1 寻找下标

问题分析：这是一道关于数组的问题，我们所要做的就是将数组下标与数值相同的数求出来，若有多个，就返回第一个。没有就打印 N。

解决方案：使用循环，将每个数都遍历一次，看是否满足条件 a[i]==i，满足就输出这个数。并停止循环，不满足并且循环结束就打印 N。

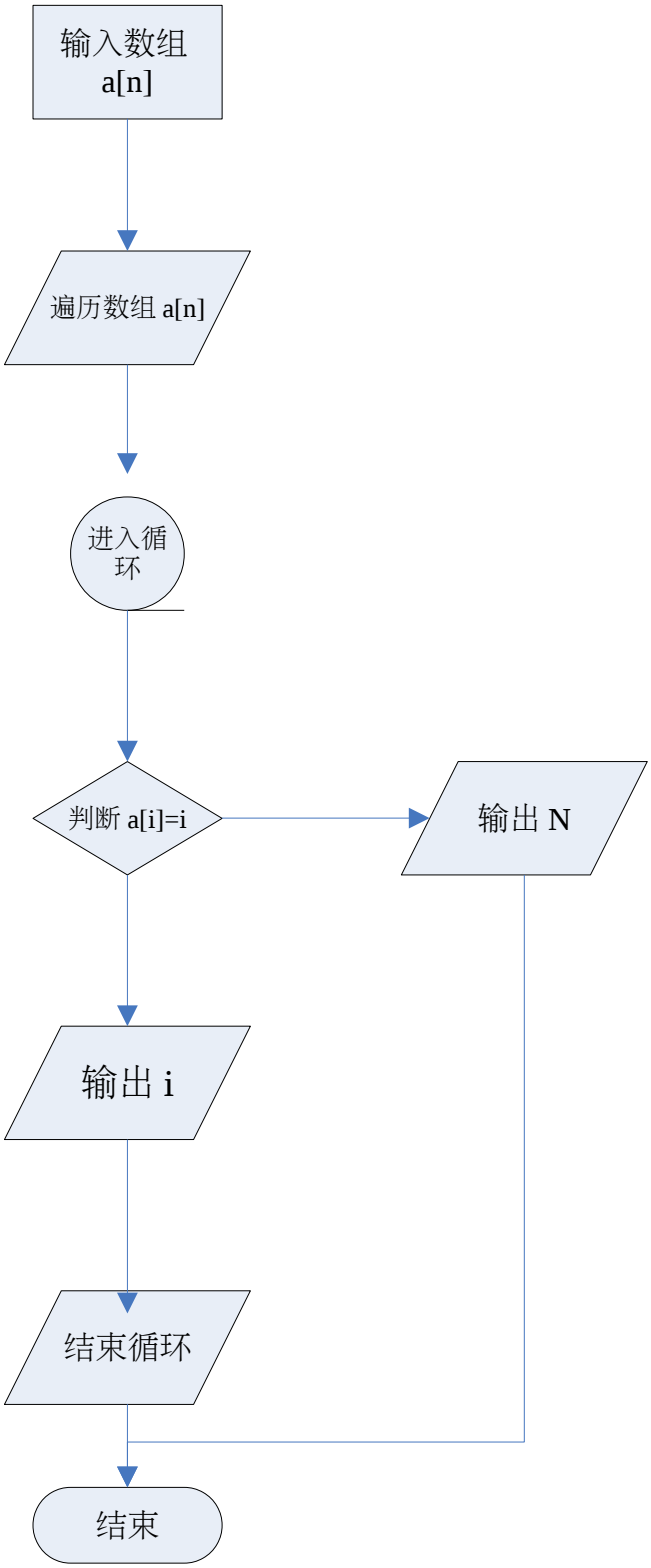
编程实现：

```
/*************************************************************************
 > File Name: 寻找下标.cpp
 > Author: zhengziqiang
 > Mail: 1174986943@qq.com
 > Created Time: 2016 年 04 月 05 日 星期二 23 时 42 分 54 秒
 ************************************************************************/

#include<iostream>
using namespace std;
int main()
{
   int n,a[n],m=0;
   cin>>n;
   for(int i=0;i<n;i++)
   {
      cin>>a[i];
   }
   for(int j=0;j<n;j++)
   {
      if(a[j]==j)
      {
         cout<<j<<endl;
         m+=1;
         break;
      }
   }
   if(m==0)
      cout<<"N"<<endl;
   return 0;

}
~
```

