

## Laboratory Exercise 3 (C++ structs)

Topics: C++ structs

C++ arrays

C strings

Sorting

Goals: Upon successful completion of this lab you should be able to:

1. Define and use a struct in a C++ program
2. Define and use a static array in a C++ program
3. Define and use C strings in a C++ programs
4. Sort the entries in an array

Related text sections:

Chapter 5

Chapter 6

Chapter 9

## Laboratory Exercise 3 Instructions

### Part 1.

1. Add the following structure definition to the program you submitted for lab 2.

```
struct worddata
{
    char * word;
    int vowels;
    int consonants;
    int digits;
    int specialchars;
};
```

2. Modify your program so that it reads the words from the file “words.txt” into an array of worddata entries. There will be no more than 100 words in the file.
3. Modify your program so that it stores the number of vowels, consonants, digits, and special characters for each word in the worddata entry for the word.
4. Modify your program so that it stores the total number of vowels, consonants, digits, and special characters in a separate worddata variable.

### Part 2.

1. Create a flow chart for a function called wordsort that will sort an array of worddata entries into ASCII lexicographical order. The function should implement the *selection* sort (as discussed in lab.) The parameters for wordsort are an array of worddata entries and a count of the number of used entries in the array. The return type is void. The entries should be sorted into ascending word order.
2. Add your wordsort function to your program. You will also need to a worddata Swap function to your program.
3. Modify the output of your program so that it prints the word instead of the word number. Words should be left justified in 12 spaces. (Numeric values should remain right justified in 8 spaces.)
4. Test your program to make sure that it reads the words from the file, calculates the appropriate values, and prints the words and their corresponding values in sorted order. Sample data is illustrated on the next page.
6. Place your completed program in the dropbox ~tiawatts/cs215drop as *yourlastnameL3.cpp*.
7. Place your completed flowchart in the dropbox ~tiawatts/cs215drop as *yourlastnameL3.sfc*.
8. Verify that both of your files (.cpp and .sfc) are in the dropbox.

Sample Input

This file contains  
15 words. It has  
lots of letters  
and very few  
special  
characters!



Sample Output

Word	Vowels	Const.	Digits	Special
15	0	0	2	0
It	1	1	0	0
This	1	3	0	0
and	1	2	0	0
characters!	3	7	0	1
contains	3	5	0	0
few	1	2	0	0
file	2	2	0	0
has	1	2	0	0
letters	2	5	0	0
lots	1	3	0	0
of	1	1	0	0
special	3	4	0	0
very	1	3	0	0
words.	1	4	0	1
	---	---	---	---
Totals	22	44	2	2

