

太戈编程
etiger.vip

信奥算法

二维数组

名次	国家/地区				总计
1	 美国	28	30	28	86
2	 英国	19	19	12	50
3	 中国	17	15	20	52
4	 俄罗斯	12	12	15	39
5	 德国	11	8	9	28
6	 法国	8	11	12	31
7	 意大利	8	9	6	23
8	 日本	8	4	18	30
9	 荷兰	8	3	3	14
10	 澳大利亚	7	8	9	24



如何储存奖牌榜呢？

储存前3名的金牌数量






0号列

1号列

0号行

1号行

2号行

国家/地区			
 美国		28	30
 英国		19	19
 中国		17	15

数组编号从0开始
不是从1开始

储存前3名的金银牌数量

```
1 #include<iostream>
2 using namespace std;
3 int f[3][2];
4 int main(){
5     f[0][0]=28; f[0][1]=30;
6     f[1][0]=19; f[1][1]=19;
7     f[2][0]=17; f[2][1]=15;
8     return 0;
9 }
```

二维数组f
包含3个一维数组

一维数组f[0]
一维数组f[1]
一维数组f[2]

二维数组定义

行数 列数

int f[3][2];

定义二维数组f
包含3行2列个
整数类型变量

f[0][0]	f[0][1]
f[1][0]	f[1][1]
f[2][0]	f[2][1]

读作f 2 0

读作f 2 1

← 一维数组f[0]

← 一维数组f[1]

← 一维数组f[2]

二维数组初始化

```
int a[2][3] = {{1, 2, 3}, {4, 5, 6}};
```

```
int b[2][3] = {1, 2, 3, 4, 5};
```

```
int c[2][3] = {{1, 2}, {4}};
```

a =

1	2	3
4	5	6

b =

1	2	3
4	5	0

c =

1	2	0
4	0	0

初始化的个数小于元素个数时
用默认值补齐

记
笔
记







实用技巧

高频错误

全局数组自动清零

局部变量忘记初始化

使用3行4列的二维整型数组**int medals[3][4]**
按照奖牌榜初始化，并打印输出

国家/地区				总计
 美国	28	30	28	86
 英国	19	19	12	50
 中国	17	15	20	52

medals[i][j]代表i号国家j号奖牌类型的数量

举例：medals[0][0]代表美国金牌数量

举例：medals[0][1]代表美国银牌数量

举例：medals[2][0]代表中国金牌数量

举例：medals[2][3]代表中国奖牌总数

二维数组输出

```
1  #include<iostream>
2  using namespace std;
3  int medals[3][4]={
4      {28,30,28,86},
5      {19,19,12,50},
6      {17,15,20,52}
7  };
8  int main(){
9      for(int i=0;i<3;++i){
10         for(int j=0;j<4;++j)
11             cout<<medals[i][j]<<" ";
12         cout<<endl;
13     }
14     return 0;
15 }
```

写在**int main()**
上方的是全局数组

枚举3个国家编号*i*从0到2

枚举4种奖牌号*j*从0到3

共输出换行几次?

易错点

```
1  #include<iostream>
2  using namespace std;
3  int medals[3][4]={
4      {28,30,28,86},
5      {19,19,12,50},
6      {17,15,20,52}
7  };
8  int main(){
9      for(int i=1;i<=3;++i){
10         for(int j=1;j<=4;++j)
11             cout<<medals[i][j]<<" ";
12         cout<<endl;
13     }
14     return 0;
15 }
```

高频错误	记笔记
数组编号越界 out of bounds	

奖牌统计







使用3行4列的二维整型数组**int medals[3][4]**

按照奖牌榜初始化，输出三行：

输出这三国共获得多少金牌

输出这三国共获得多少银牌

输出这三国共获得多少铜牌

国家/地区				总计
 美国	28	30	28	86
 英国	19	19	12	50
 中国	17	15	20	52

奖牌统计

```
1  /*
2  定义二维数组medals包含3行4列个整数类型变量
3  medals[i][j]代表i号国家j号奖牌类型的数量
4  */
5  #include<iostream>
6  using namespace std;
7  int medals[3][4]={
8      {28,30,28,86},
9      {19,19,12,50},
10     {17,15,20,52}
11 };
12 int main(){
13     cout<<medals[0][0]+medals[1][0]+medals[2][0]<<endl;
14     cout<<medals[0][1]+medals[1][1]+medals[2][1]<<endl;
15     cout<< <<endl;
16     return 0;
17 }
```

二维数组输入1

输入3行4列共12个整数，请储存在数组中，再输出

```
1  #include<iostream>
2  using namespace std;
3  int f[3][4]; ← 数组定义
4  int main(){
5      for(int i=0;i<3;i++)
6          for(int j=0;j<4;j++)
7              cin>>f[i][j];
8      for(int i=0;i<3;i++){
9          for (int j=0;j<4;j++)
10             cout<<f[i][j]<<" ";
11             cout<<endl;
12     }
13     return 0;
14 }
```

数组定义

两重循环
输入

两重循环
输出

二维数组输入2

输入第一行为n和m代表有n行m列数据。 $n, m \leq 10$ 。
之后共n行，每行m个字符。
请将输入的数据储存在数组中，并打印出来。

输入样例：

```
2 3
ABC
DEF
```

输出样例：

```
ABC
DEF
```

输入样例：

```
5 4
QWER
ASDF
ZXCV
TYUI
GHJK
```

输出样例：

```
QWER
ASDF
ZXCV
TYUI
GHJK
```



```
1  #include<iostream>
2  using namespace std;
3  char d[10][10];
4  int main(){
5      int n,m;
6      cin>>n>>m;
7      for(int i=0;i<n;i++)
8          for(int j=0;j<m;j++)
9              cin>>d[i][j];
10     for(int i=0;i<n;i++){
11         for(int j=0;j<m;j++)
12             cout<<d[i][j];
13         cout<<endl;
14     }
15     return 0;
16 }
```

