

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 $B \setminus C \setminus D \setminus E$ 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 $N \times N$ 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。

```
#include <iostream>
#include<bits/stdc++.h>
using namespace std;

int main() {
    int n;
    cin>>n;
    vector<vector<int>>>A(n,vector<int>(n));
    string s;
    cin>>s;
    int cur=0;
    vector<int>nums;
    while(1){
        int x=s.find(',');
        if(x!=-1){
            nums.push_back(stoi(s.substr(0,x)));
            s=s.substr(x+1,s.size());
        }else{
            nums.push_back(stoi(s));
            break;
        }
    }
    vector<int>vis(n,false);
    queue<int>q;

    int ret=0;
    for(auto x:nums)q.push(x),vis[x]=true,--ret;

    vector<vector<int>>>connect(n);
    for(int i=0;i<n;++i){
        string s;
        cin>>s;
        for(int j=0;j<n;++j){
```

```

        if(s[2*j]=='1'){
            connect[i].push_back(j);
            connect[j].push_back(i);
        }
    }
}

while(!q.empty()){
    auto x=q.front();
    q.pop();
    for(auto y:connect[x]){
        if(!vis[y]){
            q.push(y);
            vis[y]=true;
        }
    }
}

// int ret=0;
for(auto x:vis){
    if(x)++ret;
}

printf("%d\n",ret);
return 0;

}
// 64 位输出请用 printf("%lld")

```