二维数组

二维数组基础

```
#include<bits/stdc++.h>//1-1272      郭远摘苹果    javacn
using namespace std;                  // 思路：求二维数组的最大数和最小数的差值
int main() {
      int m,n,mi,ma,a[20][20];        //m 行，n 列，mi 最小值，ma 最大值，
      cin>>m>>n;
      for(int i=0;i<m;i++)            // 循环读入数组元素 // 先循环行
      {
            for(int j=0;j<n;j++)      // 再循环列
            {
                  cin>>a[i][j];
            }
      }
      ma = a[0][0];      // 假设第一个数是最大的和最小的
      mi = a[0][0];
      for(int i=0;i<m;i++)     // 循环数组元素，找最大的和最小的值
      {
            for(int j=0;j<n;j++)
            {
                  if(a[i][j]>ma)
                  {
                        ma = a[i][j];
                  }
                  if(a[i][j]<mi)
                  {
                        mi = a[i][j];
                  }
            }
      }
      cout<<ma-mi;
      return 0;
}
```

思路：针对每一列的数据求和

```
#include<bits/stdc++.h>//2-1274-1  javacn
using namespace std;
int main()
{
	//n 代表班级人数，s1 代表语文总成绩，s2 代表数学总成绩，s3 代表英语总成绩
	int n,a[100][3],i,j,s1=0,s2=0,s3=0;
	// 输入总人数
	cin>>n;
	// 读入每个人的成绩
	for(i=0;i<n;i++)
	{
		for(j=0;j<3;j++)
		{
			cin>>a[i][j];
		}
		// 求单科总成绩
		s1+=a[i][0];
		s2+=a[i][1];
		s3+=a[i][2];
	}
	// 求各科平均成绩，结果保留 1 位小数
	cout<<fixed<<setprecision(1);
	cout<<s1*1.0/n<<" ";
	cout<<s2*1.0/n<<" ";
	cout<<s3*1.0/n<<endl;

	return 0;
}
```

求各个科目成绩的平均分

```
#include<bits/stdc++.h>//2-1274-2  jiangyf70
using namespace std;
int main()
{
    int n, a[110][3], sx = 0, yw = 0, yy = 0;
    cin >> n;
    for(int i = 0; i < n; i++)
    {
        cin >> a[i][1] >> a[i][2] >> a[i][3];
        sx += a[i][1];
        yw += a[i][2];
        yy += a[i][3];
    }
    printf("%.1lf %.1lf %.1lf", sx * 1.0 / n, yw * 1.0 / n, yy * 1.0 / n);
    return 0;
}
```

输出杨辉三角的前 N 行，第 i 行有 i 个数。

每行的第 1 个数是 1，其余数字 = 其上方数 + 左侧数。

```
#include<bits/stdc++.h>//3-1275  javacn
using namespace std;
// 二维数组初始值清 0
int a[10][10];
int main()
{
	int n,i,j;
	cin>>n;
	for(i=0;i<n;i++)  // 先循环行
	{
		for(j=0;j<=i;j++) // 每行有 i+1 列
		{
			if(j==0)     // 第一列都为 1
			{
				a[i][j]=1;
			}
			else
			{
				// 该数 = 该数上方数 + 该数上方左侧数
				a[i][j]=a[i-1][j]+a[i-1][j-1];
			}
		}
	}
	for(i=0;i<n;i++)  // 输出数组元素
	{
		for(j=0;j<=i;j++)
		{
			cout<<a[i][j]<<" ";
		}
		cout<<endl;
	}
	return 0;
}
```

地雷数量求解

思路：遍历二维数组，求出各个位之和，判断是否是奇数。（本题也可以读一个数处理一个数，见解法二）

```
#include <bits/stdc++.h>//4-1496-1  javacn
using namespace std;
int digSum(int x)// 求 x 的各个位之和
{
	int r = 0;
	while(x!=0)
	{
		r += x%10;
		x = x/10;
	}
	return r;
}
int main(){
	int n,m,a[101][101],i,j,num=0;//num 表示奇数的个数
	cin>>n>>m;
	for(i=0;i<n;i++)  // 读入这个二维数组
	{
		for(j=0;j<m;j++)
		{
			cin>>a[i][j];
		}
	}
	for(i=0;i<n;i++)  // 遍历这个二维数组
	{
		for(j=0;j<m;j++)  // 判断各个位之和是否是奇数
		{
			if(digSum(a[i][j])%2!=0)	{	num++;	}
		}
	}
	cout<<num;
	return 0;
}
```

解法二：读一个数，判断一个数。

```
#include <bits/stdc++.h>//4-1496-2  javacn
using namespace std;
int x,n,m,s,c = 0;
int main()
{
	cin>>n>>m;
	for(int i = 1;i <= n;i++)
	{
		for(int j = 1;j <= m;j++)
		{
			cin>>x;
			s = 0;// 求各个位的和
			while(x != 0)
			{
				s = s + x % 10;// 求各个位的和
				x = x / 10;
			}

			// 如果各个位的和是奇数，说明有地雷
			if(s % 2 != 0)
				c++;
		}
	}

	cout<<c;
	return 0;
}
```

```cpp
#include<bits/stdc++.h>//5-1407 图像相似度 javacn
using namespace std;
int main()
{
	int a[100][100],b[100][100],i,j,n,m,c=0;
	cin>>n>>m;
	for(i=0;i<n;i++)  // 读入第一幅图像的值
	{
		for(j=0;j<m;j++)
		{
			cin>>a[i][j];
		}
	}
	for(i=0;i<n;i++)  // 读入第二幅图像的值
	{
		for(j=0;j<m;j++)
		{
			cin>>b[i][j];
		}
	}
	for(i=0;i<n;i++)  // 统计相同点的数量
	{
		for(j=0;j<m;j++)
		{
			if(a[i][j]==b[i][j])
			{
				c++;
			}
		}
	}

	double r = c*100.0/(n*m);
	cout<<fixed<<setprecision(2)<<r<<endl;
	return 0;
}
```

