CS 241 Data Organization using C
Lab 2: Counting Characters, Words, and Lines

Fall 2014

1 Requirements

The program given in section 1.5.4 of Kernighan and Ritchie reads a text file from the standard input stream and outputs the number of characters, words, and lines of the input file. Your task is to modify this program (or completely rewrite it) so that for each line of the input file, the characters in the line are echoed to the standard output stream preceded by the line number and a closing parenthesis and followed by the number of words in the line and the number of characters in the line enclosed in square brackets and separated by a comma.

After echoing the original file, output three more lines with the following information: the number of lines, words, and characters (as in the original, just more verbose); the line number of the line with the most characters and the number of characters on that line; the line number of the line with the most words and the number of words on that line. If two or more lines tie for the same number of words/characters, report the later line. Match the formatting given in the example below.

For example, given an input file containing:

```
It is by will alone I set my mind in motion.
It is by the juice of Sapho that thoughts acquire speed,
   the lips acquire stains, the stains become a warning.
It is by will alone I set my mind in motion.
      -- Mentat mantra from David Lynch’s Dune movie
```

Your program must output

```
1)It is by will alone I set my mind in motion.[11,44]
2)It is by the juice of Sapho that thoughts acquire speed,[11,56]
3)   the lips acquire stains, the stains become a warning.[9,56]
4)It is by will alone I set my mind in motion.[11,44]
5)      -- Mentat mantra from David Lynch’s Dune movie[8,52]
5 lines, 50 words, 252 characters
Line 4 has the most words (11)
Line 3 has the most characters (56)
```

You may assume that the input files contain only the standard ASCII printable characters (decimal codes 32 through 126) along with newline, `’\n’`, and tab, `’\t’`.

Hint: When a character, `c`, is read that is not `EOF` and not a newline, immediately echo it to the standard output stream with: `printf("%c", c);`

2 Turning in your assignment

Attach your program file **wordCount.c** into the Lab 2 assignment in UNM Learn.

3 Grading Rubric (total of 25 points)

[1 point]: The program starts with a comment stating the students first and last name.

[1 point]: The program is correctly named.

[2 points]: Program compiles without errors or warnings on **moons.cs.unm.edu** using **/usr/bin/gcc** with no options.

[5 points]: Code follows the CS-241 Coding Standard - including comments as specified, not repeating code, placement of curly brackets, etc.. See the course website for the coding standard document.

[16 points]: Correct output for **testWords.in**. The expected output is found in **testWords.out**. One point is lost for each line of incorrect output. This will be graded by a diff test with **testWords.out**. The two text files are posted on the class website.

3.1 diff test

```
% cp ~bchenoweth/public_html/cs241/lab2-wordCount/testWords.* .
% gcc wordCount.c
% ./a.out < testWords.in > myOut.txt
% diff myOut.txt testWords.out
```

The first line copies all files that begin with “`testWords.`” from the website to your current directory. NOTE: the period at the end of the cp command is the *destination* part of the cp command. In the bash shell, ’.’ means “the current directory”.

The second line compiles your code.

The third line runs your code with the standard input device redirected to the file “**testWords.in**” and with the standard output redirected to “**myOut.txt**”.

Finally, the last line performs a character-by-character comparison of your output with the required output. If diff returns silently (without printing anything), then you have passed all tests.