

## 题目：图形对称性判断

### 问题描述

给定一个二维方阵，对应代表一个图形，判断其对称性。请编写五个函数 `checkHorSym`, `checkVerSym`, `checkDiaSym`, `checkAntiDiaSym`, `checkRotSym` 分别判断方阵的水平对称性、垂直对称性、对角对称性、反对角对称性、旋转对称性。函数声明如下：

```
int checkHorSym(char **pA, int n);
int checkVerSym(char **pA, int n);
int checkDiaSym(char **pA, int n);
int checkAntiDiaSym(char **pA, int n);
int checkRotSym(char **pA, int n);
```

返回值：

1 表示符合该对称性，0 表示不符合该对称性。

参数：

`pA`: 输入参数，输入指向该方阵的二维指针

`n`: 表示该二维方阵的维度为  $n \times n$  维

**注意：函数声明已包含在主程序中，不需要自己定义。只需要提交自定义的函数代码。**

主程序如下：

```
#include <stdio.h>
#include <stdlib.h>

int checkHorSym(char **pA, int n);
int checkVerSym(char **pA, int n);
int checkDiaSym(char **pA, int n);
int checkAntiDiaSym(char **pA, int n);
int checkRotSym(char **pA, int n);

int main(void)
{
    int n,i,j,iHorSymFlag=1, iVerSymFlag=1, iDiaSymFlag=1, iAntiDiaSymFlag=1, iRotSymFlag=1;
    char **pA,*pTmp1,*pTmp2;
    scanf("%d", &n);

    pA=(char**) malloc(sizeof(char*)*n);
    for(i=0; i<n; i++)
    {
        pA[i]=(char*) malloc(sizeof(char)*n);
    }
    for(i=0; i<n; i++)
    {
        pTmp1=pA[i];
        for(j=0;j<n;j++)
        {
            if(pA[i][j] != pA[j][i])
                iRotSymFlag=0;
        }
    }
    if(iHorSymFlag==1)
        printf("Horizontal symmetry\n");
    if(iVerSymFlag==1)
        printf("Vertical symmetry\n");
    if(iDiaSymFlag==1)
        printf("Diagonal symmetry\n");
    if(iAntiDiaSymFlag==1)
        printf("Anti-Diagonal symmetry\n");
    if(iRotSymFlag==1)
        printf("Rotation symmetry\n");
}
```

```

        scanf("%c", pTmp1);

        while(*pTmp1==' ' || *pTmp1=='\n')//skip spacing or line break
        {
            scanf("%c", pTmp1);
        }
        pTmp1++;
    }

}

//horizontal symmetry
iHorSymFlag = checkHorSym(pA, n);

//vertical symmetry
iVerSymFlag = checkVerSym(pA, n);

//diagonal symmetry
iDiaSymFlag = checkDiaSym(pA, n);

//antidiagonal symmetry
iAntiDiaSymFlag = checkAntiDiaSym(pA, n);

//rotationally symmetry
iRotSymFlag = checkRotSym(pA, n);

if(iHorSymFlag)
    printf("1 ");
if(iVerSymFlag)
    printf("2 ");
if(iDiaSymFlag)
    printf("3 ");
if(iAntiDiaSymFlag)
    printf("4 ");
if(iRotSymFlag)
    printf("5 ");
printf("\n");

return 0;
}

```

### 主函数输入格式

输入的第一行包含一个整数  $n$ ，表示方阵维度，方阵为  $n \times n$  维。

第二行开始的  $n$  行表示一个方阵，每行包含  $n$  个字符  $a_1, a_2, \dots, a_n$ ，表示给定的方阵，相邻的字符之间用一个空格分隔。

### 主函数输出格式

输出一个或多个整数，表示该方阵对应图形的对称性：1 代表水平对称，2 代表垂直对称，3 代表对角对称（即以左上到右下的线为对称轴），4 代表反对角对称（即以右上到左下的线为对称轴），5 代表旋转对称（即以中心位置为轴旋转  $180^\circ$  后与原图形重合）。整数之间按照值的大小从小到大排列，以空格隔开。

### 样例输入

```
11
00000100000
00001010000
00010001000
00100000100
01000000010
10000000001
01000000010
00100000100
00010001000
00001010000
00000100000
```

### 样例输出

```
1 2 3 4 5
```

### 样例说明

样例中以 0 和 1 两种字符给出了菱形图形，满足所有对称方式。

### 评测用例规模与约定

$1 \leq n \leq 100$ ， $a_i$  可以是 0-9，或者是 a-z、A-Z 这些字母或数字字符。