CSCI 141, Project 2

Fall 2013

Objectives:

- Using complex boolean expresions.
- Understanding state-based text processing.

Due date: Submit your program by midnight, Friday, November 22 for full credit.

Flesch readability You can read about the Flesch Readability Ease test in Wikipedia. It counts the number of sentences, words, and syllables in a document and then computes the following number:

$$206.835 - 1.015\left(\frac{\text{words}}{\text{sentences}}\right) - 84.6\left(\frac{\text{syllables}}{\text{words}}\right)$$

The higher the score, the easier it is to read. Scores above 90 should be readable by an average 11-year-old. Scores between 60 and 70 should be readable by 13- to 15-year-olds. Scores below 30 are very difficult to read. Here are some scores I got from my program:

Moby Dick	64.70
Tarzan of the Apes	62.02
A Princess of Mars	53.25
A Christmas Carol	71.63
Great Expectations	70.72

You can download these in plain text format from Project Gutenberg `http://www.gutenberg.org/`. To give a fair reading, you may want to edit out the front and back matter that Gutenberg adds to the documents. It's also fun to analyze government documents and warranty statements.

It's interesting that the two books by Charles Dickens, although very different books in character, are very close on this index, indicating that Mr. Dickens has a consistent style regardless of what he is writing. Mr. Burroughs, not so much.

Reading from a file: For this exercise we need to read a text from a file. To read the entire contents into a single string, follow this pattern:

```
myfile = open('myfile.txt')
txt = myfile.read()
```

Test this out on a few short files, and examine the `txt` string to see how this works.

Program Details: We can now compute an approximation to the three quantities necessary for this assignment while going through a text string character by character, and update some statistics as we go along. We will make some simplifying assumptions about what constitutes a sentence, word, or syllable, but it should be good enough for relative comparisons between documents.

We start with three integer variables, `numSentences`, `numWords`, and `numSyllables`, all set initially to zero.

While going through the characters, you take note each time a sentence, word, or syllable begins or ends. For our purposes:

- Sentences:
  - End whenever we hit a period, semicolon, colon, question mark or exclamation mark.
  - Begin on any other character.
- Words:
  - Begin whenever we hit a letter (upper or lowercase 'a' .. 'z').
  - End on any other character.
- Syllables:
  - Begin whenever we hit a vowel (upper or lowercase 'a', 'e', 'i', 'o', 'u', 'y').
  - End on any other character.

You should write simple functions that return booleans to test all six of these. For example:

```
def sentenceEnd(char):
  return char in '.;:?!'
```

You also maintain three boolean variables, `inSentence`, `inWord`, and `inSyllable`. They all start out `False`. Every time one of them is false and we **start** that item (sentence, word, or syllable), we set that variable to `True`. Every time one of them is true and we **end** that item, we set the variable to `False` *and increment the count of that item!*

Procedure output: Write a procedure `flesch` that behaves as follows:

```
>>> flesch('tarzn10.txt')
Syllables:       134518
Words:            89752
Sentences:         5056
Readability:      62.02
```

Your numbers may not exactly match mine, depending on the text you use *etc.*, but they should be approximately the same. Exact formatting is also not important, just make it nice and neat.

Submit: Zip your program together with a text file you want to analyze, and make it so the program runs an analysis on that file when the TA hits "run."