

题目描述：

有一个字符串数组 *words* 和一个字符串 *chars*。

假如可以用 *chars* 中的字母拼写出 *words* 中的某个“单词”（字符串），那么我们就认为你掌握了这个单词。

words 的字符仅由 *a-z* 英文小写字母组成。 例如: *abc*

chars 由 *a-z* 英文小写字母和 “?”组成。其中英文问号“?”表示万能字符，能够在拼写时当做任意一个英文字母。 例如： “?” 可以当做 “a”等字母。

注意：每次拼写时，*chars* 中的每个字母和万能字符都只能使用一次。

输出词汇表 *words* 中你掌握的所有单词的个数。 没有掌握任何单词，则输出 0。

输入描述：

第 1 行输入数组 *words* 的个数，记为 *N*。

从第 2 行开始到第 *N*+1 行依次输入数组 *words* 的每个字符串元素。

第 *N*+2 行输入字符串 *chars*。

输出描述：

输出一个整数，表示词汇表 *words* 中你掌握的单词个数。

补充说明：

注意：

$1 \leq words.length \leq 100$

$1 \leq words[i].length, chars.length \leq 100$

所有字符串中都仅包含小写英文字母、英文问号

示例 1

输入：

4

cat

bt

hat

tree

atach??

输出:

3

说明:

可以拼写字符串"*cat*"、"*bt*"和"*hat*"

示例 2

输入:

3

hello

world

cloud

welldonehoneyr

输出:

2

说明:

可以拼写字符串"*hello*"和"*world*"

示例 3

输入:

```
3
apple
car
window
welldoneapplec?
```

输出:

```
2
```

说明:

可以拼写字符串"*apple*"和"*car*"

```
package main
```

```
import (  
    "fmt"  
)
```

```
func main() {  
    var n int  
    fmt.Scanln(&n)  
    res := 0  
    words := []string{}  
    for i := 0; i < n; i++ {  
        var word string  
        fmt.Scanln(&word)  
        words = append(words, word)  
    }  
    var chars string  
    fmt.Scanln(&chars)  
    charsMap := map[byte]int{}  
    for i := 0; i < len(chars); i++ {  
        c := chars[i]  
        if _, ok := charsMap[c]; ok {  
            charsMap[c]++  
        } else {  
            charsMap[c] = 1  
        }  
    }  
    for i := 0; i < n; i++ {  
        count := 0  
        for j := 0; j < len(words[i]); j++ {
```

```
        c := words[i][j]
        if num, ok := charsMap[c]; ok && num > 0 {
            count++
        }
    }
    if count+charsMap['?'] >= len(words[i]) {
        res++
    }
}
fmt.Println(res)
}
```