CSE 2050 – Programming in a Second Language

Fall 2023

Homework 3: Python Programming with Files, Sets and Dictionaries

Total Points: 30

Date Assigned: Friday, Oct 6, 2023   **Due Date:** Sunday, Oct 15, 2023

Submission Instructions: Please submit your work on Canvas as a Jupyter Notebook `ipynb` file named `cse2050_yourname_hw3.ipynb`. Remember to only use concepts covered up to Week 7 of classes to solve the problems. For example, do not use packages outside of the Python standard library or other concepts not covered in class. See the *Tips for Success* and *Course Requirements* for more information. Finally, make sure to execute the code in your notebook and show both the code and output.

Key Concepts Demonstrated

- Writing code that processes data in files
  - Reading and processing text files (Q1, Q3)
  - Reading and processing CSV files (Q2)
- Formatting and tabulating data (Q1, Q3)
- Text Processing
  - Using regular expressions to split a book into chapters (Q3)
  - Computing term frequencies (TF) in a document (Q3)
  - Creating Text-based graphics (Q2)

1. (10 points) **Alice in Wonderland**
   Note that the file, https://www.gutenberg.org/files/11/old/alice30.txt, is divided into chapters. First, use web scraping techniques to retrieve the text file from the web. Update the code provided on Slide 27 and 28 from the lecture for week 7 to read and print out the unique words in each chapter with the aid of a dictionary and a set (without hardcoding the chapters/titles). Do not include stop-words in your count based on the list of stop-words in the following file: https://cse2050.drfitz.fit/data/very/stop_words.txt. Print a count of unique words you find in each chapter. Do not print out your dictionary.

   Example dictionary (in-memory):

   ```
   {
   "Chapter I: Down the Rabbit-Hole": {word1, word2, word3},
   "Chapter II: The Pool of Tears": {word1, word2, word3},
   "Chapter III: A Caucus-Race and a Long Tale": {word2, word3, word5},
   ...
   }
   ```

   Example Output

   ```
   CHAPTER          CHAPTER TITLE                    UNIQUE WORD COUNT
   Chapter I        Down the Rabbit-Hole                     x
   Chapter II       The Pool of Tears                        y
   Chapter III      A Caucus-Race and a Long Tale            z
   ...
   ```

2. **Vocabulary Improver**
   Sometimes a person can get stuck with using the word "very" a lot. Several graphics appear online showing how one could improve their vocabulary by minimizing their use of the word "very". Figure 1 shows an example:

   Figure 1: An example of *very substitutes*

   (a) (5 points) Given
       a list of `very-substitutes`, https://repl.it/@fnembhard/very#very_words.txt, write Python code that stores this list in a dictionary of the form {"bad": "Awful", "good": "Excellent", ... }. Do not print out your dictionary. Prompt the user for a sentence and use your dictionary to help improve the user's vocabulary. For example "Baby, this LED light makes the room very bright" may become "Baby, this LED light makes the room luminous."

(b) (5 points) Use your code (function) from Part (a) to help improve the sentences in the file https://repl.it/@fnembhard/very#very_sentences.txt. Print out both sentences (original and improved) for all sentences that can be improved by highlighting the replaced phrase using brackets. For example:
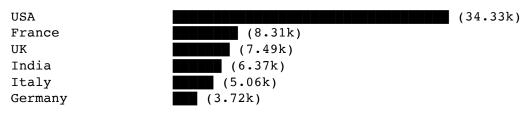
```
Input: "We're very happy together"
Output: "We're [ecstatic] together"
```

Finally, print out a count of the number of sentences you were able to improve in the form `improved/total_sentences`.

3. (10pts points) IMDB Movies
Answer the following question based on the `IMDB_Movies.csv` file available on Canvas. Create and print out a **text-based** horizontal bar chart of **the top 10 countries** with the highest number of movies in the `IMDB_Movies.csv` file. You may use an extended ASCII character* to create your chart. For scaling, let 1 ASCII character = 1000 movies. **Note that this question requires a text-based ASCII graphic. Do not use a plotting package such as Matplotlib to answer this question.**

Example Text-Based Bar Chart

```
USA       ████████████████████████████████████████ (34.33k)
France    ████████ (8.31k)
UK        ███████ (7.49k)
India     ██████ (6.37k)
Italy     █████ (5.06k)
Germany   ███ (3.72k)
```

---

*The character used in the chart above is number 219. See https://theasciicode.com.ar/