**Pseudocode for Spell Checker Program**

This program is a simple spell checker. It functions by reading a list of correct words from a dictionary file (wordlist.txt) into a data structure for quick searching. It then opens a second file (reference\_doc.txt), reads each word from it, and cleans it by removing punctuation and converting it to lowercase. Finally, it compares each cleaned word against the loaded dictionary. If a word is not found in the dictionary, the program identifies it as misspelled and displays it on the screen.

**Main Program**

1. Load Dictionary from a plain text file (one word per line)

a. Define a path to the dictionary file (wordlist.txt).

b. Initialize an empty set of strings called dictionary\_words for efficient word lookup.

c. Open the dictionary file.

d. If the file fails to open, print an error message and exit the program.

e. Read each word from the file, convert it to lowercase, and insert it into the dictionary\_words set.

f. Close the dictionary file.

1. **Handle Input Document**

a. Define a path to the input document file (reference\_doc.txt).

b. Check if the input file exists.

c. If the file does not exist, create it and write some sample text with deliberate spelling errors to it.

d. Open the input document file for reading. If the file fails to open, print an error message and exit the program.

1. **Check Spelling**

a. Initialize an empty set of strings called misspelled\_words to store and prevent duplicate output of misspelled words.

b. Read the input document file word by word.

c. For each word read:

i. Call a clean\_word function to remove punctuation and convert the word to lowercase.

ii. If the cleaned word is not empty and is not found in the dictionary\_words set:

* + If the cleaned word is not already in the misspelled\_words set:
  + Print the word as "Misspelled".
  + Insert the cleaned word into the misspelled\_words set.

1. **Finalize**

a. Close the input document file.

b. Print a final message to the user that the document has been checked.

**Helper Functions**

**Function: toLower(s) -> returns lowercase string**

1. Take a string s as input.
2. Create a new string.
3. Iterate through the input string and convert each character to its lowercase equivalent.
4. Return the new lowercase string.

**Function: clean\_word(s) -> returns cleaned string**

1. Take a string s as input.
2. Create an empty string cleaned\_s.
3. Iterate through each character c in the input string s.
4. If c is an alphabetic character, convert it to lowercase and append it to cleaned\_s.
5. Return the cleaned\_s.