

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。

```
from collections import defaultdict
```

```
list1 = input().split(',')
```

```
list2 = input().split(',')
```

```
list1.sort()
```

```
list2.sort()
```

```
def compare_list(list1, list2):
```

```
    index1, index2 = 0, 0
```

```

l1_len = len(list1)
l2_len = len(list2)
res = defaultdict(int)
result = {}
while index1 < l1_len and index2 < l2_len:
    if list1[index1] == list2[index2]:
        res[list1[index1]] += 1
        index1 += 1
        index2 += 1
    elif list1[index1] > list2[index2]:
        index2 += 1
    elif list1[index1] < list2[index2]:
        index1 += 1
for key, value in res.items():
    if value in result:
        result[value].append(key)
    else:
        result[value] = [key]
result = sorted(result.items(), key=lambda x:int(x[0]))
flag = False
for key, value in result:
    value.sort(key=lambda x: int(x))
    values = ','.join(value)
    print(f"{key}:{values}")
if not result:
    print("NULL")

compare_list(list1, list2)

```