

C 盗墓笔记之鬼玺

——by 董舒印 & Arthur

题目描述：

看密匝匝蚁排兵,乱纷纷蜂酿蜜,急攘攘蝇争血。天真刚刚躲过怪物的洗礼，便赶上了这千年不遇的鬼俑阵列。

我们的故事在这之前。

少顷前，Thor 拿着鬼玺也遭遇了鬼俑的阵列，虽然拿着鬼玺，然而要号令鬼俑，还需要吸日月之灵气，汲日月之精华，按照要求将鬼俑排列好。

小哥告诉 Thor，阵列为正方形，其中鬼俑只有两种，普通步兵俑，执旗步兵俑。对于一个 $n*n$ 的阵列，要求每一行，每一列以及每一斜行都 **只有** 一个执旗步兵俑。

(n 个执旗步兵俑必须要放入！！)

输入：

多组测试数据。

对于每组测试数据，第一行为一个整数 n ($1 \leq n \leq 20$)，表示阵列的边长。

接下来 n 行为一个 $n*n$ 的矩阵，表示 Thor 排列出的阵列。其中 0 为普通步兵俑，1 为执旗步兵俑。

输出：

对于每组数据，如果列阵成功，则输出 Move Forward!

反之输出 Bad End!

输入样例：

```
8
10000000
00001000
00000001
00000100
00100000
00000010
01000000
00010000
```

输出样例：

Move Forward!

题目分析

这个题上机的时候题目比较坑……

其实就是分析所给的矩阵是不是 N 皇后……

思路很简单，先判断 1 的个数是不是 n ，然后横行，竖行，斜行 1 的个数不超过 1 就可以了。

但是写起来比较麻烦- -#，或许因为我写的比较渣==

Arthur：楼上说的对！

Code for Reference:

```
#include<cstdio>
#include<iostream>
#include<cstring>
using namespace std;
char a[50][50];
int b[50][50];
int main()
{
    int n, c, sum;
    bool flag;//一个标记变量，如果任何一条不符合题意，后面的运算则不进行，节约时间。
    while (scanf("%d", &n) != EOF)
    {
        flag=1, sum=0;
        memset(a, 0, sizeof(a));
        memset(b, 0, sizeof(b));
        for (int i=1; i<=n; i++)
            scanf("%s", a[i]); //注意 C 风格输入字符串的时候不加&，因为
// a[i]就是二维数组 a[i][50]的地址……这么说能懂么 = =上课宋老师讲过的
        for (int i=1; i<=n; i++)
            for (int j=1; j<=n; j++)
                b[i][j]=a[i][j-1]-'0'; //这里注意是 j-1，因为默认 a[i][0]
// 是会被输入的……我个人是转换为数字算的——用字符其实也行。
        for (int i=1; i<=n; i++)
            for (int j=1; j<=n; j++)
                sum+=b[i][j];
        if (sum!=n)
            flag=0; //检查是否有 n 的 1 被放入。
        for (int i=1; i<=n&&flag; i++)
        {
            sum=0;
            for (int j=1; j<=n; j++)
                sum+=b[i][j];
            if (sum!=1)
                flag=0;
        } //检查竖行。
        for (int i=1; i<=n&&flag; i++)
        {
            sum=0;
            for (int j=1; j<=n; j++)
                sum+=b[j][i];
            if (sum!=1)
```

```

        flag=0;
    }//检查横行，这个可以复制粘贴- 然后 i 和 j 换位置就好了。
    for(int i1=1, j1=1; i1<=n&& j1<=n&& flag; i1++)
    {
        sum=0;
        for(int i=i1, j=j1; (i>=1&&i<=n)&&(j>=1&&j<=n); i--, j++)
            sum+=b[i][j];
        for(int i=i1+1, j=j1-1; (i>=1&&i<=n)&&(j>=1&&j<=n); i++, j--)
            sum+=b[i][j];
        if(sum>1)
            flag=0;
    }//其中有些东西很多余，但是不会导致出错，而且方便后面复制粘
贴.....
    for(int i1=n, j1=1; i1<=n&& j1<=n&& flag; j1++)
    {
        sum=0;
        for(int i=i1, j=j1; (i>=1&&i<=n)&&(j>=1&&j<=n); i--, j++)
            sum+=b[i][j];
        for(int i=i1+1, j=j1-1; (i>=1&&i<=n)&&(j>=1&&j<=n); i++, j--)
            sum+=b[i][j];
        if(sum>1)
            flag=0;
    }//这两个循环是用来检查从右上到左下的斜线的
    for(int i1=1, j1=n; i1<=n&& j1>=1&& flag; i1++)
    {
        sum=0;
        for(int i=i1, j=j1; (i>=1&&i<=n)&&(j>=1&&j<=n); i++, j++)
            sum+=b[i][j];
        for(int i=i1-1, j=j1-1; (i>=1&&i<=n)&&(j>=1&&j<=n); i--, j--)
            sum+=b[i][j];
        if(sum>1)
            flag=0;
    }
    for(int i1=n, j1=n; i1<=n&& j1>=1&& flag; j1--)
    {
        sum=0;
        for(int i=i1, j=j1; (i>=1&&i<=n)&&(j>=1&&j<=n); i++, j++)
            sum+=b[i][j];
        for(int i=i1-1, j=j1-1; (i>=1&&i<=n)&&(j>=1&&j<=n); i--, j--)
            sum+=b[i][j];
        if(sum>1)
            flag=0;
    }//这两个循环是用来检查从左上到右下的斜线的
    flag?cout<<"Move Forward!\n":cout<<"Bad End!\n";

