

字符串摘要

题目描述：

给定一个字符串的摘要算法，请输出给定字符串的摘要值。

1、去除字符串中非字母的符号。

2、如果出现连续字符（不区分大小写），则输出：该字符（小写）+ 连续出现的次数。

3、如果是非连续的字符（不区分大小写），则输出：该字符（小写）+ 该字母之后字符串中出现的该字符的次数。

4、对按照以上方式表示后的字符串进行排序：字母和紧随的数字作为一组进行排序，数字大的在前，数字相同的，则按字母进行排序，字母小的在前。

输入描述：

一行字符串，长度为[1,200]

输出描述：

摘要字符串

补充说明：

示例 1

输入：

aabbcc

输出：

a2b2c2

说明：

示例 2

输入：

bAaAcBb

输出：

a3b2b2c0

说明：

bAaAcBb:

第一个 *b* 非连续字母，该字母之后字符串中还出现了 2 次（最后的两个 *Bb*），所以输出

b2,

a 连续出现 3 次，输出 *a3*,

c 非连续，该字母之后字符串再没有出现过 *c*，输出 *c0*

Bb 连续 2 次，输出 *b2*

对 *b2a3c0b2* 进行排序，最终输出 *a3b2b2c0*

```
import java.util.ArrayList;
```

```
import java.util.Comparator;
```

```
import java.util.Scanner;
```

```
// 注意类名必须为 Main，不要有任何 package xxx 信息
```

```
public class Main {
```

```
    static class Zifuchuan {
```

```
        Character c;
```

```
        int value;
```

```
        public Zifuchuan(Character c, int value) {
```

```
            c = c.toString().toLowerCase().charAt(0);
```

```
            this.c = c;
```

```
            this.value = value;
```

```
}
```

```
}
```

```
public static boolean isEqual(Character c1, Character c2) {
```

```
    if (c1 == c2 || c1 + 32 == c2 || c1 - 32 == c2) {
```

```
        return true;
```

```
    }
```

```
    return false;
```

```
}
```

```
public static void main(String[] args) {
```

```
    Scanner in = new Scanner(System.in);
```

```
    // 注意 hasNext 和 hasNextLine 的区别
```

```
    String s = in.nextLine();
```

```
    StringBuilder sb = new StringBuilder();
```

```
    for(int i=0;i<s.length();i++){
```

```
        if(s.charAt(i)>='a'-32&& s.charAt(i)<='z'+32){
```

```
            Character c = s.charAt(i);
```

```
            c = c.toString().toLowerCase().charAt(0);
```

```
            sb.append(c);
```

```
        }else if(s.charAt(i)>='A'&& s.charAt(i)<='Z'){
```

```
            sb.append(s.charAt(i));
```

```
}
```

```
//System.out.println(sb);
```

```
}
```

```
s = sb.toString();
```

```
//System.out.println(s);
```

```
ArrayList<Zifuchuan> list = new ArrayList();
```

```
for (int i = 0; i < s.length(); i++) {
```

```
    if (i < s.length() - 1 && isEqual(s.charAt(i + 1), s.charAt(i))) {
```

```
        int ji = 1;
```

```
        int j = i+1;
```

```
        for (j = i + 1; j < s.length(); j++) {
```

```
            if (isEqual(s.charAt(j), s.charAt(i))) {
```

```
                ji++;
```

```
            } else {
```

```
                break;
```

```
        }
```

```
    }
```

```
    i=j-1;
```

```
    //System.out.println(s.charAt(i)+" "+ji);
```

```
    list.add(new Zifuchuan(s.charAt(i), ji));
```

```
    } else {
```

```
        int ji = 0;
```

```
for (int j = i + 1; j < s.length(); j++) {
```

```
    if (isEqual(s.charAt(i), s.charAt(j))) {
```

```
        ji++;
```

```
    }
```

```
}
```

```
//System.out.println(s.charAt(i)+" "+ji);
```

```
list.add(new Zifuchuan(s.charAt(i), ji));
```

```
}
```

```
}
```

```
list.sort(new Comparator<Zifuchuan>() {
```

```
    public int compare(Zifuchuan c1, Zifuchuan c2) {
```

```
        if (c1.value < c2.value) {
```

```
            return 1;
```

```
        } else if (c1.value > c2.value) {
```

```
            return -1;
```

```
        } else {
```

```
            if (c1.c < c2.c) {
```

```
                return -1;
```

```
            } else if (c1.c > c2.c) {
```

```
                return 1;
```

```
            } else {
```

```
return 0;
```

```
}
```

```
}
```

```
}
```

```
});
```

```
for (int i = 0; i < list.size(); i++) {
```

```
    System.out.print(list.get(i).c);
```

```
    System.out.print(list.get(i).value);
```

```
}
```

```
    System.out.println();
```

```
}
```

```
}
```