	3.85
W	1	3.85
X	1	3.85
Y	1	3.85
Z	7	26.92

The most frequently occurring character is Z.
The least frequently occurring character is B.

Note that it is OK that the percentages of the A-Z characters in the string do not add up to 100%, we would not expect this because it's a percentage of the string that is made up of each

character (and the string may includes characters that are not letters from A-Z, so things like space characters, periods, etc.).

Add some comments to your code that explain what each section of code is responsible for doing, and if needed because it may not be obvious to the read, explain how the code itself is carrying out the work it is responsible for doing.

Help

Keep in mind that we've written a lot of code in-class and in-lab that is similar to what you need to implement! We went over an example of user input of a string with spaces in-class, and finding a minimum number in an array. In-lab during week 4 you will go over examples of analyzing individual characters in a string (postal code), and finding a maximum number in an array. We have gone over examples of printing a formatted table too. So much of what you need to do in this assignment is very similar to these examples, and you should use them to help you out!

Submission

Save your solution in a file named **lettercounter.c** and put it in a zip file named **as3.zip** and submit them to the Assignment #3 dropbox on Avenue to Learn.

Marking scheme

Component	Description	Marks
String input	String input from the user with a correctly sized buffer.	15
Occurrence counting	Counting of occurrences done with an array.	30
Percentage calculation	Calculation of percentages for each character.	10
Most and least frequent chars	Identification and reporting of the most and least frequent characters A-Z to occur in the string.	25
Table	Output of formatted table	10
Comments	Sensible comments documenting the code.	10
Total:		100