

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

// 本题为考试单行多行输入输出规范示例，无需提交，不计分。

```
import java.util.*;
```

```
public class Main {
```

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

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

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

```
        str = str.replaceAll("[^a-zA-Z]", "");
```

```
        str = str.toLowerCase();
```

```
        StringBuilder jy = new StringBuilder();
```

```
        Map<Character, Integer> map = new HashMap<Character, Integer>();
```

```
        int n = str.length();
```

```
        for (int i = 0; i < n; i++) {
```

```

        char ch = str.charAt(i);
        int j = i+1;
        while (j<n&&str.charAt(j)==ch) {
            j++;
        }
        if (j-i>1){
            jy.append(ch);
            jy.append(j-i);
        } else{
            jy.append(ch);
            if(!map.containsKey(ch)) {
                int count = 0;
                for (int k = j; k<n;k++) {

                    if (str.charAt(k)==ch) {
                        count++;
                    }
                }
                map.put(ch,count);
            }
            jy.append(map.get(ch));
            map.put(ch,map.get(ch)-1);
        }
        i=j;
    }
    List<String> list = new ArrayList<String>();
    n = jy.length();
    for (int i = 0; i < n; i++){
        char ch = jy.charAt(i);
        if (ch >= 'a' && ch<='z') {
            int j = i+1;
            while (j<n && jy.charAt(j)>='0' && jy.charAt(j)<='9') {
                j++;
            }
            list.add(jy.substring(i,j));
            i= j-1;
        }
    }
    Collections.sort(list, new Comparator<String>() {
        public int compare(String a, String b) {
            String astr = a.substring(0,1);
            String anum = a.substring(1);
            String bstr = b.substring(0,1);
            String bnum = b.substring(1);

```

```
        if (anum.equals(bnum)) {
            return astr.compareTo(bstr);
        }else{
            return bnum.compareTo(anum);
        }
    }
});
for (String s:list) {
    System.out.print(s);
}
}
```