## 15-110 Hw5 - Programming Portion

Each of these problems should be solved in the starter file available on the course website. Submit your code to the Gradescope assignment Hw5 - Programming for autograding.

All programming problems may also be checked by running the starter file, which calls the function `testAll()` to run test cases on all programs.

## #1 - Counting with MapReduce - 8pts

Implement two functions (a **mapper** and a **reducer**) that can be used in the MapReduce algorithm to count the total number of capitalized words across a collection of files.

For the purposes of this problem, a word is defined as a series of characters separated from other words by either a space or a newline. A capitalized word is one that starts with a capital letter (A-Z), which may be followed by other characters that are or are not capitalized. A word that starts with a non-alphabetic character doesn't count.

First, implement the **mapper**, `mapFileToCount`, which takes a string (text from a file) and returns the number of capitalized words in that string. Note that the text from the file contains many lines of text separated by newlines- you'll need to find a way to split apart words based on both spaces **and** newlines.

We've implemented a **collector** which will call `mapFileToCount` some number of times, then store each of the results in a list.

Second, implement the **reducer**, `reduceCountsToTotal`, which takes a list of integers (the counts combined by the collector) and returns the total number of capitalized words across all the files.

You can test these functions independently by running `testMapFileToCount` and `testReduceCountsToTotal`. When they both work, you can then run `testCountMapReduce` to run the whole MapReduce process using code we provide in `mapreduce.py`.