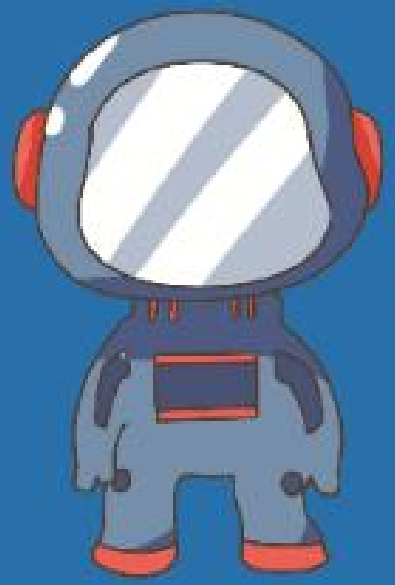
学习目标

1. 了解图形模拟的基本方法

2. 掌握图形模拟的经典题型



知识讲授

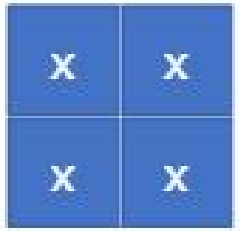
图形模拟的概念

通过代码实现图形的变化，如放大、缩小、旋转等。这里说的图形大多数可以看成是二维数组，主要是对二维数组行下标和列下标的深入研究。

图形的放大

放大的概念是指其中任意一个元素都变成n行n列的矩阵。

某一个点（一个矩阵元素）扩大两倍可以看成是下图所示：



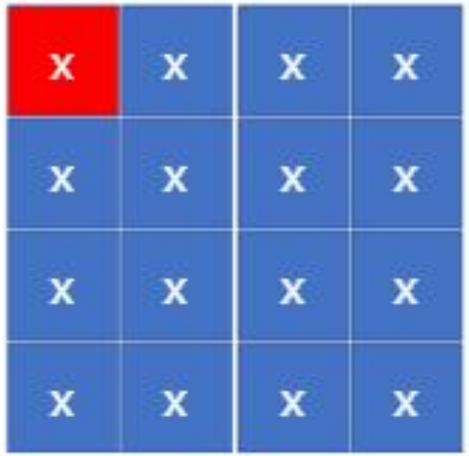
扩大n倍就是把一个元素变为n行n列，这样每一个元素的新位置都可以计算

图形的放大

点（x，y）放大成n行n列的矩阵，其左上角起始点的坐标我们可以这样计算：

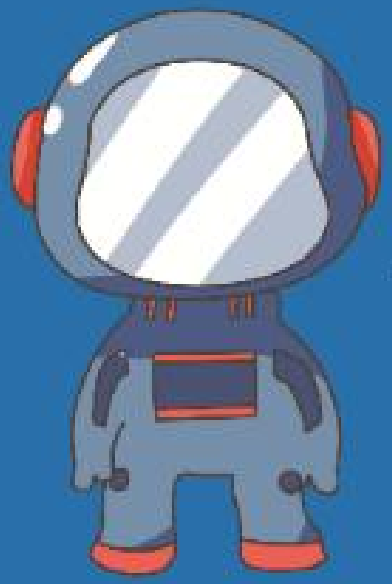
以上（x-1）个点，占用（x-1）*n 行

以左（y-1）个点，
占用（y-1）*n 列



（x，y）的占用的新位置：
x坐标: (x-1)*n+1 —— (x-1)*n+n
y坐标: (y-1)*n+1 —— (y-1)*n+n

找到新起点在新的二维数组内填写n行n列数据即可
放大以后的数组行和列都变为原先的n倍



课堂练习

图形的放大

【描述】 现在有一个全部是整数组成的矩阵，为了让大家看清，希望能把它放大n倍，（放大的概念是指其中任意一个元素都变成n行n列的矩阵）。

【输入】第1行是m，n，k，代表m行n列的矩阵，需要放大k倍。后面是一个m*n的矩阵输入。保证放大以后的矩阵不超过100*100；

【输出】放大以后的矩阵

【样例输入】

2 2 2

1 2

3 4

【样例输出】

1 1 2 2

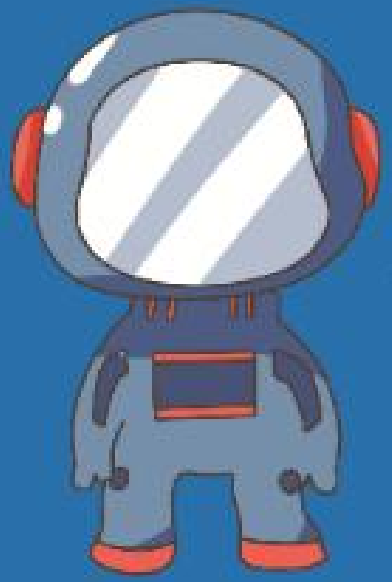
1 1 2 2

3 3 4 4

3 3 4 4

参考代码

```
using namespace std;
int a[110][110];
int ans[110][110];
int m,n,k;
int main(){
    cin>>m>>n>>k;
    for(int i=1;i<=m;i++){
            for(int j=1;j<=n;j++){
                    cin>>a[i][j];
                    int x=(i-1)*k;
                    int y=(j-1)*k;
                    for(int u=x+1;u<=x+k;u++){
                            for(int v=y+1;v<=y+k;v++){
                                    ans[u][v]=a[i][j];
                                    }
                            }
                    }
            }
    for(int i=1;i<=m*k;i++){
            for(int j=1;j<=n*k;j++){
                    cout<<ans[i][j]<<" ";
            }
            cout<<endl;
}
    return 0;
}
```