流程图：



3 发票统计

问题分析：现有 3 个人的发票，发票种类有 A，B，C 三种，现在要统计这 3 个人的发票总金额和 A，B，C 这 3 种发票的金额和，并依次打印出来。

解决方案：我们首先要做的就是定义出 6 个浮点数来表示他们的金额和，float1 float2 float3 floata floatb floatc。定义好了之后我们就要进入第一个循环在这个循环里我们要输入每个人的发票种类和发票金额。然后并在这一层循环里将每个人的发票金额总和求出来。求出来之后我们进入第二层循环，由于每个人的发票数目不同，所以我们在这一层循环里要定义一个数组来存放发票金额。最后在第一层循环里求出 floata floatb floatc.

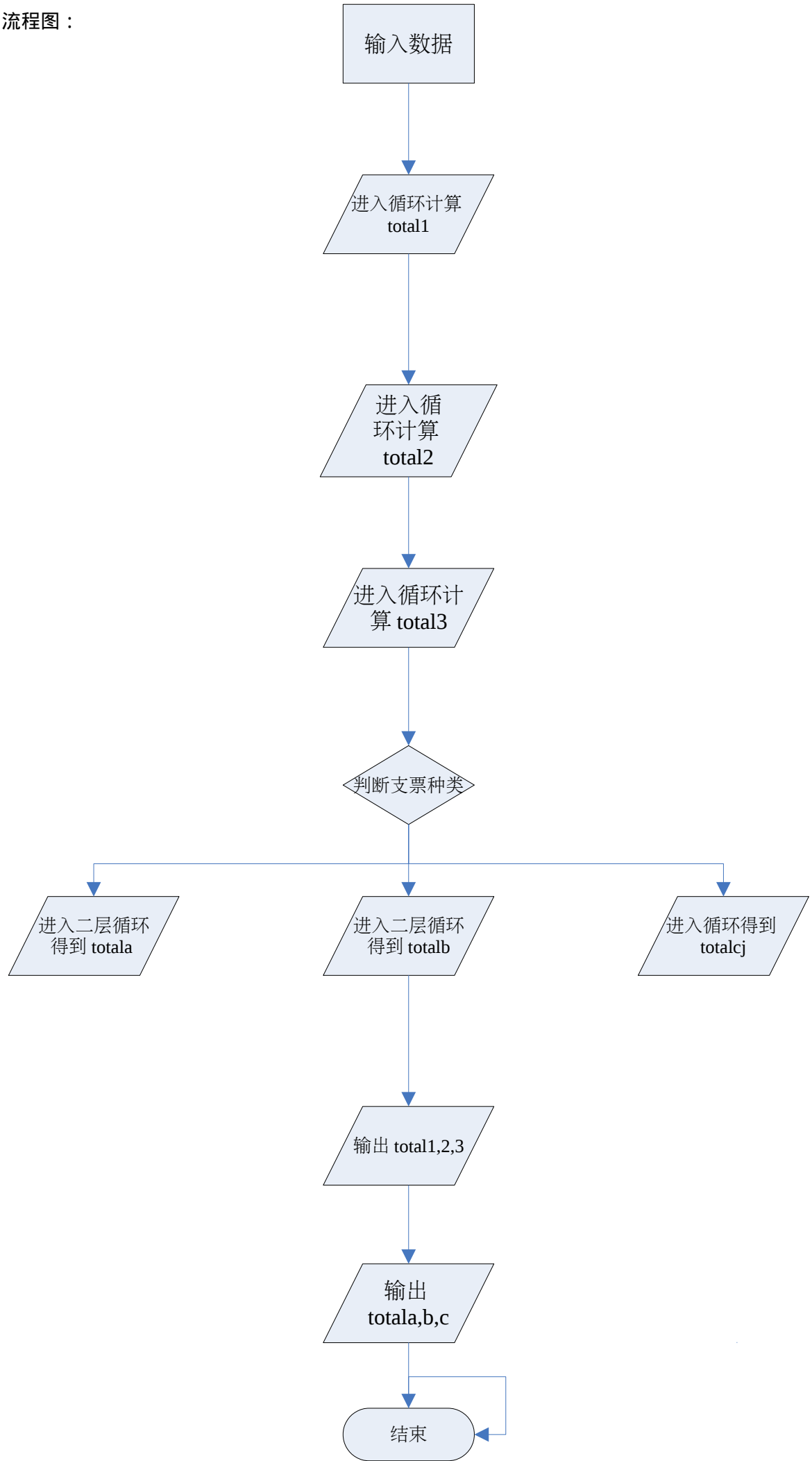
编程实现：

```
#include<iostream>
#include<iomanip>
using namespace std;
int main(){
   float total[3];
   total[0]=0.0;total[1]=0.0;total[2]=0.0;
   float totala=0.0;
   float tatalb=0.0;
   float tatalc=0.0;
   int id[3],b[3];
   for(int j=0;j<3;j++){
   cin>>id[j]>>b[j];
   char a[b[j]];
   float n[b[j]];
   for(int i=0;i<b[j];i++){
      cin>>a[i]>>n[i];
   }
   for(int k=0;k<b[j];k++){
      (total[j])+=n[k];
      if(a[k]=='A')(totala)+=n[k];
