Java-字符串-一群大雁往南飞，给定一个字符串记录地面上的游客听到的大雁叫
题目描述：
一群大雁往南飞，给定一个字符串记录地面上的游客听到的大雁叫声，请给出叫声最少由几只大雁发出。具体的：
1. 大雁发出的完整叫声为"quack"，因为有多只大雁同一时间嘎嘎作响，所以字符串中可能会混合多个 "quack"。
2. 大雁会依次完整发出 "quack"，即字符串中‘q’, ’u’, ’a’, ’c’, ’k’ 这 5 个字母按顺序完整存在才能计数为一只大雁。如果不完整或者没有按顺序则不予计数。
3. 如果字符串不是由‘q’, ’u’, ’a’, ’c’, ’k’字符组合而成，或者没有找到一只大雁，请返回 -1。

输入描述：
一个字符串，包含大雁 quack 的叫声。1 <= 字符串长度 <= 10000
字符串中的字符只有'q', 'u', 'a', 'c', 'k'
输出描述：
大雁的数量
补充说明：
示例 1
输入：
quackquack
输出：
1
说明：
一只大雁叫了两次
示例 2
输入：
quqackuack
输出：
2
说明：
最少需要两只大雁分别叫一次，第一只雁子 quqackuack ,第二只雁子 quqackuack
示例 3
输入：
quackquook
输出：
-1
说明：
给出的字符串不是 "quack"的有效组合。

```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.Scanner;

// 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);
        String quack = in.next();
        System.out.println(getNum(quack));

    }

    public static int getNum(String s) {
        LinkedList <Integer> quack = new LinkedList<>();
        ArrayList <Integer[]> ranges = new ArrayList<>();
        int u = 0, a = 0, c = 0, k = 0;
        for (int i = 0 ; i < s.length() ; i++) {
            switch (s.charAt(i)) {
                case 'q':
                    quack.add(i);
                    break;
                case 'u':
                    if (u + 1 <= quack.size()) u++;
                    break;
                case 'a':
                    if (a + 1 <= u) a++;
                    break;
                case 'c':
                    if (c + 1 <= a) c++;
                    break;
                case 'k':
                    if (c >= 1) {
                        ranges.add(new Integer [] {quack.removeFirst(), i});
                        u--;
                        a--;
                        c--;
                    }
                    break;
                default:
                    return -1;
            }
        }
        if (ranges.size() == 0) return -1;
        int res = 1;
        int ans = 1;
        for(int i=0;i<ranges.size();i++){
            int count=1;
            for (int j=i+1;j<ranges.size();j++){
                if(ranges.get(i)[1] >= ranges.get(j)[0]){
                    count++;
```

```
                        res++;
                    }

                }
                ans = Math.max(ans ,count);
            }
            return ans;

        }
    }
```