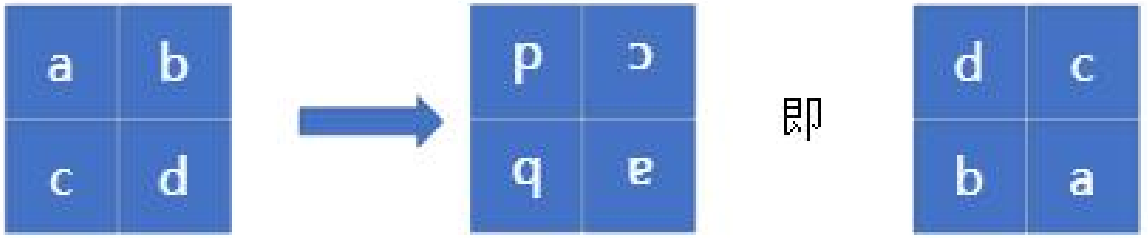


知识讲授

图形的旋转

图形的旋转，即是指其中任意一个元素都绕中心点旋转一定的角度到达新的位置

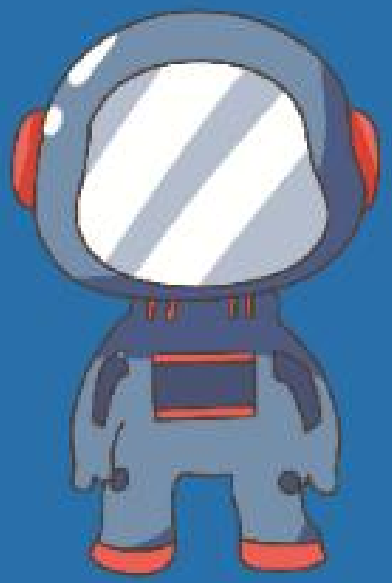
以2行2列方阵顺时针旋转180°为例：



进行图形旋转，即根据实际情况计算原来元素到了新位置的坐标

对于顺时针旋转180°，行和列都逆序了：

（i，j）=>（m-i+1 , n - j+1）



课堂练习

图形的旋转

【描述】 现在有一个m行n列的全部是**字符**组成的矩阵，需要将这个图案顺时针旋转90度，旋转完毕看是什么样子的图案

【输入】第1行是m，n，代表m行n列的矩阵；后面是一个m*n的矩阵输入。矩阵不超过100*100；

【输出】旋转后的矩阵

【样例输入】

2 2

1 2

3 4

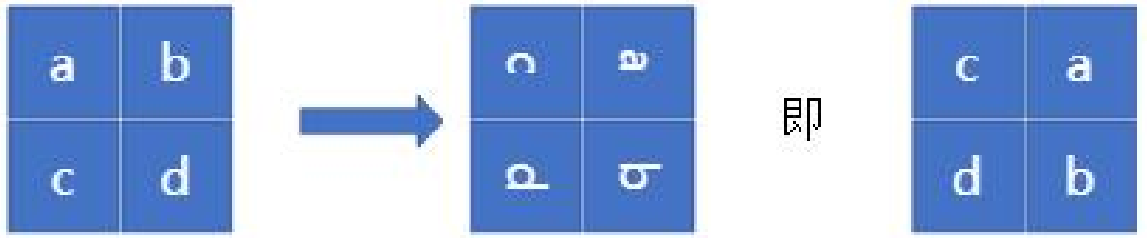
【样例输出】

3 1

4 2

图形的旋转

顺时针旋转90°



第1行=>最后一列　　第1列=>第1行，逆序

第2行=>倒数第2列　　第2列=>第2行，逆序

行号变换为新的列号，列号变换为新的行号

（i，j）=>（j，n-i+1）

参考代码

```
#include <iostream>
using namespace std;
char a[101][101], b[101][101];
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];
        b[j][n+1-i] = a[i][j];
                        }
            }
    for(int i=1;i<=m;i++){//n行m列变成了m行n列
        for(int j=1; j<=n; j++) cout<<b[i][j]<<" ";
        cout << endl;
    }
     return 0;
}
```

图形的旋转-旋转180°

【描述】 现在有一个m行n列的全部是**字符**组成的矩阵，需要将这个图案顺时针旋转180度，旋转完毕看是什么样子的图案

【输入】第1行是m，n，代表m行n列的矩阵；后面是一个m*n的矩阵输入。矩阵不超过100*100；

【输出】旋转后的矩阵

【样例输入】

2 2

1 2

