Sentence Reconstruction

OVERVIEW

In this assignment, you are given a text file with one sentence per line, however, the whitespaces from the sentences are missing. You are going to design and develop an algorithm to restore the lost whitespaces. Fortunately, all or most of the individual words from the original sentences are available in a separate file that you have full access to, one word per line.

SCOPE

Note: refer to the sample output for references.

1. You are given 2 files, sentence.txt and word.txt.
2. File “sentence.txt” contains the sentences (without whitespaces), one per line, to be restored.
3. File “word.txt” contains most of the individual words from the original sentences, one per line, all lowercase.
4. Reconstruct each sentence from file “sentence.txt” based on the word from file “word.txt”.
5. Two possible outcomes: a sentence can be either fully reconstructed or partially reconstructed
6. Given words from “word.txt” as follows:
    a. “fine”, “this”, “is”, “fine”, “a”, “i”, “beautiful”, “day”, “cannot” and “be”.
7. A sentence is fully reconstructed when ALL the words from the sentence are found in word file.
    a. print out the results with a single whitespace inserted in between words
    b. Ex: “iamfine” -> “I am fine”
    c. Ex: “thisisabeautifulday” -> “this is a beautiful day”
8. If a sentence cannot be fully restored, return the “raw” sentence as it is.
    a. Ex: “cannotberestored” -> “cannotberestored”
    b. Ex: “thisisanotherbeautifulday” -> “thisisanotherbeautifulday”
9. Save ALL results to another separate file in the order that were read from file “sentence.txt”
10. To handle punctuation marks:

    a. . (period)
    b. ; (semi-colon)
    c. : (colon)
    d. - (dash)
    e. ? (question mark)
    f. ! (exclamation mark)
    g. , (comma)
    h. “ (double quote)
    i. ‘ (single quote)

j. ((open round bracket)
k.) (close round bracket)

Examples: “grape,orange,anddark-chocolate.” -> “grape, orange, and dark-chocolate.”

SKILLS

In this assignment, you will be trained on the use of the followings:

- File IO – read and write
- List – storage and tracking
- String indexing and slicing
- List comprehensions
- Controls – if, while, for to control program flow
- Variable Scope
- Functions

DELIVERABLES

1. Design documentation (A2_School_StudentID_Design.doc)
2. Program source code (A2_School_StudentID_Source.py)
3. Output (A2_School_StudentID_Output.txt)

Zip all files above in a single file (A2_School_StudentID.zip) and submit the zip file by due date to the corresponding assignment folder under “Assignment (submission)”

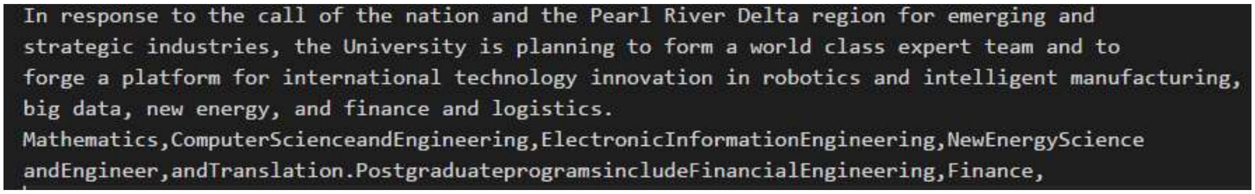
DUE DATE

Refer to website

TIPS & HINTS

- Start word matching from left to right, track the word boundaries as it iterates the sentence
- Use list to track the word boundaries
- Use zip() to generate the word boundary

SAMPLE OUTPUT



```
In response to the call of the nation and the Pearl River Delta region for emerging and
strategic industries, the University is planning to form a world class expert team and to
forge a platform for international technology innovation in robotics and intelligent manufacturing,
big data, new energy, and finance and logistics.
Mathematics,ComputerScienceandEngineering,ElectronicInformationEngineering,NewEnergyScience
andEngineer,andTranslation.PostgraduateprogramsincludeFinancialEngineering,Finance,
```

Note: the sample output shows first 4 lines were fully reconstructed and last 2 lines weren’t.

MARKING CRETERIA

Scope + Deliverables = 75%

Program style & structure = 25%