相对开音节

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（连续的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

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 java.util.*;


// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        // 注意 hasNext 和 hasNextLine 的区别
        String[] ss=in.nextLine().toLowerCase().split(" ");
        int ans=0;
        HashSet<Character> hs=new HashSet<>();
        hs.add('a');
        hs.add('e');
        hs.add('i');
        hs.add('o');
        hs.add('u');
        for(String s:ss){
            boolean f=true;
            for(char c:s.toCharArray()){
                if(c<'a'||c>'z'){
                    f=false;
```

```
                    break;
                }
            }
            if(f){
                s=new StringBuilder(s).reverse().toString();
            }
            for(int i=3;i<s.length();i++){
                boolean ff=true;
                if(s.charAt(i-3)<'a'||s.charAt(i-3)>'z'||hs.contains(s.charAt(i-3)))
ff=false;
                if(!hs.contains(s.charAt(i-2))) ff=false;

if(s.charAt(i-1)<'a'||s.charAt(i-1)>'z'||s.charAt(i-1)=='r'||hs.contains(s.charAt(i-
1))) ff=false;
                if(s.charAt(i)!='e') ff=false;
                if(ff) ans++;
            }
        }
        System.out.print(ans);
    }
}
```