

二维数组右上左下遍历

人员

温郝冬、罗启宸、董浩桢、王奕皓、周苇杰、马瑞昕、崔宸赫、邹忆航 到课，李沛都、陶汇笙、康佳、张昱霖 线上

作业检查

陶汇笙 已完成

康佳 已完成

董浩桢 已完成

罗启宸 已完成

温郝冬 已完成

周苇杰 已完成

王奕皓 未打卡

崔宸赫 未打卡

马瑞昕 上周请假

邹忆航 第一次进班

郭栩睿 未打卡

张昱霖 未打卡

李沛都 未打卡

作业

作业：**noi 1.8 23:二维数组回形遍历** 和 **东方博宜 1022. 百钱百鸡问题**

下周默写：**noi 1.8 21:二维数组右上左下遍历** 和 **东方博宜 1327. 鲜花方阵**

课堂表现

很多同学，在老师讲课时不集中注意力听课，自己写自己代码。以后一定要先听老师讲，老师讲完之后会给大家留时间写代码。

课堂默写情况

温郝冬、崔宸赫、周苇杰、王奕皓、陶汇笙、李沛都、康佳、董浩桢 默写过了，其中 陶汇笙、王奕皓、李沛都 默写两个题，提出表扬！！

课堂内容

东方博宜 1327. 鲜花方阵

```
#include <iostream>

using namespace std;

int a[15][15];

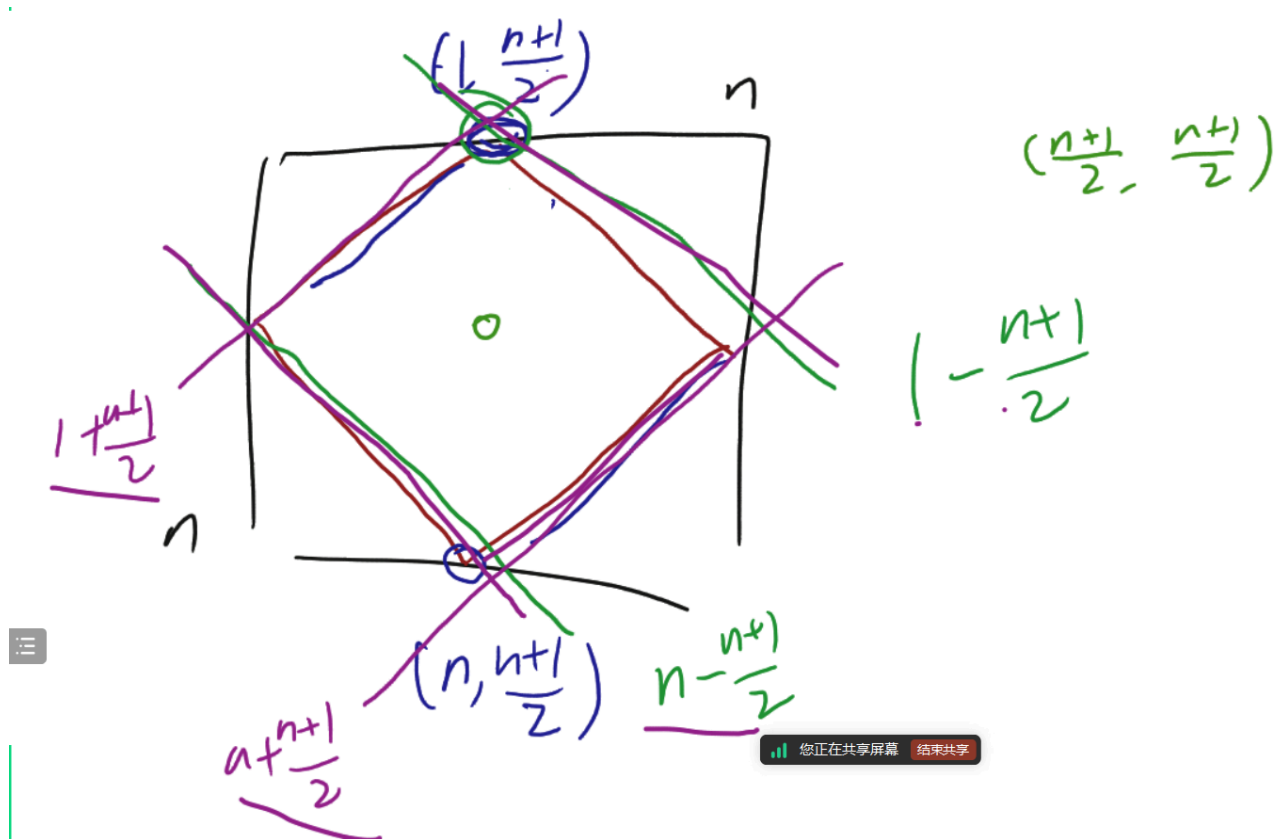
int main() {
    int n; cin >> n;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            a[i][j] = 1;
        }
    }

    int x = (n+1) / 2;

    a[x][x] = 0;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (i+j==x+1 || i+j==x+n || i-j==x-1 || i-j==x-n) {
                a[i][j] = 0;
            }
        }
    }

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            printf("%3d", a[i][j]);
        }
        cout << endl;
    }
    return 0;
}
```

noi 1.8 21:二维数组右上左下遍历



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

using namespace std;

int a[105][105];

int main()
{
    int row, col;
    cin >> row >> col;
    for (int i = 1; i <= row; i++) {
        for (int j = 1; j <= col; j++) {
            cin >> a[i][j];
        }
    }

    for (int k = 2; k <= row+col; k++) {
        // 输出第 k 条线
        // 遍历整个数组，找到那些 行+列==k 的点
        for (int i = 1; i <= row; i++) {
            for (int j = 1; j <= col; j++) {
                if (i+j == k) {
                    cout << a[i][j] << endl;
                }
            }
        }
    }
    return 0;
}
```

noi 1.8 23:二维数组回形遍历

1. 4个方向, 对于每个方向, 要有办法沿着这个方向走

上: $i--$, j 不变

下: $i++$, j 不变

左: i 不变, $j--$

右: i 不变, $j++$

$dx[] = \{0, 1, 0, -1\}$

$dy[] = \{1, 0, -1, 0\}$

(x, y) 往右走一格 $\rightarrow x+=dx[0], y+=dy[0]$

(x, y) 往下走一格 $\rightarrow x+=dx[1], y+=dy[1]$

(x, y) 往左走一格 $\rightarrow x+=dx[2], y+=dy[2]$

(x, y) 往上走一格 $\rightarrow x+=dx[3], y+=dy[3]$

2. 方向要变: $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$

$fx = 0$

转方向:

$fx++$

if ($fx==4$) {

$fx = 0;$

}

$\rightarrow fx = (fx+1)\%4$

1/2 结合起来:

$x += dx[fx], y += dy[fx]$

3. 什么时候转弯?

1. 到边上转弯

2. 如果下一个要走的位置已经走过了, 转弯 \rightarrow 输出完一个数, 就把这个位置的数变成-1

\rightarrow 综合 1/2 步, 只需要把外面一圈变为-1, 走过的位置变为-1

```
int cnt = 0;
int x = 1, y = 1, fx=0;
while (cnt < n*m) {
    cout << a[x][y];
    cnt++;
    a[x][y]=-1;
    x+=dx[fx], y+=dy[fx];
    if (a[x][y] == -1) {
        x-=dx[fx], y-=dy[fx];
        fx=(fx+1)%4;
        x+=dx[fx], y+=dy[fx];
    }
}
```

此题目不提供完整代码，要求同学们课下思考实现