

内存资源分配

有一个简易内存池，内存按照大小粒度分类，每个粒度有若干个可用内存资源，用户会进行一系列内存申请，需要按需分配内存池中的资源，返回申请结果成功失败列表。分配规则如下：

- 1、分配的内存要大于等于内存申请量，存在满足需求的内存就必须分配，优先分配粒度小的，但内存不能拆分使用。
- 2、需要按申请顺序分配，先申请的先分配。
- 3、有可用内存分配则申请结果为 `true`，没有可用内存分配则返回 `false`。

注：不考虑内存释放。

输入描述：

输入为两行字符串：

第一行为内存池资源列表，包含内存粒度数据信息，粒度数据间用逗号分割，一个粒度信息内部用冒号分割，冒号前为内存粒度大小，冒号后为数量。资源列表不大于 1024，每个粒度的数量不大于 4096

第二行为申请列表，申请的内存大小间用逗号分隔。申请列表不大于 100000

如：

64:2,128:1,32:4,1:128

50,36,64,128,127

输出描述：

输出为内存池分配结果。

如：

`true,true,true,false,false`

补充说明：

示例 1

输入：

64:2,128:1,32:4,1:128

50,36,64,128,127

输出：

`true,true,true,false,false`

说明：

内存池资源包含：64K 共 2 个、128K 共 1 个、32K 共 4 个、1K 共 128 个的内存资源；
针对 50,36,64,128,127 的内存申请序列，分配的内存依次是：64,64,128,NULL,NULL,第三次申请内存时已经将 128 分配出去，

因此输出结果是：`true,true,true,false,false`

```
nei = list(input().split(','))
```

```
length = len(nei)
```

```

ori = []
for i in nei:
    middle = list(map(int,list(i.split(':'))))
    ori.append([middle[0],middle[1]])

ori.sort(key=lambda x: x[0])
#print(ori)
ask = list(input().split(','))

length_ask = len(ask)
#for i in range(length_ask):
#    if ask[i] == '':
#        ask.remove(ask[i])
#        break
ask = list(map(int,ask))

ans = ['false' for _ in range(length_ask)]

for i in range(length_ask):
    for index,value in enumerate(ori):
        if ask[i] <= value[0] and value[1]>0:
            ans[i] = 'true'
            value[1] -= 1
            break

print(', '.join(ans))

```