CS 118 — Programming Fundamentals
Assignment #6

Due Date: Sunday, November 10th at 11:55pm on Google Classroom.

Instructions: Assignments are to be done individually. **No late assignments will be accepted.** You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. The code your write must be your own. You are encouraged to get help from the instructional staff. You may post general questions on Piazza. Do not post more than one line of code when using Piazza.

You must **submit a single zip file** containing your code and documentation on Google Classroom named ⟨*your_student_id*⟩*.zip* where ⟨*your_student_id*⟩ is something like *i19-XXXX*. This means that you must submit only **one file named *i19-XXXX.zip* containing only your source files**. Each file that you submit **must contain your name, student-id, and assignment#** on top of the file in comments. You submission must NOT contain multiple main() functions, otherwise it will not compile for grading. Test your program on a lab machine before submission.

Follow the instructions. Assignments not following the instructions will be awarded zero points.

Assignment Statement:

Text processing is an important area in the field of Artificial Intelligence. One of the tasks in text processing would be to gauge the readability of a particular piece of text. Automated Readability Index(ARI) is a score designed to gauge the readability of a text.

The formula for calculating ARI is given below:

$$4.71\left(\frac{\text{characters}}{\text{words}}\right) + 0.5\left(\frac{\text{words}}{\text{sentences}}\right) - 21.43$$

where *characters* is the number of letters and numbers, *words* is the number of alpha-numeric sequences, and *sentences* is the number of sentences, which were counted manually by the typist when the above formula was developed. Non-integer scores are always rounded up to the nearest whole number, so a score of 10.1 or 10.6 would be converted to 11.

As a rough guide, grade level 1 corresponds to ages 6-8. Reading level grade 8 corresponds to the typical reading level of a 14-year-old child. Grade 12, the highest secondary-school grade before college and corresponds to the reading level of a 17-year-old. For reference see the table below:

Score	Age	Grade Level
1	5-6	Kindergarten
2	6-7	First/Second Grade
3	7-9	Third Grade
4	9-10	Fourth Grade
5	10-11	Fifth Grade
6	11-12	Sixth Grade
7	12-13	Seventh Grade
8	13-14	Eighth Grade
9	14-15	Ninth Grade
10	15-16	Tenth Grade
11	16-17	Eleventh Grade
12	17-18	Twelfth grade
13	18-24	College student
14	24+	Professor

The index is calculated by a fixed set of rules for counting the number of sentences, words, and characters in a piece of text. This can be automated via a computer program. Here is an example. Consider the following sentence:

It was an extraordinarily windy day, and thus the riders were faced with several arduous climbs up the mountain, with the wind trying to push them back down the road.

The Readability Index for that sentence is 15 using our algorithm. The following conveys almost the same idea,

It was a very windy day. The riders had many hard climbs up mountains. The wind kept pushing them back down the road.

This set of sentences has a Readability Index of 2. This method of determining the readability of a piece of text does not do any sort of linguistic analysis so the results can be misleading, but the method usually produces a reasonable answer.

Note, these rules are a heuristic. Heuristics may not always achieve the desired outcome, but they are extremely valuable to problem-solving processes. Heuristics are valuable because they simplify the problem solving process and usually give a good answer if not always a perfect answer.

Your program must count the number of characters, number of words, and number of sentences. Certain assumptions are made about what is a character, word, and sentence in order to make it easier to write a program to do the analysis.

Sentences are the easiest to count. Each occurrence of a period, colon, semicolon, question mark, and exclamation mark is counted as a sentence. Thus the String "Wow!!!" has 1 word with 3 characters, but 3 sentences. (Again this set of rules is a heuristic. A set of rules that often gives a good answer, but occasionally gives bad or nonsensical answers. It is possible per these rules to have a sentence with no words). If a text has no sentence characters assume it has 1 sentence.

A *character* is any alpha-numeric character. Punctuation marks and spaces are not counted as characters. A *word* is sequence of one or more alpha-numeric characters delimited by white space or by a sentence terminators as listed in rule 3, whether or not it is an actual English word. White space is defined as a space, tab, a new line character, and the end of the string itself. Again this gives some results that may not make sense. Any continuous sequence of alphabets and digits is considered as a *word*.

For example the text

shopkeeper's shoes 4 his_child2ren.

contains 6 words and a single sentence containing 29 characters. In the above example *shopkeeper* and *s* are considered as two separate words and 4 is also considered a word.

Your program should continuosly ask the user for input text until the user types `quit` as input. You can assume that the text length shall not exceed 1000 characters including spaces and punctuation marks. A sample run of the program is given below:

```
Enter sentence 1: This is a sentence. So is this!

Number of sentences: 2
Number of words: 7
Number of characters: 23
Readability index: 0
```

```
Enter sentence 2: It continued raining for many days.  One day, a monkey wet in the
rain came into the forest. He sat on a branch, shivering with cold, water dripping
from its body.

Number of sentences: 3
Number of words: 31
Number of characters: 126
Readability index: 3

Enter sentence 3: There was once a poor servant-girl, who was industrious and cleanly,
and swept the house every day, and emptied her sweepings on the great heap in front of
the door. One morning when she was just going back to her work, she found a letter on
this heap, and as she could not read, she put her broom in the corner, and took the
letter to her master and mistress, and behold it was an invitation from the elves,
who asked the girl to hold a child for them at its christening. The girl did not know
what to do, but at length, after much persuasion, and as they told her that it was not
right to refuse an invitation of this kind, she consented.

Number of sentences: 3
Number of words: 126
Number of characters: 499
Readability index: 19

Enter sentence 4: quit
```

This above example is merely to show how the algorithm works regardless of if the input is standard English or not. You could even run the algorithm on source code, although the answer would not be very helpful or meaningful.

1. Implement a multi-pass algorithm. This means your run through the text three times. Once to count the sentences, once to count the words, and once to count the characters. You should create a different method for each of these passes.

2. Assume that the number of characters in each text is less than 1000.

3. Using the library functions in `<ctype.h>` will make things slightly easier.

Style issues. We will grade program hygiene as well as correctness. Did you provide a good structure to the program using functions? Did you minimize the scope of variables to the smallest necessary? Did you use meaningful identifiers? Did you provide consistent tabbing and spacing for code inside functions and if statements? Did you provide comments for your functions?

Honor Policy

This assignment is a individual learning opportunity that will be evaluated based on your ability to think independently, work through a problem in a logical manner solve the problems on your own. You may however discuss verbally or via email the general nature of the conceptual problem to be solved with your classmates or the course instructor, but you are to complete the actual assignment without resorting to help from any other person or other resources that are not authorized as part of this course. If in doubt, ask the course instructor. You may not use the Internet to search for solutions to the problem.