Homework 5

SI 206, Winter 23

In this homework, you have been given a selection of random biography pages from Wikipedia. This information can be found in the file “206_hw5_wiki_bios.txt”. Your assignment is to use regular expressions to extract information from these biographies. To be clear, each function (except for read_file) must pull the appropriate pieces from the text using regex.

To do so, you will complete the following functions in HW5.py:

1. create_names_dict(*lines_list*)

This function returns a dictionary with the keys being numbers (**1 - 10**) and the values being a tuple made up of **each biography subject's first and last name**. You can assume that all names are made up of two words (first and last name). This function should use a regular expression to find the biography pattern and then add each name to a dictionary with the keys representing that name’s position in the list of biographies and the values being the tuples of names.

The expected output should be in the format:
{1: (“Mike”, “Kearney”), 2: (“Margit”, “Symo”), ... }

2. find_compound_words(*lines_list*)

This function finds all **hyphenated compound words** used in the text file and then returns them in a list. You can assume that a word counts as a hyphenated compound word if it includes letters before and after one or more hyphens. Some examples of valid hyphenated compound words might include: “long-term”, “up-to-date”, etc. Note that there are hyphens present in the text file that don’t meet these criteria, such as “0-9” or “1938 - December 2003”.

3. find_long_words(*lines_list*)

This function finds all **words between 12 and 20 letters** (both inclusive) used in the text file and returns a tuple. The long words should only be made up of letters and ***not*** *contain any punctuation*. The first value of the tuple should be a list of all long words found and the second value should be the average length of those found words. *When computing the average, you should use floor division.*

For example, in the sample “When I look through my kaleidoscope, I feel transcendental” would return ([‘kaleidoscope’, ‘transcendental’], 13).

4. create_short_bios(*lines_list*)

This function returns a list containing **the first sentence of each person's bio**. We define a sentence as a string that begins with a capital letter and ends with one of '***.***', '***!***', or '***?***'. Capture the first sentence that occurs directly after "Bio: ". There should not be a space before the first word, and the ***'.', '!', or '?'* should** be captured. For example,

```
Name: Rani Devi
Born: unknown
Died: unknown
Bio: Rani Devi is a member of the Indian Women's Hockey Team. She
was named the Most Promising Young Player of the Tournament at the
conclusion of the second qualifier. ==External links== * Rani Devi On
Fire * The real 'Chak De' story
```

should capture 'Rani Devi is a member of the Indian Women's Hockey Team.'

5. Make at least 2 test cases each for **create_names_dict**, **find_compound_words**, **find_long_words**, and **create_short_bios**.

Extra Credit (3 points):
Write a function **calculate_in_middle(*mid, lines_list*)** to return a count of the number of times a specified string (***mid***) appears in a file. It should match the string that is in the middle of a word (not the beginning or the end). For example, if called with "be" it should match "number", but not "vibe" or "become". The word itself should ***not*** contain any punctuation, for example, "vibe." or "Elizabeth's" should not be counted. You MUST use a regular expression to earn credit for this part. (We will not be checking if you make tests for the extra credit, but feel free to write your own tests if it will help you complete this problem!)

Grading Rubric (60 points)
This rubric does not show all the ways you can lose points. For each of the functions (**create_names_dict, find_compound_words, find_long_words, create_short_bios**), you can earn:

- 6 points for creating tests for each function (3 points per test case)
- 9 points for correctly implementing each of the 4 functions

Submission:
Make at least 4 git commits and turn in your GitHub repo URL on Canvas by the due date to receive credit.