

试题编号	66
试题名称	期末测试 2-题目 3-Z 字形扫描
时间限制:	1 秒
内存限制:	256MB
问题描述	
问题描述	

对于下面给出的 4×4 的矩阵,

3 7 5 6

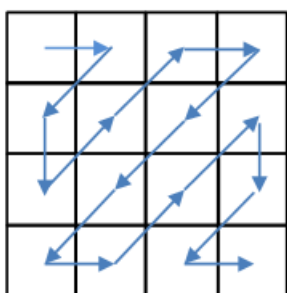
9464

7313

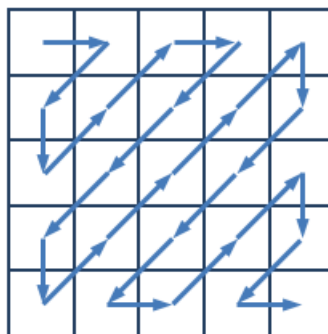
对其进行 Z 字形扫描后得到长度为 16 的序列:

1539739547366413

请实现一个 Z 字形扫描的程序，给定一个 $n \times n$ 的矩阵，输出对这个矩阵进行 Z 字形扫描的结果。



4×4矩阵Z字形扫描



5×5 矩阵Z字形扫描

输入说明

输入的第一行包含一个整数 n ，表示矩阵的大小。

输入的第二行到第 $n+1$ 行每行包含 n 个正整数，由空格分隔，表示给定的矩阵。

$1 \leq n \leq 100$, 矩阵元素为不超过 1000 的正整数

输出说明

输出一行, 包含 $n \times n$ 个整数, 由空格分隔, 表示输入的矩阵经过 Z 字形扫描后的结果。

输入样例

4

1 5 3 9

3 7 5 6

9 4 6 4

7 3 1 3

输出样例

1539739547366413

提示信息

```

*****/

#include <stdio.h>
// 去下行注释则禁用 assert()
// #define NDEBUG
#include <assert.h>

int main()
{
    int n, a[100][100];
    //int n = 4, a[4][4] = { {1,5,3,9},{3,7,5,6},{9,4,6,4},{7,3,1,3} };

    // Finite State Machine
    // last,next = 1,水平(左-->右), 2 竖直(上-->下), 3 斜下(右上-->左下), 4 斜上(左下-->右上)
    int i = 0, j = 0, last = 1, next = 1;

    scanf("%d",&n);
    for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
            scanf("%d",&a[i][j]);

    i = 0; j = 0;
    printf("%d ", a[i][j]);
    while (1)
    {
        if (i == n - 1 && j == n - 1) break;
        last = next;
        switch (last)
        {
            case 1: // 水平
                j++;
                if (i == 0) next = 3; // 转斜下
                else if (i == n - 1) next = 4; // 转斜上
                break;
            case 2: // 竖直
                i++;
                if (j == 0) next = 4; // 转斜上
                else if (j == n - 1) next = 3; // 转斜下
                break;
            case 3: // 斜下
                i++; j--;
                if (i > n - 1) i = n - 1;
                if (j < 0) j = 0;

                if (j == 0)

```

```

        {
            if (i == n - 1) next = 1; // 转水平
            if (i < n - 1) next = 2; // 转竖直
        }
        else if (i == n - 1) next = 1; // 转水平
        break;
case 4: // 斜上
    i--; j++;
    if (i < 0) i = 0;
    if (j > n - 1) j = n - 1;
    if (i == 0)
    {
        if (j == n - 1) next = 2; // 转竖直
        if (j < n - 1) next = 1; // 转水平
    }
    else if (j == n - 1) next = 2; // 转竖直
    break;
    }
    printf("%d ", a[i][j]); // 输出 last 状态下的 a[i][j]
}

return 0;
}

int main1()
{
    //int n, a[100][100];
    int n = 4, a[4][4] = { {1,5,3,9},{3,7,5,6},{9,4,6,4},{7,3,1,3} };

    // Finite State Machine
    // last,next = 1,水平(左-->右), 2 竖直(上-->下), 3 斜下(右上-->左下), 4 斜上(左下-->右上)
    int i = 0, j = 0, last = 1, next = 1;
    printf("%d ", a[0][0]);
    while (1)
    {
        if (i == n - 1 && j == n - 1) break;
        last = next;
        switch (last)
        {
            case 1: // 水平
                assert(i == 0 || i == n - 1);
                assert(j < n - 1);
                j++;
                if (i == 0) next = 3; // 转斜下

```

```

        else if (i == n - 1) next = 4; // 转斜上
        break;
    case 2: // 竖直
        assert(j == 0 || j == n - 1);
        assert(i < n - 1);
        i++;
        if (j == 0) next = 4; // 转斜上
        else if (j == n - 1) next = 3; // 转斜下
        break;
    case 3: // 斜下
        i++; j--;
        if (i > n - 1) i = n - 1;
        if (j < 0) j = 0;

        if (j == 0)
        {
            if (i == n - 1) next = 1; // 转水平
            if (i < n - 1) next = 2; // 转竖直
        }
        else if (i == n - 1) next = 1; // 转水平
        break;
    case 4: // 斜上
        i--; j++;
        if (i < 0) i = 0;
        if (j > n - 1) j = n - 1;
        if (i == 0)
        {
            if (j == n - 1) next = 2; // 转竖直
            if (j < n - 1) next = 1; // 转水平
        }
        else if (j == n - 1) next = 2; // 转竖直
        break;
    }
    printf("%d ", a[i][j]); // 输出 last 状态下的 a[i][j]
}

return 0;
}

```