      if(a[k]=='B')(totalb)+=n[k];
      if(a[k]=='C')(totalc)+=n[k];
   }
   }

   for(int k=0;k<3;k++){
   if(id[k]==1)cout<<"1 "<<fixed<<setprecision(2)<<total[0]<<endl;
   if(id[k]==2)cout<<"2 "<<fixed<<setprecision(2)<<total[1]<<endl;
   if(id[k]==3)cout<<"3 "<<fixed<<setprecision(2)<<total[2]<<endl;
   }
   cout<<"A "<<fixed<<setprecision(2)<<(totala)<<endl;
   cout<<"B "<<fixed<<setprecision(2)<<(totalb)<<endl;
   cout<<"C "<<fixed<<setprecision(2)<<(totalc)<<endl;
   return 0;
}
```

总结体会：这道题其实也可以通过编写函数来实现，在函数里我们可以返回多值就可以解决不同发票的问题。

流程图：


输入数据
进入循环计算 total1
进入循环计算 total2
进入循环计算 total3
判断支票种类
进入二层循环得到 totala
进入二层循环得到 totalb
进入循环得到 totalcj
输出 total1,2,3
输出 totala,b,c
结束

5 流感传染

问题分析：在一个 N 维矩阵里，每个矩阵的每一个点有三中状态，住着生病的人，没人，健康的人，题目告诉我们生病的人每天会向周围传染病菌，会让他的邻居得病，没人的房间则不会。

