CS 100 Lab Six – Spring 2016

WARNING – Not properly submitting your files correctly (a zip file that contains your directory lab6 with four xxx.c files within it) will result in a penalty of 20 points.

Create a directory called **lab6** on your machine. Move into that directory. Complete the four tasks shown below.

1. Name this program **words.c** – This program takes the name of an input file from the command line and reads in all the words (tokens) in the file. After processing the file, it should tell you the total number of words (tokens) in the file and the average word (token) length. You are counting total words, not unique words, so don't worry about multiple occurrences of a given word. Run your program with **./a.out inputFile**

2. Name this program **sort.c** – This program takes the name of an input file from the command line. This input file contains nothing but integers. Your program should process these integers, writing all the even numbers to the file **evens** and all the odd numbers to the file **odds**. You do not know how many integers are in the file. Run your program via the command **./a.out inputFile**

3. Name this program **most.c** – This program takes the name of an input file from the command line. This file contains numbers in the range of **0** to **99**. Your program should print out the number(s) that occur the most often within this file. Run your program via the command **./a.out inputFile** For example, if the file contained the numbers below then you would identify **7** and **90** and **93** as the most occurring numbers.

   ```
   1 93 7 21 42 59 90 21 90 83 1 90 7 93 63 93 2 97 7
   ```

   Hint – declare an array of 100 items that holds the "count" of each integer. Each time you see a given number in the input, increment that location in your "count" array by one. After reading all the values in the file, find the maximum of the "count" values and then print each element that has that count.

4. Name this program **merge.c** – This program takes the names of two input files from the command line and merges these two files into a single (new) file. Each of these files contains integers, and both input files are in ascending (sorted) order. You should create a new output file named **allData** that contains all the numbers in the two input files and is in sorted (ascending) order. The example below shows what **allData** would contain if you had the two data files **data1** and **data2** as input and ran the program with **./a.out data1 data2**

   ```
   data1:    1 5 9 15 82
   data2:    2 4 15 18 37 64 99
   allData:  1 2 4 5 9 15 18 37 64 82 99
   ```

   Hint – this problem is very similar to what you need to do for Project Three. Take a look at the project three write-up for a discussion on how to merge two (sorted) input files. You simply have to add one additional case in the initial loop – the case where the smallest value is in both input files.

Submit your lab

First, on your local machine, bundle the files in your lab6 directory into a single (compressed) file. To do this:

- PC: Using Windows Explorer, right click on the lab6 directory and select "Send To" and then "Compressed (zipped) folder"
- Mac: Using Finder, use a secondary click on the lab6 directory and then select "Compress *foldername*"

Once you have a zip file that contains your four lab6 programs, submit that file to Blackboard.

Attendance: We will circulate a roster sheet shortly after lab starts and again about half-way through the lab. Not being present to sign the roster sheet will result in a deduction of 25 points for each missed signature.