CS 352: Project 4 - Index-Creation (Python)

Due: Wednesday, 6 December 2017 (at 11:55pm)

The purpose of this assignment is to write a Python function, `createIndex`, that takes one `String` argument that is the name of a file. It should read the contents of the file, and print a sorted (in ASCII order) index of the words, one per line, where each word is followed by the sequence of line numbers (sorted numerically, separated by comma) that tell where that word occurs. If the same word appears on more than once on the same line, the corresponding number should appear only once.

The starter file for this assignment is `makeIndex.py`. It contains a dummied-up definition of the `createIndex` function. It also contains code that prompts the user for a filename and calls `createIndex`.

For the purposes of this assignment, a word is defined as a contiguous sequence of letters (i.e., 'a' through 'z' and 'A' through 'Z') that does not have any other letters immediately adjacent to it. All other characters in the file should be ignored (but see 'enhancements').

Example

As an example, let's say that the file `input.txt` contained the text

```
I am what I
am, and I (765) do not like Spam. Abc2. Abc3.
```

Then here is what a run of the program might look like (with user input underlined):

```
Please type a file name: input.txt
Abc 2
I 1,2
Spam 2
am 1,2
and 2
do 2
like 2
not 2
what 1
```

(The ordering here is based on the fact that upper-case letters come before lower-case letters in the ASCII encoding.)

Enhancements

If you complete the above successfully and document your program well, you can expect a grade of 'B'. To raise your grade, consider one doing one or more of the following:

- Perform case-folding on your comparisons so that, for example, the words 'Hammer' and 'hammer' are considered to be the same word. You might then print all words in their lower-case version. (Even better would be to print the word in its lower case version if it appears; otherwise print it in its mixed-case version.) [+1/2 grade]
- Consider a single-quote to be part of a word if it appears in the middle of a word or at the end, but not at the beginning. (A single-quote at the end is considered part of the word only if the word does not start with a single-quote.) This will allow contractions and possessives (e.g., can't, neighbors') to be recognized as words, while the single-quotes in 'fun' and 'neighbors' would be ignored. [+3/4 grade]

- Allow multiple files to be indexed by allowing more than one file to be specified [+1 grade]. For example, if `input2.txt`contained:

```
Spam and ham like jam.
I do not
like to do push-ups.
```

then the program run might look like:

```
Please type file name(s): input.txt input2.txt
Abc 2(index.txt)
I 1,2(index.txt); 2(index2.txt)
Spam 2(index.txt); 1(index2.txt)
am 1,2(index.txt)
and 2(index.txt); 1(index2.txt)
do 2(index.txt); 2,3(index2.txt)
ham (index2.txt)
jam (index2.txt)
like 2(index.txt); 1,3(index2.txt)
not 2(index.txt); 2(index2.txt)
push 3(index2.txt)
to 3(index2.txt)
ups 3(index2.txt)
what 1(index.txt)
```

Handing it in

Once you have implemented, debugged, and commented your program, you should hand in your `makeIndex.py` via Moodle (learning.up.edu).