```

```
    }  
}
```

//其实这个题不难……但是代码量比较大= =所以建议大家自己动手敲一遍。我这个代码比较啰嗦，大家很容易就能写出比这个简单的。

Arthur: 对上面的做进一步简化可得到下面这个，其中 **check** 函数检查对于某个执旗步兵的位置，八个方向是否合法。这里的 **l** 代表与这个点相差的步数。

```
#include<iostream>  
using namespace std;  
int n;  
char a[50][50];  
bool check(int i,int j)  
{  
    int counter=0;  
    for(int l=0;l<n;l++)  
        if(a[i][l]=='1')    //横行  
            counter++;  
    if(counter>1) return false; //超过一个，不合法  
    counter=0;  
    for(int l=0;l<n;l++)  
        if(a[l][j]=='1')    //竖列  
            counter++;  
    if(counter>1) return false; //超过一个，不合法  
    counter=0;  
    for(int l=0;(i+l)<n&&(j+l)<n;l++) //右下斜行  
        if(a[i+l][j+l]=='1')  
            counter++;  
    if(counter>1) return false; //超过一个，不合法  
    counter=0;  
    for(int l=0;(i-l)>=0&&(j-l)>=0;l++)  
        if(a[i-l][j-l]=='1')    //左上斜行  
            counter++;  
    if(counter>1) return false; //超过一个，不合法  
    counter=0;  
    for(int l=0;(i+l)<n&&(j-l)>=0;l++)  
        if(a[i+l][j-l]=='1')    //左下斜行  
            counter++;  
    if(counter>1) return false; //超过一个，不合法  
    counter=0;  
    for(int l=0;(i-l)>=0&&(j+l)<n;l++)    //右上斜行
```

```

        if(a[i-l][j+l]=='1')
            counter++;
    if(counter>1) return false;    //超过一个，不合法
    return true;    能运行到这里，说明上面的检查没有问题，合法
}

bool Legal()
{
    bool isLegal=true;    //isLegal 判断位置是否合法
    int counter=0;        //counter 用来判断个数是否合法
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            if(a[i][j]=='1')    //找到一个执旗步兵俑
            {
                counter++;    //计数器加一
                isLegal=check(i,j);    //判断该位置是否合法
                if(!isLegal) return false;    //如果位置不合法，直接返回 false
            }
    return counter==n&&isLegal;
}    //最终返回取决于个数是否正好是 n，以及 isLegal

int main()
{
    while(cin>>n)
    {
        for(int i=0;i<n;i++)
            for(int j=0;j<n;j++)
                cin>>a[i][j];
        Legal()?cout<<"Move Forward!\n":cout<<"Bad End!\n";
    }
}

```