Download the folder **lab\_8\_starting.zip**, unzip it and open it as a PyCharm project. It contains several **.py** and **.txt** files for this and later Labs and Homework. Do as many of the following as you can, submitting each to the **Lab 8 Assignment** link on Canvas.

**[L8-1] (line\_num.py)** Read a string **fname** from the user, then open the file with that name. Iterate over each line in the file, modifying then copying it into a new file named '**lnum\_'** **+** **fname**. Your modification should add a line number to the start of each line. The format of your modification should look like **'dddd: <original line....>'** with **dddd** being the line number, right-justified in a field of width 4 with leading zeroes, followed by a colon and space and the rest of the line. Don't forget to close both files when you are done. A text file **test\_1.txt** is provided for you to test your code: when run, it should create a new file **lnum\_test\_1.txt**, adding line numbers to the start of every line of the original file.

**[L8-2] (date\_parse.java)** Complete the provided code, finishing the function **parse\_date(dstring)**. When called, pass **dstring** as a string in the format "**mm/dd/yyyy"**, where **mm**, **dd**, and **yyyy** are integers. (Assume this input argument has the correct date format.) **parse\_date(dstring)** should split **dstring** into individual strings, convert each to integers **month**, **day**, and **year**, then return the tuple **(month,day,year)**. The provided **main()** reads **date** as a **str** value from the user, calls **parse\_date(date)**, then prints out the returned result.  
 **[L8-3]** (**count\_alice.py)** Examine the provided starting program. It shows starting code for counting the words in the text file **alice.txt**, containing the complete text of the Lewis Carroll story "Alice in Wonderland".

The basic approach: **for aline in fvar** iterates over each line, splitting it into a list of "words" using the string method **split().** Here, the definition of "words" is somewhat naive: **split()** breaks the string into non-blank tokens, so that any punctuation adjacent to a word is considered part of that word.

Do each of the following, by modifying this starting code:

**(a)** Add each of the "words" in a line to a list of all words in the file, **allwords**. Do so by converting the input line as described in steps **(a1)-(a3)**, then splitting the line into a list of words, and adding each list 'word' to **allwords.** We'll explore several possible alternatives for doing this in class.

**(a1)** Modify each line before splitting, by converting its letters to lower-case.

**(a2)** Note that punctuation is often included in words. Adjust your word list by removing punctuation marks from each input line, replacing them with blanks before splitting. Start by removing periods (**.**), question marks (**?**), commas (**,**) , exclamation marks (**!**), colons (**:**), semi-colons (**;**) and double-quotes (**"**). There are others, but be careful: some are part of words, others are not.

**(a3)** Continue if you have time with other punctuation. Hint: **import string**.It is a challenge to correctly handle single quotes and hyphens. Single quotes (**')** are especially tricky: as apostrophes, they belong to the word, but as quotes around a word or phrase, they don't. Are hyphens part of a word or not, in this document? Take a look at the **alice.txt** and search for examples.

**(b)** Print out a count of the number of such "words" in the file (the final length of **allwords**).

**Homework 7 (H7)** asks you to continue with the above word analysis coding.