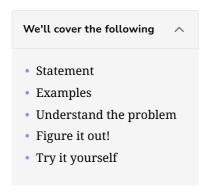
Design Add and Search Words Data Structure

Try to solve the Design Add and Search Words Data Structure problem.



Statement

Design a data structure called **WordDictionary** that supports the following functionalities:

- Constructor: This function will initialize the object.
- Add Word(word): This function will store the provided word in the data structure.
- Search Word(word): This function will return TRUE if any string in the WordDictionary object matches the query word. Otherwise, it will return FALSE. If the query word contains dots, , each dot is free to match any letter of the alphabet.

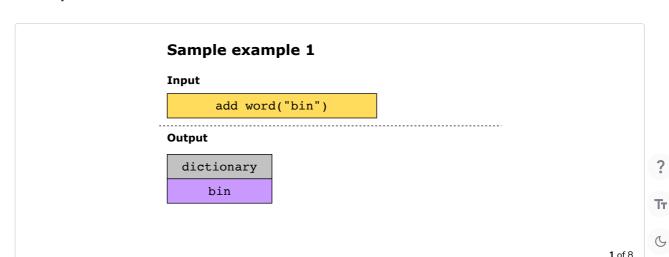
For example, the dot in the string ".ad" can have 26 possible search results like "aad", "bad", "cad", and so on.

• **Get Words():** This function will return all the words in the **WordDictionary** class.

Constraints:

- $1 \leq \mathsf{word.length} \leq 25$
- Words passed to Add Word() consist of lowercase English letters.
- Words passed to **Search Word()** consist of . or lowercase English letters.
- There will be, at most, three dots in a word passed to **Search Word()**.
- At most, 10^3 calls will be made to Add Word() and Search Word().

Examples





Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

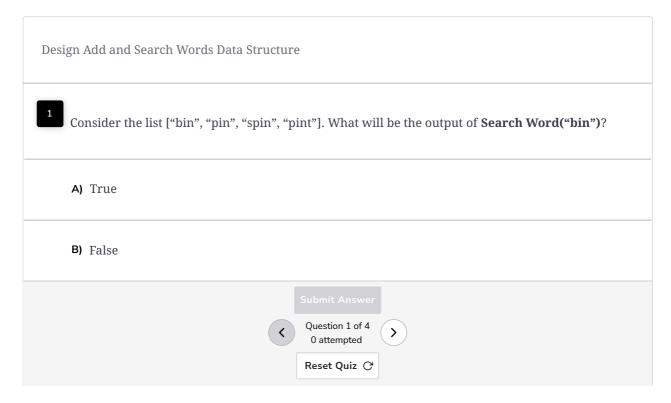
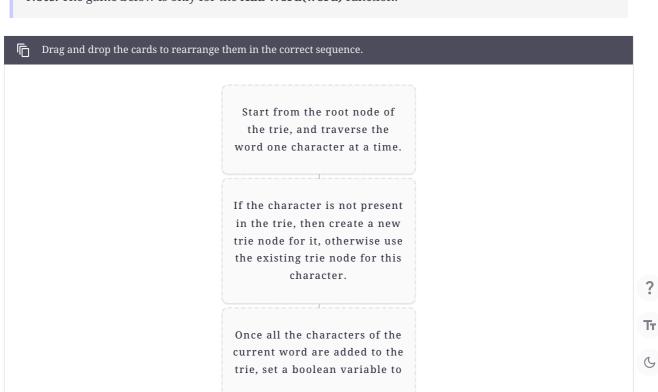
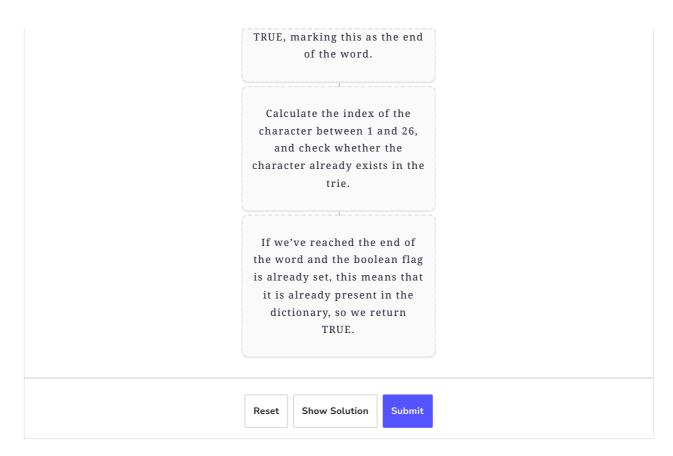


Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

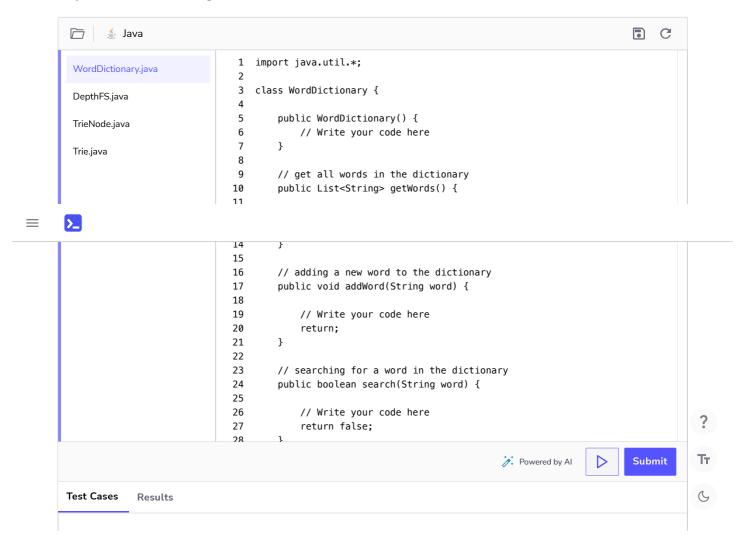
Note: The game below is only for the Add Word(word) function.



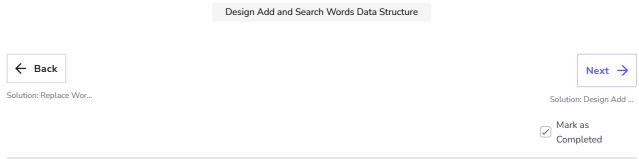


Try it yourself

Implement your solution in WordDictionary.java in the following coding playground. You will need the provided supporting code to implement your solution. We have also provided a useful code template that you may build on to solve this problem.







 Tr