

题目描述:

相对开音节构成的结构为辅音+元音 (aeiou)+辅音(r 除外)+e, 常见的单词有 bike、cake 等。

给定一个字符串, 以空格为分隔符, 反转每个单词中的字母, 若单词中包含如数字等其他非字母时不进行反转。

反转后计算其中含有相对开音节结构的子串个数 (连续子串中部分字符可以重复)。

输入描述:

字符串, 以空格分割的多个单词, 字符串长度<10000, 字母只考虑小写

输出描述:

含有相对开音节结构的子串个数, 注: 个数<10000

补充说明:

收起

示例 1

输入:

ekam a ekac

输出:

2

说明:

反转后为 make a cake 其中 make、cake 为相对开音节子串, 返回 2

示例 2

输入:

!ekam a ekekac

输出:

2

说明:

反转后为!ekam a cakeke 因!ekam 含非英文字符所以未反转, 其中 cake、keke 为相对开音节子串, 返回 2

```
import re
_string = input().split(' ')
y_list = ['a','e','i','o','u']
f_list = ['b','c','d','f','g','h','g','k','l','m','n','p','q','r','s','t','v','w','x','y','z']
pattern = [0,1,0,1]
cnt = 0
def check(str, start, end):
    if len(str) < 4:
        return False
    if end - start < 4:
        return False
    # if str[start:end]
    if str[start] in f_list:
        if str[start + 1] in y_list:
```

```
        if str[start + 2] in f_list and str[start + 2] != 'r':
            if str[start + 3] == 'e':
                return True
    return False
```

```
for i in _string:
    _str = i
    if not re.search('.*[^a-zA-Z].*', i):
        # 可以进行反转，以下为反转
        _temp = list(i)
        _temp.reverse()
        _str = ''.join(_temp)
        # 反转结束，用 01 进行标记
    for i in range(len(_str)):
        if i + 3 < len(_str) and check(_str, i, i + 4):
            cnt += 1
```

```
print(cnt)
```