求最大梯形的面积

解法一：使用二维数组求解

```
#include<bits/stdc++.h>//6-1330-1  javacn  推荐学习
using namespace std;
// 至少有 3 列，所以定义 4（从 1 开始）
int a[101][4];
int main()
{
	int n,i;
	double s,mx=0; // s: 临时面积 mx: 最大梯形面积
	cin>>n;
	for(i=1;i<=n;i++)
	{	// 第 i 个梯形，存入 3 列数据
		cin>>a[i][1]>>a[i][2]>>a[i][3];
	}
	// 都存入二维数组后，一个一个梯形计算面积并比较
	for(i=1;i<=n;i++)
	{
		s = (a[i][1]+a[i][2])*a[i][3]*1.0/2;
		if( s > mx )
		{
			mx = s;
		}
	}
	cout<<fixed<<setprecision(1)<<mx;
	return 0;
}
```

解法二：读入一组，计算一组，打擂台求最大

```
#include<bits/stdc++.h>//6-1330-2  javacn
using namespace std;

/*
本题可以用二维数组读入，也可以不用。
读入一组，计算出对应的面积，打擂台求最大
*/
int n;
int a,b,h;// 上底、下底、高
double ma;// 最大的面积
double s;// 每个梯形的面积

int main()
{
	cin>>n;
	// 读入 n 组数据
	for(int i = 1;i <= n;i++)
	{
		cin>>a>>b>>h;
		// 计算面积
		s = (a + b) * h / 2.0;
		if(s > ma)
		{
			ma = s;
		}
	}

	cout<<fixed<<setprecision(1)<<ma;
}
```

靶心数

思路：比较时遍历除了第一行、最后一行、第一列、最后一列的值

```
#include<bits/stdc++.h>//7-1384  javacn
using namespace std;
int main()
{
	int n,m,a[101][101],j,i;
	cin>>n>>m;

	for(i=0;i<n;i++)		// 读入这个二维数组 // 先循环行
	{
		for(j=0;j<m;j++)		// 再循环列
		{
			cin>>a[i][j];
		}
	}
	// 遍历除了第一行、最后一行、第一列、最后一列的值
	for(i=1;i<n-1;i++)
	{
		for(j=1;j<m-1;j++)
		{
			// 判断是否比四个方向的值都大
			if(a[i][j]>a[i-1][j]&&a[i][j]>a[i-1][j]&&a[i][j]>a[i]
[j-1]&&a[i][j]>a[i][j+1])
			{
				cout<<a[i][j]<<endl;
			}
		}
	}
	return 0;
}
```

奇偶统计

思路：读入二维数组，遍历这个二维数组，判断是奇数还是偶数，并计数

```
#include<bits/stdc++.h>//8-1398  javacn
using namespace std;
int main()
{
	//s1 表示奇数的个数，s2 表示偶数的个数
	int n,m,a[101][101],i,j,s1=0,s2=0;
	cin>>n>>m;
	for(i=0;i<n;i++)		// 读入二维数组 // 先循环行
	{
		for(j=0;j<m;j++)  // 再循环列
		{
			cin>>a[i][j];
		}
	}

	for(i=0;i<n;i++)  // 遍历二维数组
	{
		for(j=0;j<m;j++)
		{
			if(a[i][j]%2==0)  // 判断是偶数还是奇数
			{
				s2++;
			}
			else
			{
				s1++;
			}
		}
	}

	cout<<s1<<" "<<s2;
	return 0;
}
```

思路：遍历二维数组，判断是否是回文数

```
#include<bits/stdc++.h>//9-1403-1 javacn
using namespace std;
bool hws(int x)// 判断是否是回文数
{
	bool r = false;
	int y=0,m=x;
	while(m!=0) //y=x 的倒序
	{
		y = y*10+m%10;
		m = m/10;
	}
	if(y==x)
		r = true;
	return r;
}
int main()
{
	int n,m,a[101][101],i,j;
	cin>>n>>m;
	for(i=0;i<n;i++)		// 读入二维数组 // 先循环行
	{
		for(j=0;j<m;j++)		// 再循环列
			cin>>a[i][j];
	}
	for(i=0;i<n;i++)  // 按照读入顺序遍历二维数组，判断是否是回文数，是就输出
	{
		for(j=0;j<m;j++)
		{
			if(hws(a[i][j]))
				cout<<a[i][j]<<endl;
		}
	}
	return 0;
}
```

找回文数？

```
#include<bits/stdc++.h>//9-1403-2 cfxt
using namespace std;
int huiw(int x)
{
	int s=0;
	while(x)
	{
		s=s*10+x%10;
		x=x/10;
	}
	return s;
}
int main()
{
	int n,m,a[110][110];
	cin>>n>>m;
	for(int i=0;i<n;i++)
	{
		for(int j=0;j<m;j++)
		{
			cin>>a[i][j];
		}
	}
	for(int i=0;i<n;i++)
	{
		for(int j=0;j<m;j++)
		{
			if(huiw(a[i][j])==a[i][j])
				cout<<a[i][j]<<endl;
		}
	}
	return 0;
}
```

思路：比赛成绩作为二维数组输入，计数 a,b 赢的局数

```
#include<bits/stdc++.h>//10-1406-1 javacn  推荐学习
using namespace std;
int main()
{
	int n,i,j,a[101][3],aw=0,bw=0;		//aw 代表 a 赢的局数，bw 代表 b 赢的局数
	cin>>n;
	for(i=0;i<n;i++)		// 读入二维数组 // 先循环行
	{
		for(j=0;j<2;j++)		// 再循环列
		{
			cin>>a[i][j];
		}
		// 判断比赛结果谁输谁赢
		if((a[i][0]==1&&a[i][1]==2)||(a[i][0]==2&&a[i][1]==3)||(a[i][0]==3&&a[i][1]==1))
		{
			aw++;
		}
		else if((a[i][0]==1&&a[i][1]==3)||(a[i][0]==2&&a[i][1]==1)||(a[i][0]==3&&a[i][1]==2))
		{
			bw++;
		}
	}

	if(aw>bw)
		cout<<"a win";
	else if(aw<bw)
		cout<<"b win";
		else
			cout<<"tie";

	return 0;
}
```

