

一、编程题

ACM：精准核酸检测

题目描述：为了达到新冠疫情精准防控的需要，为了避免全员核酸检测带来的浪费，需要精准圈定可能被感染的人群。

现在根据传染病流调以及大数据分析，得到了每个人之间在时间、空间上是否存在轨迹的交叉。

现在给定一组确诊人员编号（ $X_1, X_2, X_3, \dots, X_n$ ），在所有人当中，找出哪些人需要进行核酸检测，输出需要进行核酸检测的人数。（注意：确诊病例自身不需要再做核酸检测）

需要进行核酸检测的人，是病毒传播链条上的所有人员，即有可能通过确诊病例所能传播到的所有人。

例如：A是确诊病例，A和B有接触、B和C有接触、C和D有接触、D和E有接触，那么B\C\D\E都是需要进行核酸检测的人。

输入描述：第一行为总人数N

第二行为确诊病例人员编号（确诊病例人员数量 $<N$ ），用逗号分割

第三行开始，为一个 $N*N$ 的矩阵，表示每个人员之间是否有接触，0表示没有接触，1表示有接触。

输出描述：整数：需要做核酸检测的人数

补充说明：人员编号从0开始

$0 < N < 100$

示例1

输入：5

```
1,2
1,1,0,1,0
1,1,0,0,0
0,0,1,0,1
1,0,0,1,0
0,0,1,0,1
```

输出：3

说明：编号为1、2号的人员，为确诊病例。

1号与0号有接触，0号与3号有接触。

2号与4号有接触。

所以，需要做核酸检测的人是0号、3号、4号，总计3人需要进行核酸检测。

代码：

```
import sys
```

```
while True:
```

```
    try:
```

```
        n = int(input())
```

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

```
        matrix = []
```

```
        for i in range(n):
```

```
            matrix.append(input().split(','))
```

```
        queue = []
```

```
        visited_person = set()
```

```
        count = 0
```

```
        for person in persons:
```

```
            index = int(person)
```

```
            queue.append(index)
```

```
        while queue:
```

```
            i = queue.pop(0)
```

```
visited_person.add(i)
for j in range(n):
    if matrix[i][j] == '1' or matrix[j][i] == '1':
        if j not in visited_person:
            queue.append(j)
            visited_person.add(j)

result = len(visited_person) - len(persons)
print(result)

except:
    break
```