

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，数字只考虑正整形，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.ArrayList;
import java.util.HashMap;
```

```

import java.util.List;
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[] split = in.nextLine().split("@");
        if(split.length < 2){
            System.out.println(split[0]);
            return;
        }
        List<String> list = new ArrayList<>();
        HashMap<String, Integer> map = new HashMap<>();
        String[] allArr = split[0].split(",");
        for (String s : allArr) {
            String[] split1 = s.split(":");
            list.add(split1[0]);
            map.put(split1[0], Integer.parseInt(split1[1]));
        }
        String[] useArr = split[1].split(",");
        for (String s : useArr) {
            String[] split1 = s.split(":");
            int count = map.get(split1[0]);
            if(count > Integer.parseInt(split1[1])){
                map.put(split1[0], count - Integer.parseInt(split1[1]));
            }
            else{
                map.remove(split1[0]);
                list.remove(split1[0]);
            }
        }
        StringBuilder builder = new StringBuilder();
        for (String s : list) {
            int val = map.get(s);
            builder.append(s).append(":").append(val).append(",");
        }
        builder.deleteCharAt(builder.length() - 1);
        System.out.println(builder.toString());
    }
}

```