

VLAN 资源池

题目描述: VLAN是一种对局域网设备进行逻辑划分的技术,为了标识不同的VLAN,引入VLAN ID(1-4094之间的整数)的概念。定义一个VLAN ID的资源池(下称VLAN资源池),资源池中连续的VLAN用开始VLAN-结束VLAN表示,不连续的用单个整数表示,所有的VLAN用英文逗号连接起来。现在有一个VLAN资源池,业务需要从资源池中申请一个VLAN,需要你输出从VLAN资源池中移除申请的VLAN后的资源池。

输入描述: 第一行为字符串格式的VLAN资源池,第二行为业务要申请的VLAN, VLAN的取值范围为[1,4094]之间的整数。

输出描述: 从输入VLAN资源池中移除申请的VLAN后字符串格式的VLAN资源池,输出要求满足题目描述中的格式,并且按照VLAN从小到大升序输出。如果申请的VLAN不在原VLAN资源池内,输出原VLAN资源池升序排序后的字符串即可。

补充说明: 输入VLAN资源池中VLAN的数量取值范围为[2-4094]间的整数,资源池中VLAN不重复且合法([1,4094]之间的整数),输入是乱序的。

示例1

输入: 1-5

2

输出: 1,3-5

说明: 原VLAN资源池中有VLAN 1、2、3、4、5,从资源池中移除2后,剩下VLAN 1、3、4、5,按照题目描述格式并升序后的结果为1,3-5。

示例2

输入: 20-21,15,18,30,5-10

15

输出: 5-10,18,20-21,30

说明: 原VLAN资源池中有VLAN 5、6、7、8、9、10、15、18、20、21、30,从资源池中移除15后,资源池中剩下的VLAN为 5、6、7、8、9、10、18、20、21、30,按照题目描述格式并升序后的结果为5-10,18,20-21,30。

示例3

输入: 5,1-3

10

输出: 1-3,5

说明: 原VLAN资源池中有VLAN 1、2、3、5,申请的VLAN 10不在原资源池中,将原资源池按照题目描述格式并按升序排序后输出的结果为1-3,5。

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
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 sc = new Scanner(System.in);
        String[] VLANPOOL = sc.nextLine().split(",");
        int VLANNeed = sc.nextInt();
        ArrayList<Integer> list = new ArrayList<>();
        for (String vlan : VLANPOOL) {
            String[] tmp = vlan.split("-");
            if (tmp.length > 1) {
                for (int i = Integer.parseInt(tmp[0]); i <= Integer.parseInt(tmp[1]); i++) {
                    list.add(i);
                }
            } else {
                list.add(Integer.parseInt(tmp[0]));
            }
        }

        for (int i = 0; i < list.size(); i++) {
            if (list.get(i) == VLANNeed) {
                list.remove(i);
            }
        }
    }
}
```

```

    }
}
Collections.sort(list);
StringBuilder sb = new StringBuilder();

int i = 0;
while (i < list.size()) {
    int first = list.get(i);
    int j = 1;
    while (j <= list.size() - 1 - i) {
        if (list.get(i) + j == list.get(i + j)) {
            j++;
        } else {
            break;
        }
    }
    if (j == 1) {
        sb.append(first);
        sb.append(",");
        i++;
    } else {
        sb.append(String.valueOf(first) + "-" + String.valueOf(first + j - 1) + ",");
        i = i + j;
    }
}
System.out.println(sb.toString().substring(0, sb.length() - 1));
}
}

```