Homework 5
SI 206, Fall 22

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. **find_bio_names(string_list)**

This function returns a dictionary with the keys being numbers (1 - 10) and the values being the names of each biography subject. 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 names themselves.

The expected output should be in the format:

{1: "Mike Kearney", 2: "Margit Symo", … }

2. **find_possessives(string_list)**

This function finds all possessives used in the text file and then returns them in a list. A word counts as a possessive if it includes letters before and after an apostrophe. Valid possessives might include: Holden's, Julia's, etc. Note that there are many apostrophes present in the text file that don't meet these criteria.

3. **find_section_headings(string_list)**

This function finds and returns all the section headings which match the following conditions:

1. The heading should start with 2 or more equals signs, like "=="
2. There are words between the equals signs
3. The heading ends with 2 or more equals signs, like "=="
4. The heading has the same number of equals signs on each side

These are examples of valid section headings:
==Albums==
===Producer Compilation Albums===

5. **find_birth_years(string_list)**

This function returns a dictionary where the keys are the names of the biography subjects and the values are integers representing each subject's year of birth. Where the year of birth is unknown, you should save the string 'unknown' instead of a year.

Example:

{'Mike Kearney': 1953, 'Alexander Champion': 'unknown', etc.}

6. Make at least 2 test cases each for **find_bio_names**, **find_contractions_and_possessives, find_section_headings**, and **find_birth_years**.

Extra Credit (3 points):

Write a function ***count_mid(string_list, middle)*** to return a count of the number of times a specified string 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". Make sure to account for punctuation (e.g., ',' or '?') in your regular expression. 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 (**find_bio_names, find_contractions_and_possessives**, **find_section_headings**, and **find_birth_years)**, you can earn:

- 6 points for creating tests (2 points per test case, up to a max of 6 points)

  to assess that function

- 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.