3 4

【样例输出】

4 3

2 1

参考代码

```
#include <iostream>
using namespace std;
char a[101][101], b[101][101];
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];
                b[n-i+1][m-j+1] = a[i][j];
        }
    }
    for(int i=1;i<=n;i++){
        for(int j=1; j<=m; j++) cout<<b[i][j]<<" ";
        cout << endl;
    }
     return 0;
}
```

图形的左右翻转

【描述】 现在有一个m行n列的全部是**字符**组成的矩阵，需要将这个图案左右翻转，翻转完毕看是什么样子的图案

【输入】第1行是m，n，代表m行n列的矩阵；后面是一个m*n的矩阵输入。矩阵不超过100*100；

【输出】左右翻转后的矩阵

【样例输入】

2 2

1 2

3 4

【样例输出】

2 1

4 3

参考代码

```
#include <iostream>
using namespace std;
char a[101][101], b[101][101];
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];
            b[i][m-j+1] = a[i][j];
        }
    }
    for(int i=1;i<=n;i++){
        for(int j=1; j<=m; j++) cout<<b[i][j]<<" ";
        cout << endl;
    }
    return 0;
}
```

小结

1. 图形变化的重点是什么？

2. 二维数组除了int类型还有什么类型

3. 旋转和翻转的区别

作业

1、矩阵转置　http://noi.openjudge.cn/ch0108/10/

2、图像旋转　http://noi.openjudge.cn/ch0108/11/

3、变幻的矩阵　http://noi.openjudge.cn/ch0108/12/

4、图像模糊处理　http://noi.openjudge.cn/ch0108/13/