石头剪刀布？

```
#include<bits/stdc++.h>//10-1406-2 jiangyf70
using namespace std;
int main()
{
	int n, a, b, sum = 0;
	cin >> n;
	for(int i = 0; i < n; i++)
	{
		cin >> a >> b;
		a--;
		b--;
		if(a != b)
		{
			if((a + 1) % 3 == b)
				sum++;
			else
				sum--;
		}

	}

	if(sum > 0)
		cout << "a win";
	else if(sum < 0)
		cout <<"b win";
		else
			cout << "tie";
    return 0;
}
```

思路：用字符型二维数组读入地图，遍历地图的每一格，如果是*（地雷）直接输出
如果是?(非地雷格)，计算其八方向中有多少个地雷并输出

```
#include<bits/stdc++.h>//11-1580-1 javacn 推荐学习
using namespace std;
char a[110][110];// 存储地图
int n,m;
int main() {
	cin>>n>>m;
	int i,j,c;
	for(i = 1;i <= n;i++){  // 读入地图
		for(j = 1;j <= m;j++){  cin>>a[i][j];     }
	}
	for(i = 1;i <= n;i++)    // 输出地图
	{
		for(j = 1;j <= m;j++)
		{
			if(a[i][j] == '*')       cout<<a[i][j];
			else
			{
				c = 0;       // 计算 i,j 这个单元格的 8 方向中有几个地雷
				if(a[i-1][j-1]=='*')     c++;
				if(a[i-1][j]=='*')       c++;
				if(a[i-1][j+1]=='*')     c++;
				if(a[i][j+1]=='*')       c++;
				if(a[i+1][j+1]=='*')     c++;
				if(a[i+1][j]=='*')       c++;
				if(a[i+1][j-1]=='*')     c++;
				if(a[i][j-1]=='*')       c++;
				cout<<c;
			}
		}
		cout<<endl;// 第 i 行打印结束，输出换行符
	}
	return 0;
}
```

```cpp
#include <iostream>//11-1580-2  anselxu  仅供参考
using namespace std;
int a[105][105];// 存储输出
int main()
{
	int n,m;
	char c;
	cin>>n>>m;
	for(int i=1;i<=n;i++)
		for(int j=1;j<=m;j++)
		{
			cin>>c;// 读入符号
			if(c=='*')// 如果读入的是地雷，进行赋值操作
			{	// 地雷的位置赋值一个大于 8 的数，相邻位置 --> 累加
				a[i][j]=9;
				a[i-1][j]++;
				a[i-1][j+1]++;
				a[i-1][j-1]++;
				a[i][j-1]++;
				a[i][j+1]++;
				a[i+1][j]++;
				a[i+1][j+1]++;
				a[i+1][j-1]++;
			}
		}
	for(int i=1;i<=n;i++)
	{
		for(int j=1;j<=m;j++)
			if(a[i][j]<=8)	cout<<a[i][j];// 输出时判断是否有效地雷数
			else  cout<<'*';// 无效地雷数输出地雷符号
			cout<<endl;
	}
	return 0;
}
```

扫雷（mine）

根据题目可以知道每个格子相邻的地雷数最多 8 个，所以可以利用此特征，将有地雷的方格赋值一个大于 8 的整数，输出时判断，小于等于 8 的存储单元才是有效的地雷数，否则输出地雷符号。 读入地图时某位置如果是地雷则该位置赋值为 9，对该点相邻的 8 个点累加 1，有‘*’号的点本身赋值大于 8，累加后也大于 8，对输出无影响。 边界问题，数组从 f（1，1）开始，最小行列坐标点的上面和左面有行下标 [0] 和列下标 [0]，不会越界。

每个小组的最大年龄

```
#include<bits/stdc++.h>//12-1996  wupeng
using namespace std;
int main()
{
	int a[110][110],r[110];
	int n,m,ma,x;

	cin >> n >> m;
	for(int i=1; i<=n; i++)
	{
		ma = 0;		// 每行年龄的最大值
		for(int j=1; j<=m; j++)
		{
			cin >> a[i][j];
			ma = max(ma, a[i][j]);
		}
		r[i] = ma;  // 存入结果数组
	}

	for(int i=1; i<=n; i++)
	{
		cout << r[i] << endl;
	}

	return 0;
}
```

```
#include<bits/stdc++.h>//13-1997-1  javacn  推荐学习
using namespace std;
bool isprime(int x) {
	if(x<=1)	return false;
	for(int i=2; i*i<=x; i++)
	{
		if(x%i==0)		return false;
	}
	return true;
}
int a[1100][1100];
int n,m,c;
int main() {
	cin>>n>>m;
	for(int i = 1; i<=n; i++) {
		for(int j = 1; j<=m; j++) {
			cin>>a[i][j];
		}
	}
	for(int i = 1; i<=n; i++)
	{
		for(int j = 1; j<=m; j++)
		{
			if(isprime(a[i][j])==true&&isprime(a[i-1][j])==false&&isprime(a[i][j-1])==false&&isprime(a[i+1][j])==false&&isprime(a[i][j+1])==false&&isprime(a[i-1][j-1])==false&&isprime(a[i+1][j-1])==false&&isprime(a[i-1][j+1])==false&&isprime(a[i+1][j+1])==false)
			{
				c++ ;
			}
		}
	}
	cout<<c;
	return 0;
}
```

孤独的素数

```
#include <iostream>//13-1997-2 leirui  仅供参考
using namespace std;
int main() {
	int prime[1005] = {0};
	for(int i=1;i<=1000;i++){
		int cnt = 0;
		for(int j=1;j<=i;j++){  if(i%j==0)		cnt++;		}
		if(cnt==2)  prime[i]++;
	}

	int a[55][55]={0};
	int n,m,cnt=0;
	cin>>n>>m;
	for(int i=1;i<=n;i++)
		for(int j=1;j<=m;j++)
			cin>>a[i][j];
	for(int i=1;i<=n;i++)
	{
		for(int j=1;j<=m;j++)
		{
			if(prime[a[i][j]]==1)// 是质数
			{		// 周围八个元素都不是质数
				if(prime[a[i-1][j-1]]==0&&prime[a[i-1][j]]==0  &&
prime[a[i-1][j+1]]==0  &&prime[a[i][j-1]]==0  &&  prime[a[i][j+1]]==0  &&
prime[a[i+1][j-1]]==0&&prime[a[i+1][j]]==0 && prime[a[i+1][j+1]]==0)
				{
					cnt++;
				}
			}
		}
	}
	cout<<cnt;
	return 0;
}
```

