

二维数组右上左下遍历

人员

韩承睿、刘嘉航、辛帅辰、李翰如、方俊喆、崔吉诺、刘祺、刘峰烁、秦显森、齐振玮、牟茗、高健桓、谢亚锴、牛同泽、徐浩然 到课

作业

noi 1.8 21: 二维数组右上左下遍历

复习课上的 东方博宜.2000 和 noi 1.8 02 两道题目，下节课上课会要求默写

课堂表现

今天课堂题目有些难度，同学们课下要好好复习

课堂内容

文件输入/输出

```
#include <cstdio>

using namespace std;

int main() {
    freopen("abc.in", "r", stdin);
    freopen("abc.out", "w", stdout);
    ...
    fclose(stdin); // 可以不写
    fclose(stdout); // 可以不写
    return 0;
}
```

2000 - 两个数相邻吗?

输入两个数x,y

1. 先找到 数x的位置(i1,j1) 和 数y的位置(i2,j2)

2. 判断 (i1,j1) 和 (i2,j2) 是否相邻

-> 相邻的条件?

-> 横坐标相同, 纵坐标差1

-> 纵坐标相同, 横坐标差1

```
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;
        }
    }
}
```

(i1, j1) (i2, j2)

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

```

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

```

noi 1.8 02:同行列对角线的格子

1. 输出同行，较简单
2. 输出同列，较简单
3. 输出 左上到右下，所有点的 横坐标-纵坐标 相同（从上到下遍历）
4. 输出 左下到右上，所有点的 横坐标+纵坐标 相同（从下到上遍历）

```

#include <iostream>

using namespace std;

int a[15][15];

int main() {
    int n, x, y;
    cin >> n >> x >> y;
    for (int i = 1; i <= n; i++) printf("(%d,%d) ", x, i);
    cout << endl;

    for (int i = 1; i <= n; i++) printf("(%d,%d) ", i, y);
    cout << endl;

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (i-j == x-y) {
                printf("(%d,%d) ", i, j);
            }
        }
    }
    cout << endl;

    for (int i = n; i >= 1; i--) {
        for (int j = 1; j <= n; j++) {
            if (i+j == x+y) {
                printf("(%d,%d) ", i, j);
            }
        }
    }
    cout << endl;
    return 0;
}

```

noi 1.8 21:二维数组右上左下遍历

1. 枚举第 i 斜 (i 从 $2 \sim n+m$ 遍历)
2. 对于第 i 斜来说, 从 $(1, i-1)$ 往左下角遍历。每次让 横坐标 $++$, 纵坐标 $--$

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 100 + 5;
int w[maxn][maxn];

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

    for (int i = 2; i <= n+m; i++) {
        int x = 1, y = i-1;
        while (1) {
            if (x>=1 && x<=n && y>=1 && y<=m) {
                cout << w[x][y] << endl;
            }
            x++, y--;
            if (x>n || y<1) {
                break;
            }
        }
    }
    return 0;
}

```