**AC50002 Programming Languages for data Engineering Python Assignment**

In this assignment understanding the problem is key to knowing the kind of functions and module to use in the code. I wrote this script to generate abbreviations from a list of words, score the abbreviations based on certain rules and write the results to an output file.

I started by importing four different modules. ‘os’ module for file and directory operations, ‘re’ for regular expressions, ‘defaultdict’ from ‘collections’ for creating dictionaries with default values, and ‘traceback’ for printing exception traceback information.

**File Reading Functions:**

read\_file(file\_path): this line reads the content of a file.

read\_values(file\_path): this line of code reads key-value pairs from a file.

txt\_abbrevs(file\_path): this line reads a list of abbreviations from a file.

I proceeded to generate functions to satisfy the given conditions.

**Functions**

**Word Processing Functions:**

process\_words(word\_list): Processes words by removing non-alphabetic characters and converting them to uppercase.

join\_words(word\_list): Joins processed words.

**Abbreviation Generation Function:**

gen\_abbrevs(word\_list): Generates abbreviations for a list of processed words using generate\_abbreviations.

**Abbreviation Generation Logic:**

generate\_abbreviations(word): Generates abbreviations for a given word.

remove\_duplicates\_from\_dict(duplicates, abbreviations\_dict): this removes the duplicate abbreviations from the dictionary.

**Word Indexing:**

wordindex(input\_string): this line creates a dictionary mapping characters to their positions in a word.

**Scoring Functions:**

getAbbreviationIndex(result, abbreviation): Retrieves the positions of characters in an abbreviation.

score\_abbreviation(score\_card, abbrevs\_dict): Scores abbreviations based on position and letter scores.

**Main Function:**

main(): the main function takes the user input for a file name, reads values and abbreviations, processes and generates abbreviations, scores them, and writes the results to an output file.

I used the example in the assignment booklet as a Chrisidisi\_testfile.txt to run it at first before bringing in trees.txt tile. The example contains three words. Cold, cool and C++ Code. Although I had some issues trying to get the final answer to give me the abbreviation with the lowest score. I tried using the sorted function to arrange the abbreviations with their respective scores in an ascending order and displaying the first abbreviation but it wasn’t providing the desired result as at the time of carrying out this project.

**Execution:**

The script is executed if it's the main module (if \_\_name\_\_ == "\_\_main\_\_":).

**Exception Handling:**

The script includes basic exception handling to print traceback information in case of errors.

**Output File:**

The final results are written to a file named in the format "Chrisidisi\_<filename>\_abbrevs.txt."

References

1. The python tutorial (no date) Python documentation. Available at: <https://docs.python.org/3/tutorial/index.html>.
2. Python tutorial. Available at: https://www.w3schools.com/python/default.asp .