Lab 14-2 – Dictionaries

Goals

- Work with dictionaries
- File processing

Setup

- Create a new project named **ITP115_L14_2_*Lastname_Firstname*** where *Lastname* is replaced with your last/family name and *Firstname* is replaces with your first name.
- Add the **story.txt** file to your project.
- Create a new .py file in your project named **ITP115_L14_2_*Lastname_Firstname*** where *Lastname* is replaced with your last/family name and *Firstname* is replaces with your first name.
- Your new file must begin with comments in the following format *(replace the name and email with your actual information):*

```
# Name
# ITP 115, Spring 2020
# Lab 14-2
# USC email
```

Background

- A concordance is an alphabetical listing of words in text string that also indicates the line numbers on which the words occur.

Requirements

- Use a dictionary to represent a concordance.
  - Words in the text will be the keys.
  - The line numbers on which the words appear will be the value.
  - Since it is likely that a word will appear on more than one line, the value will be a **list** of integers, not just a single integer.
  - Your program should ignore *capitalization* and *punctuation*. For example, “Tree”, “tree”, and “tree?” should all be considered the same word.
- Use text file “**`story.txt`**” to test your program

  - You must open and read the file within your program.
- Your program must use a main function with no global variables
- Hint:
  - Recall that strings have **`strip`** method which will remove any character from *both* the beginning and the string
  - For example:

```
word = "fight on"          # word is "fight on"
word = word.strip("nf")    # remove either "n" or "f" from
                           # the beginning or the end
print(word)                # word is "ight o"
```

Possible Implementation

Write the following functions:

- **`readFile(fileName)`**
  - Parameters (1): the file name of the dictionary
  - Returns: a dictionary
  - Loop through each line and look at each word in the line to create a dictionary where the words are the keys and the value for each key is a list of the line number where each word occurred.
  - Close the file object
  - Return the dictionary
- **sortKeys(dictionary)**
  - Parameters (1): the dictionary
  - Returns: a list of the keys in alphabetical order
  - Use the sort method on a list
- **main()**
  - Call the readFile function to create a dictionary and store it in a variable.
  - Call the sortKeys function to get a list of sorted keys and store it in a variable.
  - Loop through the sorted keys list to print each word and its concordance (line numbers) using the dictionary.

Sample Output

```
Here is the concordance for the file 'story.txt'
a:  [1, 1, 5, 9]
am:  [5, 9]
know:  [5, 9]
...
who:  [5, 9]
without:  [6, 10]
you:  [5, 6, 9]
zen:  [1, 8]
```

Deliverables and Submission Instructions

- Create a zip file containing your Python code. This cannot be done within PyCharm. Find the file or folder on your computer and compress it.
  a. Windows*:*
     1. Using File Explorer, select your lab folder
     2. Right click
     3. Send to ->
     4. Compressed (zipped) folder
  b. Mac OSX:
     1. Using Finder, select your lab folder
     2. Right click
     3. Compress "*FileName*"
- Upload the zip file to your Blackboard section:
  1. On Blackboard, click on the Labs item in the course menu on the left.
  2. Click on the specific item for this assignment (starts with L and a number).
  3. Click on the Browse My Computer button and select your zip file.
  4. Click the Submit button.