找朋友　思路：1. 小 T 的年龄是 a[t1][t2]2. 统计出第 x 行有多少个值是 a[t1][t2]，第 y 列有多少个值是 a[t1][t2]，答案 = 统计出来的总数 - 2（包括了 2 次自己）

```
# include<bits/stdc++.h>//14-1998  javacn
using namespace std;
int a[210][210];
int n,m,t1,t2,c = 0;
int main()
{
	cin>>n>>m;  // 读入数组
	for(int i = 1;i <= n;i++)
	{
		for(int j = 1;j <= m;j++)
		{
			cin>>a[i][j];
		}
	}
	cin>>t1>>t2;	// 读入小 t 的位置
	for(int j = 1;j <= m;j++)// 为了好理解，我们用 j 来循环列
	{
		if(a[t1][j] == a[t1][t2])// 循环第 t1 行（本质上是循环 m 列的每个数）
		{
			c++;
		}
	}

	for(int i = 1;i <= n;i++)	// 循环第 t2 列（本质上是循环 n 行的每个数）
	{
		if(a[i][t2] == a[t1][t2])
		{
			c++;
		}
	}
	cout<<c - 2;
	return 0;
}
```

操场换位置，求数组最大、最小数下标，交换后输出：

```
#include <bits/stdc++.h>//15-1999  javacn
using namespace std;
int a[210][210];
int main() {
	int n,m,x1,y1,x2,y2;	//注意：y1不能定义到main外面会和系统函数同名
	cin>>n>>m;		//x1 y1：最大数下标 //x2 y2：最小数下标
	x1 = 1;// 设定初始值
	y1 = 1;
	x2 = 1;
	y2 = 1;
	for(int i = 1; i <= n; i++)
	{
		for(int j = 1; j <= m; j++)
		{
			cin>>a[i][j];
			if(a[i][j] > a[x1][y1])
			{	x1 = i;
				y1 = j;
			}
			if(a[i][j] < a[x2][y2])
			{	x2 = i;
				y2 = j;
			}
		}
	}
	swap(a[x1][y1],a[x2][y2]);	//交换
	for(int i = 1; i <= n; i++)	//输出
	{
		for(int j = 1; j <= m; j++)
			cout<<a[i][j]<<" ";
		cout<<endl;
	}
	return 0;
}
```

两个数相邻吗？

```
#include<bits/stdc++.h>//16-2000    475214719
using namespace std;
int main()
{
	int i,j,m,n,t1,t2,t1i,t1j,t2i,t2j,a[201][201];
	cin>>n>>m;
	for(i=1;i<=n;i++)
		for(j=1;j<=m;j++)
			cin>>a[i][j];

		cin>>t1>>t2;

		for(i=1;i<=n;i++)
			for(j=1;j<=m;j++)
			{
				if(a[i][j]==t1)
				{
					t1i=i;
					t1j=j;
				}
				if(a[i][j]==t2)
				{
					t2i=i;
					t2j=j;
				}
			}

		if((t1i-t2i==-1||t1i-t2i==1)&&(t1j-t2j==0)||(t1j-t2j==-1||t1j-t2j==1)&&(t1i-t2i==0))
			cout<<'Y'<<endl;
		else
			cout<<'N'<<endl;
	return 0;
}
```

二维数组图形

思路：观察要赋值的几个数下标，可得规律，对行下标和列下标相同的位置赋值 1

```
#include<bits/stdc++.h>//1-1190
using namespace std;

int a[20][20];
int main()
{
	int n,i,j;
	cin>>n;
	// 循环赋值元素的个数，赋值 n 个数
	// 该循环表示，循环赋值元素的个数
	for(i=0;i<n;i++)
	{
		a[i][i]=1;
	}

	// 输出
	// 循环行
	for(i=0;i<n;i++)
	{
		// 循环列
		for(j=0;j<n;j++)
		{
			cout<<setw(3)<<a[i][j];
			//setw 场宽，限制输出位置大小，排列整齐
		}
		// 第 i 行输出结束，输出换行
		cout<<endl;
	}

	return 0;
}
```

```
5
  1  0  0  0  0
  0  1  0  0  0
  0  0  1  0  0
  0  0  0  1  0
  0  0  0  0  1
```

思路：观察要赋值的数的下标，可得规律

```
5
  0   0   0   0   1
  0   0   0   1   0
  0   0   1   0   0
  0   1   0   0   0
  1   0   0   0   0
```

```
#include<bits/stdc++.h>//2-1191   javacn
using namespace std;

int a[20][20];
int main()
{
	int n,i,j;
	cin>>n;
	// 循环赋值元素的个数，赋值 n 个数
	// 该循环表示，循环赋值元素的个数
	for(i=0;i<n;i++)
	{
		a[i][n-1-i]=1;
	}
	// 输出
	// 循环行
	for(i=0;i<n;i++)
	{
		// 循环列
		for(j=0;j<n;j++)
		{
			cout<<setw(3)<<a[i][j];
		}
		// 第 i 行输出结束，输出换行
		cout<<endl;
	}
	return 0;
}
```

```
5
  1  2  3  4  5
  6  7  8  9 10
 11 12 13 14 15
 16 17 18 19 20
 21 22 23 24 25
```

```cpp
#include<bits/stdc++.h>//3-1184  javacn
using namespace std;

int a[20][20];
int main()
{
	int n,x=1,i,j;//x 是要为数组赋的值
	cin>>n;

	// 赋值 n 次，每次赋值 n 个数，每次赋值的数字是递增的
	//i 代表的是赋值的次数
	for(i=0;i<n;i++)
	{
		//j 代表的是第 i 次赋值的个数
		for(j=0;j<n;j++)
		{
			a[i][j] = x;
			x++;
		}
	}
	// 输出
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			cout<<setw(3)<<a[i][j];
		}
		cout<<endl;
	}
	return 0;
}
```

```cpp
#include<bits/stdc++.h>//4-1185  jiangyf70
using namespace std;
int a[20][20];
int main()
{
	int n, k = 1;
	cin >> n;
	for(int i = n - 1; i >= 0; i--)
	{
		for(int j = 0; j < n; j++)
		{
			a[i][j] = k++;// 相当于 a[i][j] = k,k++;
		}
	}

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

