题目描述：

给定一个连续不包含空格字符串，该字符串仅包含英文小写字母及英文文标点符号(逗号、分号、句号)，同时给定词库，对该字符串进行精确分词。

说明：

1.精确分词： 字符串分词后，不会出现重叠。即"ilovechina" ，不同词库可分割为 "i，love，china" "ilove，china"，不能分割出现重叠的"i，ilove，china",i 重叠出现

2.标点符号不成词，仅用于断句

3.词库：根据外部知识库统计出来的常用词汇例：

dictionary=["i","love","china","lovechina","ilove"],

4.分词原则：采用分词顺序优先且最长匹配原则

"ilovechina"，假设分词结果 [i,ilove,lo,love,ch,china,lovechina] 则输出 [ilove，china]

错误输出：[i,lovechina], 原因："ilove ">优先于 "lovechina"成词

错误输出：[i,love,china] 原因："ilove" >"i" 遵循最长匹配原则

输入描述：

字符串长度限制：0<length<256

词库长度限制： 1<length<100000

第一行输入待分词语句 "ilovechina"

第二行输入中文词库 "i,love,china,ch,na,ve,lo,this,is,the,word"

输出描述：

按顺序输出分词结果 "i,love,china"

示例 1

输入：

```
ilovechina

i,love,china,ch,na,ve,lo,this,is,the,word
```

输出：

```
i,love,china
```

说明：

示例 2

输入：

```
iat

i,love,china,ch,na,ve,lo,this,is,the,word,beauti,tiful,ful
```

输出：

```
i,a,t
```

说明：

单个字母，不在词库中且不成词则直接输出单个字母

示例 3

输入：

```
ilovechina,thewordisbeautiful
i,love,china,ch,na,ve,lo,this,is,the,word,beauti,tiful,ful
```

输出：

```
i,love,china,the,word,is,beauti,ful
```

说明：

标点符号为英文标点符号

# 字典树

```
class Node:
    def __init__(self, c, word=None):
        self.char = c
        self.word = None
        self.hashmap = {}

class Tree:
    def __init__(self):
        self.root_map = {}

    def add_word(self, word):
        hashmap = self.root_map
        for i, c in enumerate(word):
            if c not in hashmap:
                hashmap[c] = Node(c)
            if i == len(word) - 1:
                hashmap[c].word = word
            hashmap = hashmap[c].hashmap


text = input()

tree = Tree()
ciku = input().split(',')
for word in ciku:
    tree.add_word(word)

result = []
left = 0
while left < len(text):
    if not ('a' <= text[left] <= 'z'):    #  标点符号，看下个
        left += 1
        continue

    #  找最长匹配
    match = None
    hashmap = tree.root_map
    current = left
    while current < len(text) and text[current] in hashmap:
        node = hashmap[text[current]]
        if node.word is not None:
            match = node.word
        hashmap = node.hashmap
```

```
            current += 1

        if match is None:
            match = text[left]
        result.append(match)
        left += len(match)

print(','.join(result))
```