60-141 – Introduction to Programming II Winter, 2017
Assignment 3
(Submission Deadline: Sunday, Mar. 12, 2017 before 23:59)

Text Analysis

The availability of computers with string-manipulation capabilities has resulted in some rather interesting approaches to analyzing the writings of great authors. Much attention has been focused on whether William Shakespeare ever lived. Some scholars find substantial evidence that Christopher Marlowe actually penned the masterpieces attributed to Shakespeare. Researchers have used computers to find similarities in the writings of these two authors. Regardless of this particular controversy, this assignment examines three methods for analyzing texts with a computer.

Your task is to:

Write a complete, well documented C program that reads several lines of text and prints three tables indicating:

1) the number of occurrences of each letter of the alphabet in the complete text
2) the number of one-letter words, two-letter words, three-letter words, and so on, appearing in the complete text
3) the number of occurrences of each different word in the complete text

Note that the term "complete text" used above refers to all characters in all lines of inputted text.

Requirements and Hints:

1. Given several lines of text, your program should implement at least the following functions (in other words, additional functions may be useful, depending on the approach used):
   a) **void letterAnalysis( )**, that gets several lines of text and the number of lines of text as input parameters and prints a table indicating the number of occurrences of each letter of the alphabet in the complete text.
   b) **int wordLengthAnalysis( )**, that gets several lines of text, the number of lines of text and a length as input parameters and returns the number of occurrences of words with that length appearing in the text. The main() function should call this function with different word lengths and then prints a table indicating the number of one-letter words, two-letter words, three-letter words, and so on, in the text. The maximum word length may be assumed to be 20 letters.
   c) **void wordAnalysis( )**, that gets several lines of text and the number of lines of text as input parameters and prints a table indicating the number of occurrences of each different word in the text. The program should include the words in the table in the same order in which they appear in the text.

Input:

Input is a number (say, int N) and several (i.e. N) lines of text from a file (using input redirection) or from the user (using standard keyboard input). For this, first the number (N) of text lines should be read, and then each line of text should be read individually. The maximum number of lines is 10 but

each text line might have different lengths (however, the maximum number of characters in any individual line is 80).

Hint: You can use a two dimensional array or an array of pointers to save the text lines.

For example

```
4
To be, or not to be? That is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles,
```

Output:

The table below is intended to illustrate the output for this assignment, but it is presented for visual convenience only. Your program will actually produce the output reports so that various outputs are generated one after another (First output, followed by Second output, followed by Third output). In particular, note the formatting issues to be dealt with. For the First output, the letter count must be right justified in a column of prescribed width. For the Second output, the word "**word**" is used only if the number of words is 1, otherwise "**words**" is used. Finally, for the Third output, "**times**" is used only when the number of occurrences is greater than 1.

First output	Second output	Third output
Total letter counts: a: 10 b: 4 c: 0 d: 2 e: 15 f: 5 g: 3 h: 6 **...**	1 word of length 1 10 words of length 2 8 words of length 3 4 words of length 4 5 words of length 6 2 words of length 7 1 word of length 8 2 words of length 9 1 word of length 10	"To" appeared 1 time "be," appeared 1 time "or" appeared 1 time "not" appeared 1 time "to" appeared 3 times "be?" appeared 1 time "That" appeared 1 time "is" appeared 1 time ...

Note: You do not have to separate punctuation marks (such as comma, dot, or question mark) from the words. For example, both "**be,**" and "**be?**" counted as a three-letter words above.

Requirements:

- Write and document a complete C program that is capable of satisfying the requirements of this assignment problem.
- UNDOCUMENTED OR POORLY DOCUMENTED code will automatically lose 50% marks.
- PLAGIARIZED work will not be graded and receive a mark of ZERO and reported according to the Senate bylaws.
- The question can use of I/O redirection. Please review the textbook for an example on using I/O redirection from flat files.
- TO SUBMIT: No later than the submission deadline, your assignment should be uploaded in

Blackboard. Late submissions are not accepted and will receive a mark of ZERO.
- Upload your work as an attachment, include both the source file (assign3.c) and the script file (assign3.txt) - see below how to create the script file.

To create a script file (one that logs your compilation steps and your output in a text file):
1. script assign3.txt
2. cat assign3.c
3. cat input.txt
4. cc assign2.c
5. a.out < input.txt
6. ls -l
7. exit (DO NOT FORGET THIS STEP!!)

The example script execution presumes that the input was specified in the file input.txt. This may be changed if the input is provided interactively within the program.

If you are compiling your code under Cygwin shell, then you need to change line 5 to:

5. ./a.exe < input.txt

In line 4, students may prefer to use the gcc compiler to ensure improved warning and error diagnostic reports from compiling the program. You must have access to gcc in order to make this change, however.

NOTE: Submissions that are not received correctly by the deadline will automatically receive a ZERO mark. **Late assignment submissions are not accepted!**

NOTES:
1. Your assignment must be RECEIVED by the due date and time. Late assignment submissions are NOT accepted. Keep your script file, and all your code unmodified as proof of its completion in case it is not received.
2. It is your responsibility to get an early start on the assignment, research and ask questions ahead of time from the due date.
3. Marks will be deducted for unclear code (improper spacing and alignment, hard to read programs and missing outputs).
4. Make sure you turn in a complete script file that clearly shows: your code, your compilation process, a listing of the directory showing your source file(s) and the a.out with the date/time stamps, and the output. DO NOT SUBMIT a.out FILES!

5. **PLAGIARISM:** CHEATING IS NOT TOLERATED – PLAGIARISM IS CHEATING! You must submit your own work. Students who are suspected of copying someone else's work will be reported to the department's chair and the Dean of Science and be dealt with in accordance with the University policies. You should not share your code with others. Codes that are similar to each other will BOTH be reported as potential evidence of copying. It is imperative that you write your own code.

6. Authorized/limited help on this assignment may be provided directly from your Lecture or Lab instructors and Teaching Assistants.