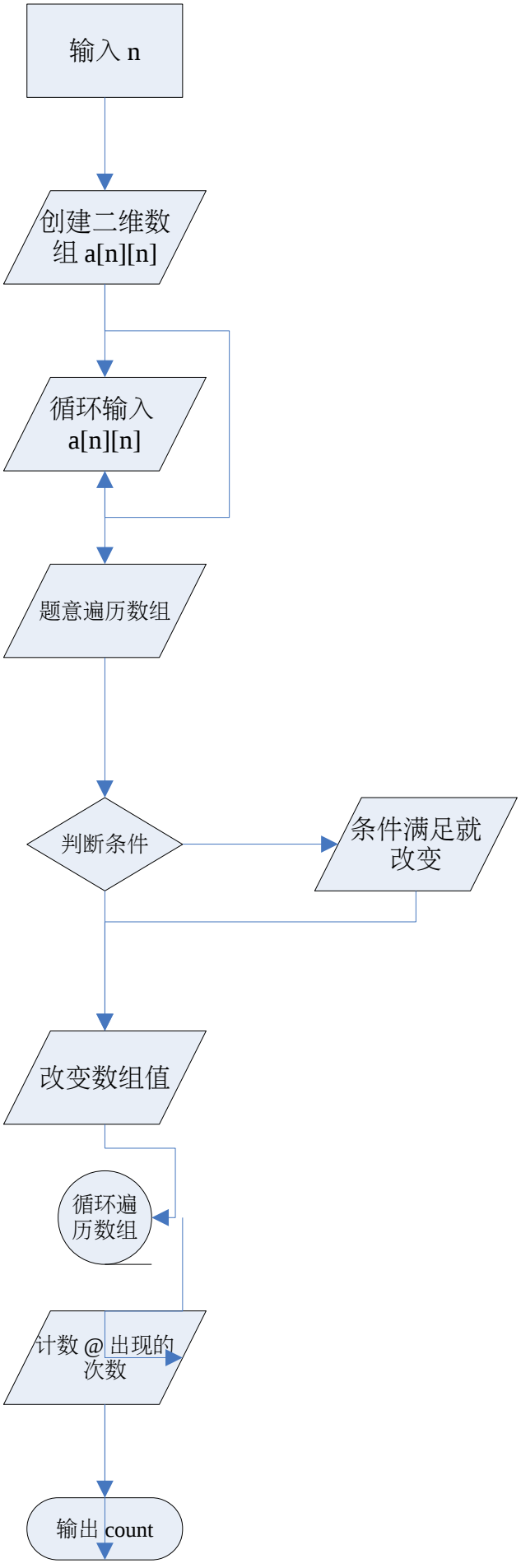
解决方案：首先我们创建一个二维 char 型数组来代表这个 n 维矩阵，每一天之后得病的人就会让周围的人得病（房间空着的人除外）。

编程实现：

```
#include<iostream>
using namespace std;
int *fun(char *a[n][n],int m){
   for(int i=1;i<m;i++){
      for(int j=0;j<n;j++){
         for(int k=0;k<n;k++){
            if(*a[j][k]=='@'){
               if(j>0&&j<n-1&&k>0&&k<n-1){
                  if(*a[j-1][k-1]!='#')*a[j-1][k-1]='@';
                  if(*a[j+1][k-1]!='#')*a[j+1][k-1]='@';
                  if(*a[j-1][k+1]!='#')*a[j-1][k+1]='@';
                  if(*a[j+1][k+1]!='#')*a[j+1][k+1]='@';
               }
            }
         }
      }
   }
}
int main(){
   int n;
   cin>>n;
   char a[n][n];
   for(int i=0;i<n;i++){
      for(int j=0;j<n;j++)cin>>a[i][j];
   }
   int m;
   cin>>m;
   int **p;
   p=a;
   fun(p,m);
   int count=0;
   for(int k=0;k<n;k++){
      for(int l=0;l<n;l++){
         if(*(p[k]+l)=='@')count+=1;
      }
   }
   count<<count<<endl;
}
```

总结体会：这道题比较复杂，我考虑了很久才想出来，我们要分两种情况讨论，一是矩阵边缘的情况，二是我们要考虑矩阵房子空着的情况，所以这道题要分多种情况讨论显得特别特别复杂。

流程图：


输入 n
创建二维数组 a[n][n]
循环输入 a[n][n]
题意遍历数组
判断条件
条件满足就改变
改变数组值
循环遍历数组
计数 @ 出现的次数
输出 count