```
5
 21 22 23 24 25
 16 17 18 19 20
 11 12 13 14 15
  6  7  8  9 10
  1  2  3  4  5
```

```cpp
#include<bits/stdc++.h>//5-1186  jiangyf70
using namespace std;
int a[20][20];
int main()
{
	int n, k = 1;
	cin >> n;
	for(int j = 0; j < n; j++)
	{
		for(int i = 0; i < n; i++)
		{
			a[i][j] = k++;// 相当于 a[i][j] = k,k++;
		}
	}

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

```
5
  1  6 11 16 21
  2  7 12 17 22
  3  8 13 18 23
  4  9 14 19 24
  5 10 15 20 25
```

```cpp
#include<bits/stdc++.h>//6-1187  jiangyf70
using namespace std;
int a[20][20];
int main()
{
	int n, k = 1;
	cin >> n;
	for(int j = n - 1; j >= 0; j--)
	{
		for(int i = 0; i < n; i++)
		{
			a[i][j] = k++;
		}
	}

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

```
5
 21 16 11  6  1
 22 17 12  7  2
 23 18 13  8  3
 24 19 14  9  4
 25 20 15 10  5
```

```cpp
#include <bits/stdc++.h>//7-1188  jiangyf70
using namespace std;
int a[10][10];
int main()
{
	int n;
	cin >> n;
	int k = n * n;
	for(int i = 0; i < n; i++)
	{
		for(int j = 0; j < n; j++)
		{
			a[i][j] = k--;
		}

	}
	for(int i = 0; i < n; i++)
	{
		for(int j = 0; j < n; j++)
		{
			cout << setw(3) << a[i][j];
		}
		cout << endl;
	}

	return 0;
}
```

```
5
 25 24 23 22 21
 20 19 18 17 16
 15 14 13 12 11
 10  9  8  7  6
  5  4  3  2  1
```

```cpp
#include <bits/stdc++.h>//8-1189  jiangyf70
using namespace std;
int a[10][10];
int main()
{
	int n;
	cin >> n;
	int k = 1;
	for(int i = 0; i < n; i++)
	{
		for(int j = n - 1; j >= 0; j--)
		{
			a[i][j] = k++;
		}
	}

	for(int i = 0; i < n; i++)
	{
		for(int j = 0; j < n; j++)
		{
			cout << setw(3) << a[i][j];
		}
		cout << endl;
	}

	return 0;
}
```

```
5
  5  4  3  2  1
 10  9  8  7  6
 15 14 13 12 11
 20 19 18 17 16
 25 24 23 22 21
```

```
5
  1   2   3   4   5
  2   3   4   5   4
  3   4   5   4   3
  4   5   4   3   2
  5   4   3   2   1
```

```cpp
#include <bits/stdc++.h>//9-1193-1   javacn
using namespace std;

int main()
{
	int a[20][20] = {0},i,j,n;
	cin>>n;
	// 循环赋值的次数
	for(i = 0;i < n;i++)
	{
		// 第 i 次赋值的个数
		for(j = 0;j <= i;j++)
		{
			a[i-j][j] = i+1;// 左上角
			a[n-1-j][j+(n-1-i)] = i + 1;// 右下角
		}
	}

	// 输出
	for(i = 0;i < n;i++)
	{
		for(j = 0;j < n;j++)
		{
			cout<<setw(3)<<a[i][j];
		}
		cout<<endl;
	}
	return 0;
}
```

```cpp
#include <bits/stdc++.h>//9-1193-2  jiangyf70
using namespace std;
int a[10][10];
int main()
{
	int n;
	cin >> n;
	for(int i = 0; i < n; i++)
	{
		for(int j = 0; j <= i; j++)
		{
			a[i - j][j] = i + 1;
			a[n - 1 - i + j][n- 1 - j] = i + 1;  //围绕中心点对称填数
		}
	}

	for(int i = 0; i < n; i++)
	{
		for(int j = 0; j < n; j++)
		{
			cout << setw(3) << a[i][j];
		}
		cout << endl;
	}

	return 0;
}
```

```cpp
#include<iostream>//10-1192-1 liangls 推荐学习
#include<algorithm>
#include<iomanip>
using namespace std;
int main()
{
	int n,a[15][15],t=1;
	cin>>n;
	for(int i=1; i<=n; ++i)
	{
		for(int j=1; j<=n; ++j)
		{
			a[i][j]=i+j-1;
		}
	}

	for(int i=1; i<=n; ++i)
	{
		for(int j=1; j<=n; ++j)
		{
			cout<<setw(3)<<a[i][j];
		}
		cout<<endl;
	}
	return 0;
}
```

```
5
  1  2  3  4  5
  2  3  4  5  6
  3  4  5  6  7
  4  5  6  7  8
  5  6  7  8  9
```

每次开始数均与行数相等，因此，t 赋值为 i 即可

```
#include<iostream>//10-1192-2  liangls 推荐学习
#include<algorithm>
#include<iomanip>
using namespace std;
int main()
{
	int n,a[15][15],t=1;
	cin>>n;
	for(int i=1; i<=n; ++i)
	{
		t=i;
		for(int j=1; j<=n; ++j)
		{
			a[i][j]=t;
			t++;
		}
	}

	for(int i=1; i<=n; ++i)
	{
		for(int j=1; j<=n; ++j)
		{
			cout<<setw(3)<<a[i][j];
		}
		cout<<endl;
	}
	return 0;
}
```

本题关键点在于观察每行的不同，起点均为上一行起点+1，所以只需在每个内层循环结束后，将 t 的值赋值为 a[i][1]+1 即可

