26 Autocomplete

13 points

Introduction

As you may know, most text messaging apps use autocomplete mechanisms in order to predict what you are trying to type. Commonly the autocomplete is also customized according to the user's usual speech patterns.

We are developing what we expect to be the next most popular text messaging app, and this feature is a must!

Those mechanisms can be based in very complex algorithms, but since this is our first version, let's make it simpler:

The user will have typed some text to which we have access. He or she has also started typing the next word, the one we must predict.

Taking into account the contents of the given text, our app must suggest the next word to be typed.

To clarify how this should work, our manager has given us the following rules:

- The suggested word must start with the same characters the user has typed.
- If there is no word that satisfies the previous rule, we suggest the same 'word' the user has typed.
- If we have more than one word starting with the user's input, we suggest the most common one.
- In case two or more suggestions are equally common, we must suggest the one that comes first in alphabetical order.

Input

- Text in one line.

HINT: be careful with capital letters ('Printer' must be treated the same as 'printer'). Also, we can expect to only have alphabetical characters.

- The word we want to autocomplete. To make things a little bit easier, the word we want to autocomplete will only contain lowercase letters.

Output

Suggestion for the input word, in lowercase.

Example 1

Input

```
How many cookies could a good cook cook If a good cook could cook cookies
coo
```

Output

```
cook
```

Example 2

Input

```
this is just a normal text
hou
```

Output

```
hou
```

Example 3

Input

```
animal animal animal animate animate animate
anima
```

Output

```
animal
```