

## P1205 [USAC01.2] 方块转换 Transformations

### 题目描述

一块  $n \times n$  正方形的黑白瓦片的图案要被转换成新的正方形图案。写一个程序来找出将原始图案按照以下列转换方法转换成新图案的最小方式：

转  $90^\circ$ ：图案按顺时针转  $90^\circ$ 。

转  $180^\circ$ ：图案按顺时针转  $180^\circ$ 。

转  $270^\circ$ ：图案按顺时针转  $270^\circ$ 。

反射：图案在水平方向翻转（以中央铅垂线为中心形成原图案的镜像）。

组合：图案在水平方向翻转，然后再按照 1~3 之间的一种再次转换。

不改变：原图案不改变。

无效转换：无法用以上方法得到新图案。

如果有多种可用的转换方法，请选择序号最小的那个。

只使用上述 7 个中的一个步骤来完成这次转换。

### 输入格式

第一行一个正整数  $n$ 。

然后  $n$  行，每行  $n$  个字符，全部为 @ 或 -，表示初始的正方形。

接下来  $n$  行，每行  $n$  个字符，全部为 @ 或 -，表示最终的正方形。

### 输出格式

单独的一行包括 1~7 之间的一个数字（在上文已描述）表明需要将转换前的正方形变为转换后的正方形的转换方法。

### 输入样例

```
3
@-@
---
@@@
@-@
@--
--@
```

输出样例

1

解析

这道题的几个关键字：找规律、模拟、遍历。即，首先找到前四个操作中数据的变化规律，然后挨个模拟七种行为，直到找到答案。

1、旋转90度的规律

原始图像

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

x

顺时针旋转90度

	0	1	2
0	7	4	1
1	8	5	2
2	9	6	3

把对应关系列成表，找规律

n	y1	x1	y2	x2
3	0	0	0	2
3	0	1	1	2
3	0	2	2	2
3	1	0	0	1
3	1	1	1	1
3	1	2	2	1
3	2	0	0	0
3	2	1	1	0
3	2	2	2	0

$y2 = x1$   
 $x2 = n-1-y1$   
 $b[x][n-y-1] = a[y][x]$   
将a点的数据映射到b点

2、旋转180度的规律

原始图像				顺时针旋转180度			
	0	1	2		0	1	2
0	1	2	3	0	9	8	7
1	4	5	6	1	6	5	4
2	7	8	9	2	3	2	1

把对应关系列成表，找规律

n	y1	x1	y2	x2
3	0	0	2	2
3	0	1	2	1
3	0	2	2	0
3	1	0	1	2
3	1	1	1	1
3	1	2	1	0
3	2	0	0	2
3	2	1	0	1
3	2	2	0	0

$y2 = n-1-y1$   
 $x2 = n-1-x1$   
 $b[n-y-1][n-x-1] = a[y][x]$

### 3、旋转270度的规律

原始图像				顺时针旋转270度			
	0	1	2		0	1	2
0	1	2	3	0	3	6	9
1	4	5	6	1	2	5	8
2	7	8	9	2	1	4	7

把对应关系列成表，找规律

n	y1	x1	y2	x2
3	0	0	2	0
3	0	1	1	0
3	0	2	0	0
3	1	0	2	1
3	1	1	1	1
3	1	2	0	1
3	2	0	2	2
3	2	1	1	2
3	2	2	0	2

$$y2 = n-1-x1$$

$$x2 = y1$$

$$b[n-x-1][y] = a[y][x]$$

### 4、镜像

原始图像				镜像后			
	0	1	2		0	1	2
0	1	2	3	0	3	2	1
1	4	5	6	1	6	5	4
2	7	8	9	2	9	8	7

把对应关系列成表，找规律

n	y1	x1	y2	x2
3	0	0	0	2
3	0	1	0	1
3	0	2	0	0
3	1	0	1	2
3	1	1	1	1
3	1	2	1	0
3	2	0	2	2
3	2	1	2	1
3	2	2	2	0

$$x2 = n-x1-1$$

$$y2 = y1$$

$$b[y][n-x-1] = a[y][x]$$

编码

```
#include<bits/stdc++.h>

using namespace std;

//原始字符数组
char a[12][12];
//变化后的字符数组
char b[12][12];
```

```

//待对比的字符数组
char c[12][12];
//临时用数组
char temp[12][12];

int n;

bool compare(char org[12][12], char to[12][12]) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (org[i][j] != to[i][j]) {
                return false;
            }
        }
    }
    return true;
}

//90度旋转
void rotation90(char org[12][12], char to[12][12]) {
    for (int y = 0; y < n; y++) {
        for (int x = 0; x < n; x++) {
            to[x][n - y - 1] = org[y][x];
        }
    }
}

//180度旋转
void rotation180(char org[12][12], char to[12][12]) {
    for (int y = 0; y < n; y++) {
        for (int x = 0; x < n; x++) {
            to[n - y - 1][n - x - 1] = org[y][x];
        }
    }
}

//270度旋转
void rotation270(char org[12][12], char to[12][12]) {
    for (int y = 0; y < n; y++) {
        for (int x = 0; x < n; x++) {
            to[n - x - 1][y] = org[y][x];
        }
    }
}

//镜像
void mirror(char org[12][12], char to[12][12]) {
    for (int y = 0; y < n; y++) {

```

```

        for (int x = 0; x < n; x++) {
            to[y][n - x - 1] = org[y][x];
        }
    }
}

```

//组合方式

```

bool component(char org[12][12], char to[12][12]) {
    //先镜像
    mirror(org, temp);
    //再执行各个旋转
    rotation90(temp, to);
    if (compare(c, to)) {
        return true;
    }
    rotation180(temp, to);
    if (compare(c, to)) {
        return true;
    }
    rotation270(temp, to);
    if (compare(c, to)) {
        return true;
    }
    return false;
}

```

```

int main() {
    scanf("%d", &n);
    //读入原始字符
    for (int y = 0; y < n; y++) {
        for (int x = 0; x < n; x++) {
            cin >> a[y][x];
        }
    }
    //读入对比字符
    for (int y = 0; y < n; y++) {
        for (int x = 0; x < n; x++) {
            cin >> c[y][x];
        }
    }
    rotation90(a, b);
    if (compare(c, b)) {
        cout << 1;
        return 0;
    }
    rotation180(a, b);
    if (compare(c, b)) {
        cout << 2;
        return 0;
    }
}

```

```
}  
rotation270(a, b);  
if (compare(c, b)) {  
    cout << 3;  
    return 0;  
}  
mirror(a, b);  
if (compare(c, b)) {  
    cout << 4;  
    return 0;  
}  
if (component(a, b)) {  
    cout << 5;  
    return 0;  
}  
if (compare(c, b)) {  
    cout << 6;  
    return 0;  
}  
cout << 7;  
return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

