

题目描述：

现有两个整数数组，需要你找出两个数组中同时出现的整数，并按照如下要求输出：

- 1、有同时出现的整数时，先按照同时出现次数（整数在两个数组中都出现并且出现次数较少的那个）进行归类，然后按照出现次数从小到大依次按行输出。
- 2、没有同时出现的整数时，输出 **NULL**。

输入描述：

第一行为第一个整数数组，第二行为第二个整数数组，每行数据中整数与整数之间以英文逗号分隔，整数的取值范围为 $[-200,200]$ ，数组长度的范围为 $[1,10000]$ 之间的整数。

输出描述：

按照出现次数从小到大依次按行输出，每行输出的格式为:出现次数:该出现次数下的整数升序排序的结果。

格式中的":"为英文冒号，整数间以英文逗号分隔。

补充说明：

示例 1

输入：

5,3,6,-8,0,11

2,8,8,8,-1,15

输出：

NULL

说明：

两个整数数组没有同时出现的整数，输出 **NULL**。

示例 2

输入：

5,8,11,3,6,8,8,-1,11,2,11,11

11,2,11,8,6,8,8,-1,8,15,3,-9,11

输出：

1:-1,2,3,6

3:8,11

说明：

两个整数数组中同时出现的整数为-1、2、3、6、8、11，其中同时出现次数为1的整数为-1,2,3,6(升序排序)，同时出现次数为3的整数为8,11(升序排序)，先升序输出出现次数为1的整数，再升序输出出现次数为3的整数。

// 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);
```

```
        while (in.hasNext()) { // 注意，如果输入是多个测试用例，请通过 while 循环处理多个测试用例
```

```
            String input1 = in.nextLine();
```

```
            String input2 = in.nextLine();
```

```
            String[] strArr1 = input1.split(",");
```

```
String[] strArr2 = input2.split(",");
Map<Integer, Integer> map1 = new HashMap<>();
Map<Integer, Integer> map2 = new HashMap<>();
for (int i = 0; i < strArr1.length; i++) {
    map1.put(Integer.parseInt(strArr1[i]),
        map1.getDefault(Integer.parseInt(strArr1[i]), 0) + 1);
}
for (int i = 0; i < strArr2.length; i++) {
    map2.put(Integer.parseInt(strArr2[i]),
        map2.getDefault(Integer.parseInt(strArr2[i]), 0) + 1);
}
Map<Integer, Set<Integer>> mergeMap = new TreeMap<>();
for (Map.Entry<Integer, Integer> entry : map1.entrySet()) {
    int key = entry.getKey();
    if (map2.containsKey(key)) {
        int cnt = Math.min(map1.get(key), map2.get(key));
        if (mergeMap.containsKey(cnt)) {
            mergeMap.get(cnt).add(key);
        } else {
            Set<Integer> set = new TreeSet<>();
            set.add(key);
            mergeMap.put(cnt, set);
        }
    }
}
if (mergeMap.isEmpty()) {
    System.out.println("NULL");
    return;
}
for (Map.Entry<Integer, Set<Integer>> entry : mergeMap.entrySet()) {
    StringBuilder sb = new StringBuilder();
    sb.append(entry.getKey() + ":");
    for (int num : entry.getValue()) {
        sb.append(num + ",");
    }
    System.out.println(sb.toString().substring(0, sb.length() - 1));
}
}
```