Cosc 2150 Program 1
Due: Oct 4, 2019 100 points

In this assignment you will implement your first spell checker. For this version, you are going to implement the spell checking with a Linked List. You can NOT use the STL list for your linked list, instead you are going to write the linked list code and then use it in the spell checking.

You will be given a dictionary, in which the words are in a random order. The words in the dictionary are unique, but some may have capital letters. The dictionary is large, having over 125,000 words in it. Reading this in word by word will be easy, since there is one word per line. The dictionary will be entered into the linked list you wrote.

The book has almost a million words in it! This will be a bit more complicated, since every line will have multiple words. Read each word and then send it to a "clean word" method. You will need to write a "clean word" function that will isolate individual words from the file, since specs below. You will need use the word and check it again the dictionary data structure above. You will also be timing this as well.

"Clean word" method:

- This is to be implemented as a separate method, so it can be called for either a dictionary word or book word.
- When reading the book, a word will be defined as between two spaces (or end of line).
- The function will remove any non-letters, except an apostrophe (') and numbers from the word.
- All letters will changed to lower case.
- It will then return the updated word. The word could now be blank. The calling method will need to deal with this possibility.

Program requirements:

1. You must write a linked list. You cannot use the STL list. You can however use other STL methods and functions.
2. "Clean word" method must implemented as described above.
3. Any word that starts with a number (after returning from the clean word function) will be skipped for spell checking.
4. You will use the time code provided to time the spell checking. The time to initial the dictionary IS NOT part of the timing.
5. Output the following information. See the next page for required format of the output.
   - Time to spell check
   - Number of words spell correctly
   - And the Number of compares, and average number of compares for spelled correctly.
   - Number of words not spelled correctly
   - And the Number of compares, and average number of compares for misspelled
   - Number of words skipped.
6. Use the timing code provided, to time the run time.

Output section: No other output, besides what is below, number is RED are expected to be different, but use the same formatting. Removing any of your debugging lines.

```
dictionary size 133168
Done checking and these are the results
finished in time: 1306.53
There are 950068 words found in the dictionary
     172150028 compares. Average: 181
There are 27075 words NOT found in the dictionary
     19383592 compares. Average: 715
There are 2544 words not checked.
```

Run time for extra credit.
To compete for extra credit, first the program must run correctly and meet all the specification above.

- 5 gears for finishing in under 45 minutes on the Pi device.
  - Roughly 8 minutes or less on the linux system, but all timings will be on the Pis
- addition 5 gears for finishing in under 30 minutes
  - Roughly 5 minutes or less on the Linux system.
- Finally, 10 addition gears if you can beat my time. Which is listed in the format section. On the linux system, it ran in 98.64 seconds (1.6 minutes).

Turn in:
Hard copy:
A cover page with the following information

Cosc 2150
Program #1
your Name
Repo name
Competing: YES or NO

in large font at the top of the page. At the bottom of the page, include a non-empty statement of help delivered and help received. It is OK to state that no help was given or received. It is **NOT** ok to omit the statement of help.

Soft Copy:

1. Copy any .cpp and .h files to the repo. Use this link to create the repo: https://classroom.github.com/a/XRi7rGsI
2. Edit readme.md file, add the following:
   - Name
   - Competing: YES or NO
     - And list your best run time
   - List anything that doesn't work (that you know of)
3. Lastly, verify all the necessary files are on the github website. If the files are missing then you **DID NOT** turn it in.