

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。

```
import functools
import sys
from collections import Counter,defaultdict
import copy
from itertools import permutations
import re
import math
from queue import Queue

arr1=[int(x) for x in input().split(",")]
arr2=[int(x) for x in input().split(",")]

num1_map={}
for num in arr1:
    num1_map[num]=num1_map.get(num,0)+1
num2_map={}
for num in arr2:
    num2_map[num]=num2_map.get(num,0)+1

flag=True

same_num_map={}

for num in num1_map.keys():
    if num in num2_map.keys():
        flag=False
        count = min(num1_map[num],num2_map[num])
        if same_num_map.get(count) is None:
            same_num_map[count]=[]
        same_num_map[count].append(num)
```

```
if flag:
    print("NULL")
else:
    result=list(same_num_map.keys())
    result.sort()
    for count in result:
        same_num_map[count].sort()
        print(f"{count}:{','.join(map(str,same_num_map[count]))}")
```