

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。

```
const rl = require("readline").createInterface({ input: process.stdin });
var iter = rl[Symbol.asyncIterator]();
const readline = async () => (await iter.next()).value;
```

```
void async function () {
    // Write your code here
    while(line = await readline()){
        line = line.split(',').map(Number)
        let line2 = (await readline()).split(',').map(Number)
        let map1 = new Map()
        let map2 = new Map()
        line.forEach((item)=>{
            map1.set(item,(map1.get(item)+1) || 1)
        })
        line2.forEach((item)=>{
            map2.set(item,(map2.get(item)+1) || 1)
        })
        let result = new Map()
        let set1 = new Set(line)
        let set2 = new Set(line2)
        set1.forEach(item=>{
            if(set2.has(item)){
                let num1 = map1.get(item)
                let num2 = map2.get(item)
                let num = Math.min(num1,num2)
                if(result.has(num)){
                    let tmp = result.get(num)
                    tmp.push(item)
                }
            }
        })
    }
}
```

```
        result.set(num,tmp)
      }else{
        result.set(num,[item])
      }
    }
  })
  let resultArr = [...result].sort((a,b)=>a[0]-b[0])
  if(resultArr.length==0) {
    console.log("NULL")
    return
  }
  resultArr.forEach(item=>{
    let key = item[0]
    let val = item[1].map(Number).sort((a,b)=>a-b).join(',')
    console.log(`${key}:${val}`)
  })
}
}()
```