

## 字符串统计

题目描述：给定两个字符集合，一个为全量字符集，一个为已占用字符集。已占用的字符集中的字符不能再使用，要求输出剩余可用字符集。 输入描述：1、输入为一个字符串，一定包含@符号。@前的为全量字符集，@后的字为已占用字符集。 2、已占用字符集中的字符一定是全量字符集中的字符。字符集中的字符跟字符之间使用英文逗号分隔。 3、每个字符都表示为字符加数字的形式，用英文冒号分隔，比如 a:1，表示 1 个 a 字符。 4、字符只考虑英文字母，区分大小写，数字只考虑正整数，数量不超过 100。 5、如果一个字符都没被占用，@标识仍然存在，例如 a:3,b:5,c:2@ 输出描述：输出可用字符集，不同的输出字符集之间回车换行。 注意，输出的字符顺序要跟输入一致。不能输出 b:3,a:2,c:2 如果某个字符已全被占用，不需要再输出。

### 示例1

输入：a:3,b:5,c:2@a:1,b:2

输出：a:2,b:3,c:2

说明：全量字符集为3个a，5个b，2个c。

已占用字符集为1个a，2个b。

由于已占用字符不能再使用，因此，剩余可用字符为2个a，3个b，2个c。

因此输出a:2,b:3,c: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
```

```
        while (in.hasNext()) {
```

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

```
            String[] ss = s.split("@");
```

```
            if (ss.length == 1) {
```

```
                System.out.println(ss[0]);
```

```
            } else {
```

```
                String dict = ss[0], used = ss[1];
```

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

```
                String[] ds = dict.split(",");
```

```
                List<String> list = new ArrayList<>();
```

```
                for (String d : ds) {
```

```
                    String[] dd = d.split(":");
```

```
                    list.add(dd[0]);
```

```
                    map.merge(dd[0], Integer.valueOf(dd[1]), Integer::sum);
```

```
                }
```

```

String[] us = used.split(",");
for (String u : us) {
    String[] uu = u.split(":");
    map.merge(uu[0], -Integer.valueOf(uu[1]), Integer::sum);
}
StringJoiner sj = new StringJoiner(",");
for (String key : list) {
    if (map.get(key) == 0) continue;
    sj.add(key + ":" + map.get(key));
}
System.out.println(sj.toString());
}
}
}
}

```