Problem J. HangmanSolver

Problem Statement

In the game Hangman one player thinks of a secret word and the other player attempts to guess it.

At the beginning of the game, the player who has the secret word reveals its length. In each turn of the game the guessing player may guess any letter of the alphabet. Each time they do so, the player who knows the secret word reveals the precise positions of all occurrences of the guessed letter in the secret word.

An example game:

- At the beginning the guessing player sees "______": six unknown letters.
- They guess the letter 'x'. The other player reveals that there are no 'x' in the word.
- At this moment, the guessing player can eliminate some possible answers. For example, the answer cannot be "oxford" because "oxford" contains an 'x' while the word we are guessing does not.
- Next, they guess the letter 'a'. The other player reveals that there are two 'a' in the word: "_a___a".
- Again, the guessing player can eliminate some possible answers. For example, the answer cannot be "rwanda" because the second letter of "rwanda" is 'w', not 'a'. Also, the answer cannot be "canada" because all the 'a's in the unknown word have already been revealed and "canada" has an extra 'a'.
- At this moment, the answer could still be "camera", "lambda", or "saliva", for example.

Your friend came up with a secret word and you are trying to guess it. You are given:

- The String **lettersGuessed** with the letters you already guessed.
- The String **revealed**: the word you are guessing. This string was obtained from the word you are guessing: All occurrences of the letters you already guessed are revealed. All other letters of the secret word are replaced by underscores ('_').
- the String[] **possibleAnswers**. You know the other player and you expect that the secret word is one of the words in this list.

If the revealed part of the word matches *exactly one* of the options in **possibleAnswers**, return that option. Otherwise, return an empty String.

Definition

Class: HangmanSolver
Method: guess
Parameters: String, String, String[]
Returns: String
Method signature:
String guess(String lettersGuessed, String revealed, String[] possibleAnswers)
(be sure your method is public)

Constraints

- **lettersGuessed** will have between 0 and 25 characters, inclusive.
- All characters in **lettersGuessed** will be lowercase English letters ('a'-'z').
- All characters in **lettersGuessed** will be distinct.
- **revealed** will have between 1 and 10 characters, inclusive.
- All characters in **revealed** will be lowercase English letters ('a'-'z') and underscores ('_').
- **possibleAnswers** will have between 1 and 50 elements, inclusive.
- Each element of **possibleAnswers** will have the same length as **revealed**.
- All characters in **possibleAnswers** will be lowercase English letters ('a'-'z').
- All elements of **possibleAnswers** will be distinct.

Examples

0)

```
""

"________"

{"topcoder"}

Returns: "topcoder"
```

If there is only one possible answer, you can guess it even before you have guessed any letters.

1)

```
"grzn"

"_r_ng_"

{"banana", "cherry", "potato"}

Returns: ""
```

Your friend has clearly chosen a secret word that is not on your list.

2)

```
"a"

"_a_a_a"

{"banana", "cherry", "potato"}

Returns: "banana"
```

Out of the three options, only "banana" matches the revealed part of the word.

3)

```
"wyxzd"
```

```
"______"
```

```
{"banana", "cherry", "potato"}
```

```
Returns: ""
```

We know that the answer isn't "cherry" because you already guessed a 'y' and nothing got revealed. The other two candidate answers are still plausible. As we have more than one possible answer, we return an empty string.

This problem statement is the exclusive and proprietary property of TopCoder, Inc. Any unauthorized use or reproduction of this information without the prior written consent of TopCoder, Inc. is strictly prohibited. (c)2022, TopCoder, Inc. All rights reserved.