```
#include<iostream>//10-1192-3  liangls 推荐学习
#include<algorithm>
#include<iomanip>
using namespace std;
int main()
{
	int n,a[15][15],t=1;
	cin>>n;
	for(int i=1; i<=n; ++i)
	{
		for(int j=1; j<=n; ++j)
		{
			a[i][j]=t;
			t++;
		}
		t=a[i][1]+1;
	}

	for(int i=1; i<=n; ++i)
	{
		for(int j=1; j<=n; ++j)
		{
			cout<<setw(3)<<a[i][j];
		}
		cout<<endl;
	}
	return 0;
}
```

```cpp
#include <bits/stdc++.h>//10-1192-4  jiangyf70 仅供参考
using namespace std;
int a[10][10];
int main()
{
	int n;
	cin >> n;
	for(int i = 0; i < n; i++)
	{
		for(int j = 0; j <= i; j++)
		{
			a[i - j][j] = i + 1;
			a[n - 1 - i + j][n- 1 - j] = 2 * n - 1 - i;

		}
	}

	for(int i = 0; i < n; i++)
	{
		for(int j = 0; j < n; j++)
		{
			cout << setw(3) << a[i][j];
		}
		cout << endl;
	}

	return 0;
}
```

```
5
  1   1   1   1   1
  1   2   2   2   2
  1   2   3   3   3
  1   2   3   4   4
  1   2   3   4   5
```

```cpp
#include<bits/stdc++.h>//11-1196-1  remedy1314  推荐学习
using namespace std;
int a[1010][1010]; int n;
int main() {
	cin>>n;
	for(int i=1;i<=n;i++)
	{
		for(int j=1;j<i;j++)
			cout<<setw(3)<<j;
		for(int j=i;j<=n;j++)
			cout<<setw(3)<<i;
		cout<<endl;
	}
	return 0;
}
```

分上下两部分

```cpp
#include <bits/stdc++.h>//11-1196-2  ku_sunny 推荐学习
using namespace std;
int a[11][11],n;
int main(){
	cin >>n;
	for(int i=1;i<=n;i++)
	{
		for(int j=i;j<=n;j++)	{	a[i][j]=i;	}
		for(int j=1;j<i;j++)	{	a[i][j]=j;	}
	}
	for(int i=1;i<=n;i++)
	{
		for(int j=1;j<=n;j++){
			cout << "  "<< a[i][j];
		}
		cout << endl;
	}
	return 0;
}
```

```cpp
#include<bits/stdc++.h>/12-1197-1   tc_ljp6
using namespace std;
int main()
{
	int n;
	cin>>n;
	for(int i=1;i<=n;i++)
	{
		for(int j=1;j<=n;j++)
		{
			cout<<setw(3)<<n+1-max(i,j);
		}
		cout<<endl;
	}
	return 0;
}
```

```
5
  5  4  3  2  1
  4  4  3  2  1
  3  3  3  2  1
  2  2  2  2  1
  1  1  1  1  1
```

```cpp
#include <bits/stdc++.h>//12-1197-2 wupeng
using namespace std;
int main()
{
	int a[20][20]={0};
	int n;
	cin >> n;

	for(int i=1; i<=n; i++)
	{
		for(int j=1; j<=n-i+1; j++)
		{
//			cout << i << ' ' << n-i+1 << ' ' << j << endl;
			a[n-i+1][j] = i;
//			cout << i << ' ' << (n-i+1)-j+1 << ' ' << n-i+1 << endl;
			a[(n-i+1)-j+1][n-i+1] = i;
		}
	}

	for(int i=1; i<=n; i++)
	{
		for(int j=1; j<=n; j++)
		{
			cout << setw(3) << a[i][j];
		}
		cout << endl;
	}

	return 0;
}
```

```
4
    1    3    8   20
         2    5   12
              3    7
                   4
```

```cpp
#include <bits/stdc++.h>//13-1205  jessechiu
using namespace std;
int data[120][120];
int main()
{
	int n,temp;
	cin>>n;
	for(int i=0;i<n;i++)
	{
		for(int j=0;j<n;j++)
		{
			if(i==j)
				data[i][j] = i + 1;
		}
	}
	for(int i=0;i<n-1;i++)
	{
		for(int j=0;j<n-1-i;j++)
		{
			data[j][j+1+i] = data[j][j+i] + data[j+1][j+1+i];
		}
	}
	for(int i=0;i<n;i++)
	{
		for(int j=0;j<n;j++)
		{
			if(data[i][j] > 0)
				printf("%5d",data[i][j]);
			else
				printf("%5s","");
		}
		printf("\n");
	}
	return 0;
}
```

```
5
  5  4  3  2  1
  4  5  4  3  2
  3  4  5  4  3
  2  3  4  5  4
  1  2  3  4  5
```

```cpp
#include<bits/stdc++.h>//14-1195-1  tc_ljp6
using namespace std;
int main(){
	int n;
	cin>>n;
	for(int i=1;i<=n;i++){
		for(int j=1;j<=n;j++){
			cout<<setw(3)<<n-abs(i-j);
		}
		cout<<endl;
	}
	return 0;
}
#include <bits/stdc++.h>//14-1195-2  hyb
using namespace std;
int a[20][20],n;
int main(){
	int i,j;
	cin>>n;
	for(i=1; i<=n; i++)
	{
		for(j=1; j<=n-i+1; j++)
		{
			a[i][n-j+1] = i+j-1;
			a[n-j+1][i] = i+j-1;
			//cout<<i<<" "<<n-j+1<<" "<<i+j-1<<endl;
		}
	}
	for(i=1; i<=n;i++)
	{
		for(j=1; j<=n; j++)	{	cout<<setw(3)<<a[i][j]; }
		cout<<endl;
	}
	return 0;
}
```

