

实验任务 3-4 简单数据管理

题目描述:

```
political_divisions = [  
    {  
        'name': 'Alabama', 'abbreviation': 'AL', 'counties': [  
            'Autauga', 'Baldwin', 'Barbour', 'Bibb'  
        ]  
    },  
    {  
        'name': 'Alaska', 'abbreviation': 'AK', 'counties': [  
            'Aleutians East', 'Anchorage', 'Bristol Bay'  
        ]  
    }  
]
```

上面的程序段定义了一个列表，包含美国阿拉巴马州、阿拉斯加州的部分郡级行政单位信息。我们提供了 `political_divisions.py` 程序，定义了 `political_divisions` 列表，以类似的格式包含了美国七个州的部分行政区信息。

`political_divisions` 列表包含若干项目，每个项目为一个字典项，表示一个州份。每个州份分别包含三个字典项：

`name` 项对应的值为一个字符串，表示州的名称；

`abbreviation` 项对应的值为一个字符串，表示州的缩写；

`counties` 项对应的值为一个列表，表示州所含的部分郡级行政单位名称。

现在要求你编写一个程序，可以对 `political_divisions.py` 程序中的 `political_division` 列表之内的信息进行两种查询操作：

操作 1：给定一个州的名称或缩写，并给出一个字母，输出字典内包含的所有首字母为该字母的郡（`counties`）的信息；

如：1 AK B 表示搜索所有阿拉斯加州以 B 开头的郡级行政单位名称；

操作 2：给出一个郡名称，输出字典内所有包含同名郡的州的缩写；

如：2 Washington 表示搜索所有包含名为 Washington 的郡级行政单位的州份缩写。

请你对用户的 n 个查询操作，分别完成查询操作。

输入格式：

输入数据共 $n+1$ 行：

第一行为一个整数 n ，表示所有查询的总次数；

接下来的 n 行，每行包含一个查询：

对于第一种查询，数字 1 后有一个空格，其后为表示州名称或缩写的字符串，由大小写字母组成，其后再跟一个空格，为查询的首字母，大小写均有可能；

对于第二种查询，数字 2 后有一个空格，其后为表示郡级行政单位名称的字符串，由大小写字母和空格组成。

输出格式：

输出包含 n 行，每行包含数个以英文逗号+空格隔开的字符串，包含每次查询的所有结果，以字典序排列；最后一个结果后没有逗号。

实验指导：

1. 设计思路

对于两种查询，我们的思路都是类似的，即遍历所有可能的选项，筛选出其中符合条件的，然后使用相应格式输出。对于第一种查询，我们可以遍历州份，找到对应的州，再从其郡的名单里筛选出符合要求首字母的郡名；对于第二种查询，我们需要遍历所有郡名，找到对应的州的缩写。

本题内容较为综合，不论是输入处理，还是最后排序输出，都包含了一些值得关注的细节。依写法不同，可能需要多重循环嵌套。

2. for 循环与数据结构配合使用

for 循环不仅可以和 range 结构搭配，也可以用于遍历数据结构，具体用法为 `for element in structure`。当 structure 是列表或元组时，element 依次被赋值为 structure 内部的各项内容的值；当 structure 是字典时，element 依次被赋值为字典的每个 key 值。

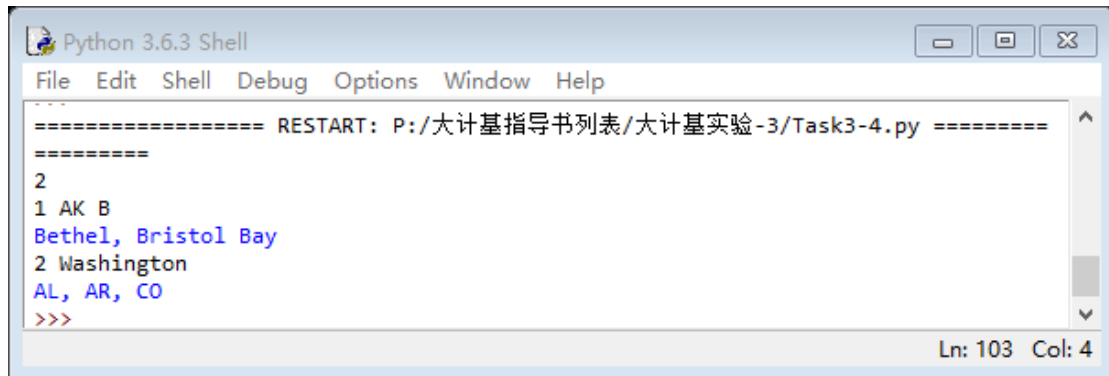
试研究以下程序：

```
odds = [1,3,5,7]
for num in odds:
    print(num)          #输出 1, 3, 5, 7
dishes = {'eggs': 2, 'sausage': 1}
for obj in dishes:
    print(obj)          #输出 eggs, sausage
for obj in dishes:
    print(dishes[obj])  #输出 2, 1
```

3. 内置的排序方法

对一个包含相同数据类型的列表，Python 提供了内置的排序方法。如对于列表 `lis`，`lis.sort()` 会直接调整 `lis` 的内部元素顺序，使其顺序从小到大。使用 `sorted(lis)` 则可以直接获得一份 `lis` 的有序拷贝。具体结果可以自行试验完成。

参考运行结果：



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
===== RESTART: P:/大计基指导书列表/大计基实验-3/Task3-4.py =====
=====
2
1 AK B
Bethel, Bristol Bay
2 Washington
AL, AR, CO
>>>
```

Ln: 103 Col: 4

实验任务 3-5 质数的和（选做）

题目描述：

给定 2 个整数 a, b 求出它们之间（不含 a, b ）所有质数的和。

输入格式：

输入数据包含一行， 包含两个数字， 分别为 $a, b(a \leq b \leq 500000)$

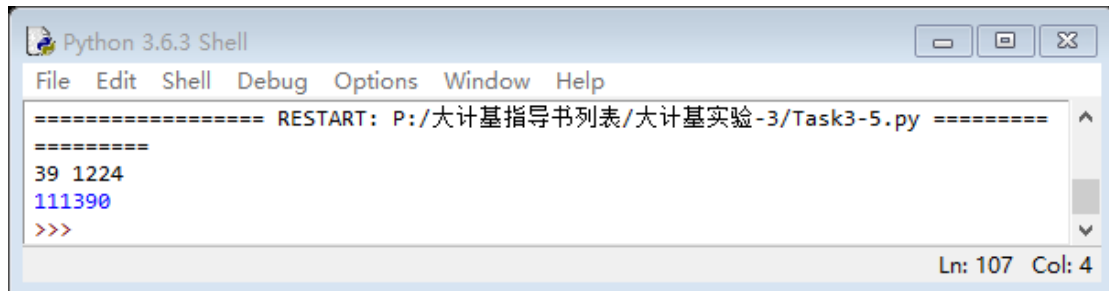
输出格式：

输出数据包含一行， 为一个整数， 表示 a, b 之间（含 a, b ）所有质数的和。

实验指导：

使用搜索引擎（如百度），查找在值很大的情况下，较快批量寻找质数的方法，理解该方法，并以此为基础编程。

参考运行结果：



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
===== RESTART: P:/大计基指导书列表/大计基实验-3/Task3-5.py =====
=====
39 1224
111390
>>>
Ln: 107 Col: 4
```

提示：本题测试数据很大，采用效率较低的方法很可能出现 TLE 错误。可以采用题目范围内较大的数据进行测试，如果运行时间较长，尽量考虑其他方法。效率较低但答案正确的方法也可以拿到 40%~80% 分数。