

Overview

This program is designed to identify the longest compound word and the second-longest compound word in a given list of words. A compound word is defined as a word formed by combining two or more shorter words from the same list.

The program processes two input files (Input_01.txt and Input_02.txt), each containing a list of words, and outputs the required compound words along with the processing time for each file.

Approach and Logic

1. Input Handling

The program reads words from the input file using a Scanner. Each word is trimmed of whitespace and stored in a list.

2. Sorting Words

The word list is sorted in descending order of word length. This ensures that the longest words are checked first, making it easier to identify the longest compound words early.

3. Compound Word Detection

A helper method `isCompoundWord` recursively checks whether a given word can be formed by combining shorter words from the list:

Base Case: If the word exists in the set (and it's not the original word being checked), it is considered valid.

Recursive Step: The word is split into a prefix and suffix. If the prefix exists in the set and the suffix can also be broken down into valid components, the word is a compound word.

4. Result Storage

The program keeps track of the longest and second-longest compound words as it iterates through the sorted list.

5. Timing

The time taken to process each file is calculated and displayed.