

# 有趣的数字图形

---

## 人员

---

温郝冬、李沛都、罗启宸、董浩桢、王奕皓、周苇杰、陶汇笙、康佳、郭栩睿、崔宸赫 到课

## 作业

---

必做：东方博宜 1327. 鲜花方阵

选做：noi 1.8 21:二维数组右上左下遍历

下节课默写 东方博宜 1207 - 有趣的数字图形IV 或者 东方博宜 2000 - 两个数相邻吗？

## 课堂表现

---

同学们课上都比较认真，不过大家自己找错误的能力还有待提升，目前遇到问题只会找老师解决

要培养 遇到问题自己先进行分析找错误 的习惯

## 课堂默写情况

---

王奕皓、李沛都、张昱霖、罗启宸、崔宸赫、康佳、温郝冬、郭栩睿、董浩桢 8位同学都默写出来了，提出表扬！！！！

## 课堂内容

---

**东方博宜 1203 - 拐角VIII (上周作业)**

先构造下面右边的二维数组，然后

把第1行所有比1小的数变成1

把第2行所有比2小的数变成2

把第3行所有比3小的数变成3

...

把第n行所有比n小的数变成n

```

#include <bits/stdc++.h>

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] = n-j+1;
        }
    }

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

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

```

## 东方博宜 1207 - 有趣的数字图形IV

```

          4
        3  7
      2  5 12
    1  3  8 20

```

1. 斜着有 1 2 3 ... n (方法)
2. 把右下角赋值，等于 左边 + 上边

判断点 (i,j) 是不是在右下角 -> if (i+j > n+1)  
 $a[i][j] = a[i][j-1] + a[i-1][j]$

```

#include <bits/stdc++.h>

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++) {
            if (i+j == n+1) {
                a[i][j] = j;
            }
        }
    }

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

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (i+j < n+1) {
                cout << "    ";
            }
            else {
                printf("%5d", a[i][j]);
            }
        }
        cout << endl;
    }
    return 0;
}

```

东方博宜 2000 - 两个数相邻吗？

```

int ai, aj, bi, bj;
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= m; j++) {
        if (w[i][j] == a) {
            ai = i;
            aj = j;
        }
        if (w[i][j] == b) {
            bi = i;
            bj = j;
        }
    }
}

```

a -> (ai, aj)

b -> (bi, bj)

a, b 是否相邻 -> 一个相等, 另一个挨着

```

if ((ai==bi && aj-bj==1) || (aj==bj && ai-bi==1) || (aj==bj && bi-ai==1) || (ai==bi &&
bj-aj==1)) {
    cout << "Y";
}
else {
    cout << "N";
}

```

if (ai==bi && aj-bj==1) // a在右边

if (aj==bj && ai-bi==1) // a在下边

if (aj==bj && bi-ai==1) // a在上边

if (ai==bi && bj-aj==1) // a在左边

```

#include <bits/stdc++.h>

using namespace std;

int a[205][205];

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

    int i1, j1, i2, j2;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            if (a[i][j] == x) {
                i1 = i, j1 = j;
            }
            if (a[i][j] == y) {
                i2 = i, j2 = j;
            }
        }
    }

    if ((i1==i2&&j1-j2==1) || (i1==i2&&j2-j1==1) || (j1==j2&&i1-i2==1) || (j1==j2&&i2-
i1==1)) {
        cout << "Y" << endl;
    }
    else {
        cout << "N" << endl;
    }
    return 0;
}

```