AC50002 PROGRAMMING LANGUAGES FOR DATA ENGINEERING - PYTHON ASSIGNMENT

Surya Bolisetti – 2545137

Description: This Python script generates word abbreviations by considering the least valued letters based on 'values.txt' and a set of rules. It reads letter values, processes input words from a file, and produces abbreviations in Upper-case with associated scores. The abbreviation with the least score according to letter values and rules is identified. This is selected as the acceptable abbreviation among others. The output, which includes original words and their corresponding abbreviations, is stored in 'surname_filename_abbrevs' format. The script ensures clarity and conciseness through optimized loops, and modular functions, improving readability.

Git-hub link: https://github.com/Chanhakya/Python-Word-abbreviator.git

Design: The code features the following key functions

1. File Reader:
   - Reads values from a specified file, creating a dictionary mapping letters to values.
   - Sorts and returns sorted values dictionary.
2. Least Valued Letter function:
   - Identifies least valued letter and its score in a word.
   - Initializes variables to track the least letter and its score.
   - Iterates through each letter, updating the least letter and score accordingly.
3. Score Checker:
   - Checks least valued letters and scores for each word.
4. Abbreviator:
   - Generates Abbreviation and score for a word.
   - Handles single-word, multiple-word, and two-word scenarios.
5. Main Function:
   - Orchestrates the process.
   - Obtains letter values.
   - Processes input handles quotes and writes words and abbreviations into an output file.

The code also ensures user interaction by querying filename and surname from user. It also handles cases where single lettered words, multiple lettered words and quotes are involved in words.

Description of Files:

A. Source files:
- trees.txt – file with names of trees which need to be abbreviated.
- values.txt – file with values for each individual alphabets

B. Python file:
- Abbreviator-Program.py – Python code file for the main program

C. Test file:

- Test File.txt – File with 5 words which help in testing rules for the abbreviations.
- Case-Test file_abbrevs.txt – Output for test file using main program.

D. Output File:

- bolisetti_trees_abbrevs.txt: Final output file for the program using trees.txt

Execution of code:

Upon execution of code, prompt to provide the input file name (in this case trees.txt) and surname appears. The output will be saved in the format 'surname_filename_abbrevs.txt'. The 'trees.txt' file is available in the same GitHub repository.

Testing of the program:

The program was tested by checking Test file.txt with the program. Testfile.txt had 5 words under special conditions which ensured that all rules have been met by the program. The abbreviations of these 5 words have been manually calculated.

Test Word	Manual Abbreviation	Output Abbreviation
Cold	CLD	CLD
abbreviation	AON	AON
Py	'blank'	'blank'
Moore's Law	MSL	MSL
Object-oriented programming	OOP	OJP

Thus, the code has been tested and the output has been compared with expected results, ensuring that program runs successfully and gives satisfactory results for 4 cases out of 5.
The program did not give expected results for the hyphenated word case of Object-oriented programming.
To correct this issue, the code has been modified to replace the hyphen with a space, so that the string does not have the Hyphen, but this method was not fruitful.

References:

- Python videos from MyDundee for functions.
- Assignment file for inputs in Test file.txt.
- https://books.goalkicker.com/PythonBook/