**Kotlin/Java Developer Test Task: Text Analysis Program v2**

***Task Description:***

Develop a program in **Kotlin** or **Java** that analyzes a given text file and provides statistical insights about the words used in it.

***Requirements:***

1. **Input**:
2. Use the text file provided at the following URL as input: [moby.txt](https://courses.cs.washington.edu/courses/cse390c/22sp/lectures/moby.txt).
3. The program should read and analyze this file.
4. **Functionality**:
5. **Count Words**: Calculate and display the total number of words in the text.
6. **Top 5 Words**: Identify and display the top 5 most frequently used words, along with their usage counts. Exclude prepositions, pronouns, conjunctions, and articles from this analysis.
7. **Unique Words**: List all unique words in alphabetical order, excluding prepositions, pronouns, conjunctions, and articles.
8. **Exclusions**:
9. Ignore prepositions (e.g., "in", "on", "at"), pronouns (e.g., "he", "she", "it"), conjunctions (e.g., "and", "or", "but"), and articles (e.g., "the", "a", "an"). A predefined list of such words should be used in the program for filtering.
10. Ignore plural postfix “'s”
11. Ignore modal verbs and their forms (“is”, “was”)
12. **Case Insensitivity**:
13. Words should be treated as case insensitive. For example, "Hello", "hello", and "HELLO" should be considered the same word.
14. **Output**: The program should produce the following:
15. Total word count (excluding the filtered words).
16. Top 5 most frequent words with counts (excluding the filtered words).
17. Alphabetically sorted list of all unique words (excluding the filtered words) - top 50.
18. **Demonstrate Results**:
19. Before the interview, run your program on the provided file and prepare the output.
20. Save the output to a separate file or display it in the console for presentation during the interview.
21. Mention the actual time you spent on the task, even if it exceeds or falls short of 2 hours.
22. **Submit the Code**:
23. Share the program code in any git repo.
24. Name the GIT repo as pattern <First Part>\_<Second>\_<Third>
25. First Part – is first most used word that you found
26. Second – is fifths most used word that you have found
27. Third – processing time in seconds to create required Output.
28. Example: carpet\_bag\_10s
29. ensuring it is well-commented and easy to understand.

***Constraints:***

* Assume that words are sequences of letters separated by non-letter characters (e.g., spaces, punctuation).
* Special characters, numbers, and punctuation should not be considered part of a word.
* Use a reasonable predefined list for exclusions. You do not need to handle all possible words in these categories.

***Language Requirement:***

The program must be implemented in **Kotlin** or **Java**.

***Time Limit:***

This task is estimated to take **around 2 hours**. This is not a strict time limit, but please document the actual time spent.