HOMEWORK:

A "bag" is a data structure that handles duplicate items with a counter rather than a new entry. The UML for a "bag node" might look like this. We'll keep it simple and make it a struct.

BagNode
+ BagNode * : next + dataValue: string + dataCount: int

The first time an instance of a string is inserted, dataValue contains the string and dataCount is set to one. If another instance of that same string is inserted, dataCount is incremented. The "remove" operation decrements the counter until there is only one left; then on a subsequent remove the entire node is removed; or, as actual removal and node deletion is expensive, simply set the count to zero. This is typical in real programming, in which data structures are expected to grow but rarely shrink. The traversal displays the string, and the count if it is greater than one.

Ordering strings produces another issue; the relational operators use ASCII values so

```
betty < Wilma
```

is false when it should be true. You'll need to write a function that performs lexicographic comparison of strings, as we desire our traversal to be in lexicographical order. Your function need consider only alphabetic letters and digits. Digits come before letters, so in the text "99 Red Balloons" 99 comes before both "Red" and "Balloons."

If duplicate words in a file differ in the case of the first letter (*e.g.* "The" and "the") store only the lower-case version. If duplicate words in the file always begin with the same case (proper nouns always begin with a capital letter) store that version.

Your program should perform the following tasks:

- Display a friendly greeting to the user
- Prompt the user for the name of a file that contains whitespace-delimited text
- Accept that file name and attempt to open the file
    - If the file fails to open, display an appropriate error message and exit
- Process the file by
    - reading the next token in the file
    - removing any leading and / or trailing punctuation
        - we consider punctuation to be anything other than alphanumeric
        - You wrote this code for the palindromes program, remember?
    - inserting the resulting word into the bag
- Close the file
- Prompt the user for another file name, for output
- Accept that file name and open the file
- Dump the traversal of the bag into that file and close the file

Your program should accept the names of the files as command-line parameters. If two file names are given, process the first and dump the traversal into the second. If only *one* file name is given, process the file and dump the traversal to the console. This is convenient for testing the program with relatively small input files. If no command-line parameters are given, prompt for both as specified above.

Running your program with a file containing the test string

```
“The quick brown fox jumps over the lazy dog, Sir!”
```

should result in the following output:

```
brown
dog
fox
jumps
lazy
over
quick
Sir
the (2)
```

The instructor will run your program against both small and large flat-text files. You should test your code with a variety of text files and word counts. Project Gutenberg is your friend here; pretty much any book that is out of copyright is available there as a flat text file. Song lyrics (readily available online, and a good test since they usually have lots of duplicate words), single chapters of books, and short stories make good starting tests. Your code should process “Alice in Wonderland” (a relatively short novel) in under two minutes. The somewhat longer “Moby Dick” and “King James Bible” files are on the professor’s computer as the final tests of your work. Do you know how many times the word “the” appears in the King James Bible? It’s a rather large number. No student code has yet successfully processed the instructor’s flat-text file of “The Oxford English Dictionary (Unabridged).” If you are the first, you get an A. Maybe.

Also try it with the short story “Bottle Party” by John Collier (flat text available online) and your program might crash in a surprising way. Why did it crash? How will you fix it?