```
5
  A  B  C  D  E
  B  C  D  E  A
  C  D  E  A  B
  D  E  A  B  C
  E  A  B  C  D
```

```cpp
#include<iostream>//15-1194-1  anselxu
#include<iomanip>
using namespace std;
int main(){
	int n;
	char a='A'-1;
	cin>>n;
	for(int i=1;i<=n;i++){
		for(int j=1;j<=n;j++)
		{
			if(i+j-1<=n)	cout<<setw(3)<<char(a+i+j-1);
			else		cout<<setw(3)<<char(a+(i+j-1)-n);
		}
		cout<<endl;
	}
	return 0;
}
```

```cpp
#include<bits/stdc++.h>//15-1194-2  remedy1314
using namespace std;
int a[1010][1010];
int n; int main() {
	cin>>n;
	int x=0;
	for(int i=1;i<=n;i++)
	{
		int j;
		char k='A'+i-1;
		for(j=1;i+j-1<=n;j++)	cout<<setw(3)<<k++;
		k='A';
		for(;j<=n;j++)	cout<<setw(3)<<k++;
		cout<<endl;
	}
	return 0;
}
```

```cpp
#include <bits/stdc++.h>//15-1194-3  wupeng
using namespace std;
int main()
{
	char a[20][20];
	int n,x=1;
	cin >> n;
	// 赋值 n 次
	// i：代表赋值的次数
	for (int i=1; i<=n; i++)
	{
		for(int j=1; j<=i; j++)
		{	// 第 i 次要赋值 n 个数，j 代表第 i 次赋值元素的个数
			a[i-j+1][j] = i+'A'-1;
		}
	}

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

	for(int i=1; i<=n; i++)		// 输出数组
	{
		for (int j=1; j<=n; j++)
		{
			cout << setw(3) << a[i][j];
		}
		cout << endl;
	}
	return 0;
}
```

```cpp
#include <bits/stdc++.h>//15-1194-4  hyb
using namespace std;
int n;
char a[20][20];
int main(){
	int i,j;
	cin>>n;
	for(i=1;i<=n; i++){
		for(j=1; j<=i; j++){
			a[j][i-(j-1)] = 'A'+i-1;
		}
	}
	for(i=2; i<=n; i++){
		for(j=2; j<=n-i+2; j++){
			a[i+(j-2)][n-j+2] =  ‘A’ +(i-2);
		}
	}
	for(i=1; i<=n; i++){
		for(j=1; j<=n; j++){
			cout<<setw(3)<<a[i][j];
		}
		cout<<endl;
	}
	return 0;
}
```

```cpp
#include <bits/stdc++.h>//16-1198  hyb
using namespace std;
int a[20][20],n;
int main(){
	cin>>n;
	int j,i;
	for(i=1; i<=n; i++)
	{
		for(j=1; j<=i; j++)
		{
			a[n-i+1][n-i+1+j-1] = i;
			a[n-i+1+j-1][n-i+1] = i;
		}
	}

	for(i=1; i<=n; i++)
	{
		for(j=1; j<=n; j++)
		{
			cout<<setw(3)<<a[i][j];
		}
		cout<<endl;
	}
	return 0;
}
```

```
5
  5  5  5  5  5
  5  4  4  4  4
  5  4  3  3  3
  5  4  3  2  2
  5  4  3  2  1
```

```
5
  1   2   3   4   5
  2   2   3   4   5
  3   3   3   4   5
  4   4   4   4   5
  5   5   5   5   5
```

```cpp
#include<bits/stdc++.h>//17-1199-1  tc_ljp6
using namespace std;
int main(){
	int n;
	cin>>n;
	//规律  行数和列数两个数字中取最大值
	for(int i=1;i<=n;i++){
		for(int j=1;j<=n;j++){
			cout<<setw(3)<<max(i,j);
		}
		cout<<endl;
	}
	return 0;
}
#include <bits/stdc++.h>//17-1199-2  hyb
using namespace std;
int a[20][20],n;
int main(){
	cin>>n;
	int i,j;
	for(i=1; i<=n; i++){
		for(j=1; j<=n-i+1; j++){
			a[i][i+j-1] = i+j-1;
			a[i+j-1][i] = i+j-1;
		}
	}

	for(i=1; i<=n; i++){
		for(j=1; j<=n; j++){
			cout<<setw(3)<<a[i][j];
		}
		cout<<endl;
	}
	return 0;
}
```

依次赋值：

第 1 行的 1，第 n 行的 1

第 2 行的 2，第 n-1 行的 2

……

```
5
  1   1   1   1   1
  2   2   2   2   1
  3   3   3   2   1
  4   4   3   2   1
  5   4   3   2   1
```

```
#include <bits/stdc++.h>//18-1200-1 javacn
using namespace std;
int main()
{
	int a[10][10] = {0},n;
	cin>>n;
	// 赋值次数
	for(int i = 0;i < n;i++)
	{
	  // 第 i 次赋值 n-i 个数
		for(int j = 0;j < n - i;j++)
		{
			a[i][j] = i + 1;// 横向赋值
			a[j + i][n - i - 1] = i + 1;// 纵向辅助
		}
	}

	for(int i = 0;i < n;i++)
	{
		for(int j = 0;j < n;j++)
		{
			cout<<setw(3)<<a[i][j];
		}
		cout<<endl;
	}

	return 0;
}
```

```cpp
#include <bits/stdc++.h>//18-1200-2 fidel2021
using namespace std;
int n, a[20][20];
int main()
{
	cin >> n;
	for(int i=1; i<=n; ++i)
	{
		a[i][n+1-i]=i;
		for(int j=1; j<=n-i; ++j)
		{
			a[i][j]=a[i+j][n+1-i]=a[i][n+1-i];
		}
	}

	for(int i=1; i<=n; ++i)
	{
		for(int j=1; j<=n; ++j)
		{
			cout << setw(3) << a[i][j];
		}
		cout << endl;
	}
	return 0;
}
```

