Assignment #2

The file named “QA_Pairs.txt” is a corpus created by Dr. Vivian Hu’s research team. It consists of the question answer pairs which are generated from the Game Of Thrones TV series across the eight seasons. Each answer is paired with a question. This corpus can be applied in many fields, such as text classification, text generation, text summarization, sentiment analysis, chatbot in information retrieval and natural language processing. Note that please cite and acknowledge the research team if you would like to use this corpus in other projects.

Input:
QA_Pairs.txt

Requirements:

A. Write your functions to the following tasks. More than a function is permitted for each task.

B. Do not write the same function in each task, import it if you need.

C. Write at least 2 test cases for EACH task.

D. Treat lowercase and uppercase equally.

E. Formats of the outputted files are given.

The tasks:

1. How many QA pairs in QA_Pairs.txt? Here (q1, a1) is a pair, where q stands for question, and a for answer.
2. Are these pairs unique? For example: (q1, a1) (q1, a1) are identical and overlapping; (q1, a1) (q1, a2) are overlapping, and (q1, a1) (q2, a1) are overlapping as well. If not unique, find the overlapping pairs, and generate a unique_QA_Pairs.txt file and an Overlapping.txt file. The format of unique_QA_Pairs.txt and Overlapping.txt are the same as QA_Pairs.txt.

   For (q1, a1) (q1, a1), keep (q1, a1) once; put (q1, a1) in Overlapping.txt.
   For (q1, a1) (q1, a2) and (q1, a1) (q2, a1) , keep (q1, a1) (i.e., the first occurrence of a pair with q1) and delete the others; put (q1, a1), (q1, a2) and (q2, a1) in Overlapping.txt.

   If the original pairs are all unique, rename the original QA_Pairs.txt to be unique_QA_Pairs.txt, and still submit Overlapping.txt as an empty file.
3. Store the pairs from unique_QA_Pairs.txt as a dictionary.
4. Extract all questions in a file called Questions.txt. Format sample is given as Questions.txt.
5. Extract all answers in a file called Answers.txt. Format sample is given as Answers.txt.
6. Find the term frequency of each word (that is, the count of each word) in unique_QA_Pairs.txt, and output the frequencies as Frequency.txt. Format sample is given as Frequency.txt.
7. Rank the words by the decreasing order of their frequencies and output them as Decreasing_Frequency.txt. The format is the same as Frequency.txt.

Submission:

I. There is a .py file for each task, in total there are 7 .py files. Don’t forget your test cases in the corresponding .py file. Name your .py file as t1.py, t2.py …, t7.py.

II. There are 7 files which are generated in the tasks, as:

i. unique_QA_Pairs.txt
ii. Overlapping.txt
iii. QA dictionary
iv. Questions.txt
v. Answers.txt
vi. Frequency.txt
vii. Decreasing_Frequency.txt