

题目描述:

在一个地图中(地图由 $n*n$ 个区域组成)，有部分区域被感染病菌。感染区域每天都会把周围（上下左右）的 4 个区域感染。

请根据给定的地图计算，多少天以后，全部区域都会被感染。

如果初始地图上所有区域全部都被感染，或者没有被感染区域，返回 -1

输入描述:

一行 $N*N$ 个数字（只包含 0,1，不会有其他数字）表示一个地图，数字间用,分割，0 表示未感染区域，1 表示已经感染区域

每 N 个数字表示地图中一行，输入数据共表示 N 行 N 列的区域地图。

例如输入 1,0,1,0,0,0,1,0,1，表示地图

1,0,1

0,0,0

1,0,1

输出描述:

一个整数，表示经过多少天以后，全部区域都被感染

补充说明:

$1 \leq N < 200$

示例 1

输入:

1, 0, 1, 0, 0, 0, 1, 0, 1

输出:

2

说明：

1 天以后，地图中仅剩余中心点未被感染；2 天以后，全部被感染。

示例 2

输入：

0,0,0,0

输出：

-1

说明：

无感染区域

示例 3

输入：

1,1,1,1,1,1,1,1,1,1

输出：

-1

说明：

全部都感染

```
#include <iostream>
```

```
#include<vector>
```

```
#include<string>
```

```
#include<sstream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
int main() {
```

```
    vector<int> nums;
```

```
    string str;
```

```
    getline(cin,str);
```

```
    stringstream ss(str);
```

```

string token;
int znum = 0;
int onum = 0;
while(getline(ss,token,', ')){
    int number = stoi(token);
    nums.push_back(number);
    if(number==1){
        ++onum;
    }
    if(number==0){
        ++znum;
    }
}

int n = sqrt(nums.size());
int cnt = 0;
vector<int> tmp;
while(onum!=nums.size() && znum != nums.size()){
    znum = 0;
    onum = 0;
    for(int i = 0;i<nums.size();++i){
        if(1==nums[i]){
            ++onum;
            if(i%n!=0){
                tmp.push_back(i-1);
            }
            if((i+1)%n!=0){
                tmp.push_back(i+1);
            }
            if(i-n>=0){
                tmp.push_back(i-n);
            }
            if((i+n)/n<n){
                tmp.push_back(i+n);
            }
        } else {
            ++znum;
        }
    }
    for(auto &c:tmp){
        nums[c] = 1;
    }
    tmp.clear();
    ++cnt;
}

```

```
}  
if(cnt == 0){  
    cout << -1 << endl;  
} else {  
    cout << cnt-1 << endl;  
}  
}  
// 64 位输出请用 printf("%lld")
```