```cpp
#include <bits/stdc++.h>//19-1201  fidel2021
using namespace std;
int n, a[20][20];
int main()
{
	cin >> n;
	for(int i=1; i<=n; ++i)
	{
		a[n+1-i][i]=i;
		for(int j=i+1; j<=n; ++j)
		{
			a[n+1-i][j]=a[j-i][i]=i;
		}
	}

	for(int i=1; i<=n; ++i)
	{
		for(int j=1; j<=n; ++j)
		{
			cout << setw(3) << a[i][j];
		}
		cout << endl;
	}
	return 0;
}
```

```
5
  1  2  3  4  5
  1  2  3  4  4
  1  2  3  3  3
  1  2  2  2  2
  1  1  1  1  1
```

```cpp
#include <bits/stdc++.h>//20-1202  fidel2021
using namespace std;
int n, a[20][20];
int main()
{
	cin >> n;
	for(int i=1; i<=n; ++i)
	{
		//第一列单独处理
		a[i][1]=n+1-i;
		for(int j=2; j<=n; ++j)
		{
			if(j<=n+1-i)
			{
				a[i][j]=n+1-i;
			}
			else
			{
				a[i][j]=a[i][j-1]+1;
			}
		}
	}

	for(int i=1; i<=n; ++i)
	{
		for(int j=1; j<=n; ++j)
		{
			cout << setw(3) <<a[i][j];
		}
		cout << endl;
	}
	return 0;
}
```

```
5
  5  5  5  5  5
  4  4  4  4  5
  3  3  3  4  5
  2  2  3  4  5
  1  2  3  4  5
```

```cpp
#include <bits/stdc++.h>//21-1203    fidel2021
using namespace std;
int n, a[20][20];
int main()
{
	cin >> n;
	for(int i=1; i<=n; ++i)
	{
		// 第 n 列单独处理
		a[i][n]=i;
		for(int j=n-1; j>=1; --j)
		{
			if(n+1-j<=i)
			{
				a[i][j]=i;
			}
			else
			{
				a[i][j]=a[i][j+1]+1;
			}
		}
	}

	for(int i=1; i<=n; ++i)
	{
		for(int j=1; j<=n; ++j)
		{
			cout << setw(3) << a[i][j];
		}
		cout << endl;
	}
	return 0;
}
```

```
5
  5  4  3  2  1
  5  4  3  2  2
  5  4  3  3  3
  5  4  4  4  4
  5  5  5  5  5
```

```
#include <bits/stdc++.h>//22-1204  zhangqingbin125
using namespace std;
int kong(int z)
{
	for(int i=1;i<=z;i++)
	{
		cout<<setw(5)<<" ";
	}
	return 0;
}

int main()
{
	int n;
	cin>>n;

	for(int i=1;i<=n;i++)
	{
		for(int j=1;j<=n-i+1;j++)
		{
			cout<<setw(5)<<j;
		}
		cout<<endl;
		kong(i);
	}

	return 0;
}
```

```
4
    1    2    3    4
         1    2    3
              1    2
                   1
```

```cpp
#include <bits/stdc++.h>//23-1206    wupeng
using namespace std;
int main()
{
	int a[20][20]={0};
	int n;
	cin >> n;

	// 赋值次数
	for (int i=1; i<=n; i++)
	{
		// j 代表第 i 次，赋值元素的个数
		for(int j=1; j<=n-i+1; j++)
		{
			a[j+i-1][n-j+1] = i;
		}
	}
	// 输出数组
	for(int i=1; i<=n; i++)
	{
		for(int j=1; j<=n; j++)
		{
			if(a[i][j]!= 0)
			{
				cout << setw(5) << a[i][j];
			}
			else
			{
				cout << setw(5) << ' ';
			}
		}
		cout << endl;
	}
	return 0;
}
```

```
4
                    1
               1    2
          1    2    3
     1    2    3    4
```

```
#include<bits/stdc++.h>//24-1207  fidel2021
using namespace std;
int n,a[20][20];
int main()
{
	cin>>n;
	for(int i=1;i<=n;i++)
	{
		for(int j=1;j<=n;j++)
		{
			if(i+j<n+1)
				cout<<setw(5)<<' ';
			else
			{
				if(j!=n-i+1)
				{
					a[i][j]=a[i-1][j]+a[i][j-1];
				}
				if(j==n-i+1)
				{
					a[i][j]=n-i+1;
				}
				cout<<setw(5)<<a[i][j];
			}
		}
		cout<<endl;
	}
	return 0;
}
```

```
4
                   4
              3    7
         2    5   12
    1    3    8   20
```

有趣的数字图形

```
4
   1    3    8   20
   3    2    5   12
   8    5    3    7
  20   12    7    4
```

```cpp
#include <bits/stdc++.h>//25-1385  jiangyf70
using namespace std;
int a[20][20];
int main(){
	int i = 0, j = 0, k = 0, n;
	cin >> n;

	for(int i = 0; i < n; i++)
	{
		a[i][i] = ++k;
	}
	k = 1;
	while(k < n)
	{
		i = 0;
		j = i + k;
		while(i < n && j < n)
		{
			a[i][j] = a[i][j-1] + a[i+1][j];
			a[j][i] = a[i][j];
			i++;
			j++;
		}
		k++;
	}

	for(int i = 0; i < n; i++)
	{
		for(int j = 0; j < n; j++)
			cout << setw(5) << a[i][j];
		cout << endl;
	}
	return 0;
}
```

```cpp
#include <bits/stdc++.h>//26-1327    jiangyf70
using namespace std;
int a[20][20];
int main()
{
	int n;
	cin >> n;
	int x = n / 2;

	for(int i = 0; i < n; i++)
	{
		for(int j = 0; j < n; j++)
		{
			if(!(abs(i - x) + abs(j - x) == x))
				a[i][j] = 1;
				// 曼哈顿距离坐标（i, j）到（x, x）的距离
		}
	}
	a[x][x] = 0;
	for(int i = 0; i < n; i++)
	{
		for(int j = 0; j < n; j++)
		{
			cout << setw(3) << a[i][j];
		}
		cout << endl;
	}
	return 0;
}
```

```
5
1  1  0  1  1
1  0  1  0  1
0  1  0  1  0
1  0  1  0  1
1  1  0  1  1
```