← d[i][j]代表棋盘i号行j号列的字符

两重循环
输入

两重循环
输出

二维数组小结

定义

```
int f[109][109];
```

```
char d[10][10];
```

使用固定值定义数组大小

输出

使用两重循环
输出数组每个元素d[i][j]

循环变量i控制行
循环变量j控制列

输入

使用两重循环
输入数组每个元素d[i][j]

循环变量i控制行
循环变量j控制列

找棋子

在3*3的棋盘上有几颗棋子，请输出他们的位置。'0'代表没有棋子，'W'代表白棋子，'B'代表黑棋子。

输入样例：

000

0W0

B00

输出样例：

1 1 W

2 0 B

输入样例：

0B0

0B0

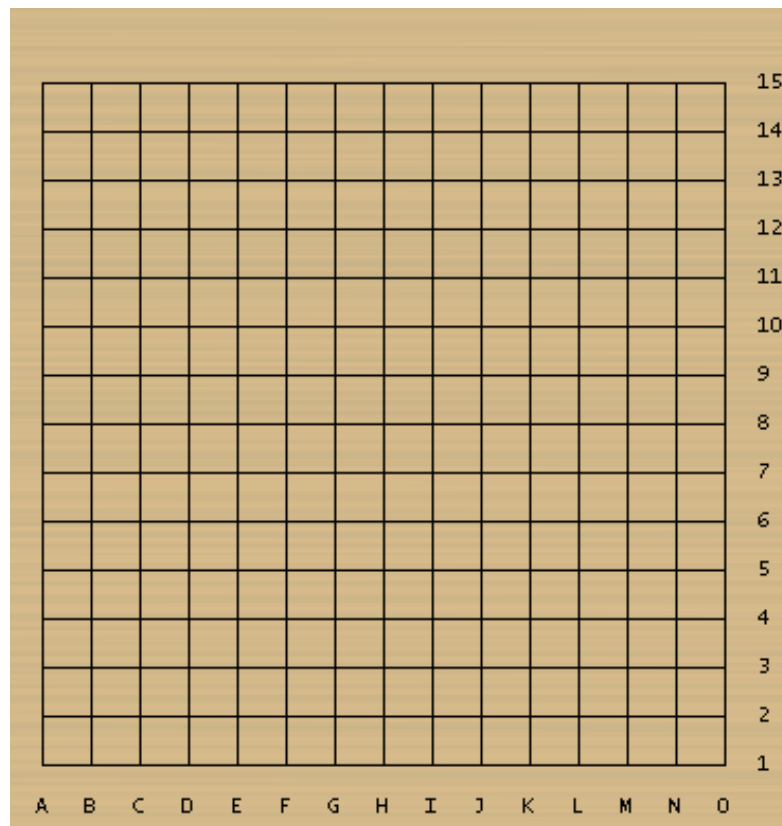
0B0

输出样例：

0 1 B

1 1 B

2 1 B



找棋子

```
1 #include<iostream>
2 using namespace std;
3 char p[3][3];
4 int main(){
5     for(int i=0;i<=2;i++)
6         for(int j=0;j<=2;j++)
7             cin>>p[i][j];
8
9     for(int i=0;i<=2;i++)
10        for(int j=0;j<=2;j++)
11            if(p[i][j]=='W' || p[i][j]=='B')
12                cout<<i<<' '<<j<<' '<<p[i][j]<<endl;
13 }
```

p[i][j]代表棋盘i号行j号列的字符

枚举3行,行号i从0到2

枚举3列,列号j从0到2

请同学翻译第10行

杨辉三角形

输入一个正整数n，打印出一个n行的杨辉三角形。n<10

输入样例：
4

输出样例：
1
1 1
1 2 1
1 3 3 1

输入样例：
6

输出样例：
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

定义二维数组
yh[10][10]

yh[i][j] 代表
杨辉三角形第i行第j列

杨辉三角形

输入一个正整数 n ，打印出一个 n 行的杨辉三角形。 $n < 10$

	$j=0$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	
$i=0$	0	0	0	0	0	0	缓冲带
$i=1$	0	1	0	0	0	0	
$i=2$	0	1	1	0	0	0	
$i=3$	0	1	2	1	0	0	
$i=4$	0	1	3	3	1	0	
$i=5$	0	1	4	6	4	1	

每格都依赖
正上方
和左上方

缓冲带

若没有缓冲带会怎样?


```
1 #include<iostream>
2 using namespace std;
3 int n,yh[10][10];
4 int main(){
5     cin>>n;
6     yh[1][1]=1;
7     for(int i=2;i<=n;i++)
8         for(int j=1;j<=i;j++)
9             yh[i][j]=yh[i-1][j-1]+yh[i-1][j];
```

yh[i][j] 代表
杨辉三角形第i行第j列

全局数组自动清零

枚举n-1行,行号i从2到n

枚举i列,列号j从1到i

请同学翻译第9行

```

1  #include<iostream>
2  using namespace std;
3  int n,yh[10][10]; ←
4  int main(){
5      cin>>n;
6      yh[1][1]=1;
7      for(int i=2;i<=n;i++)
8          for(int j=1;j<=i;j++)
9              yh[i][j]=yh[i-1][j-1]+yh[i-1][j];
10     for(int i=1;i<=n;i++){
11         for(int j=1;j<=i;j++)
12             cout<<yh[i][j]<<" ";
13         cout<<endl;
14     }
15     return 0;
16 }

```

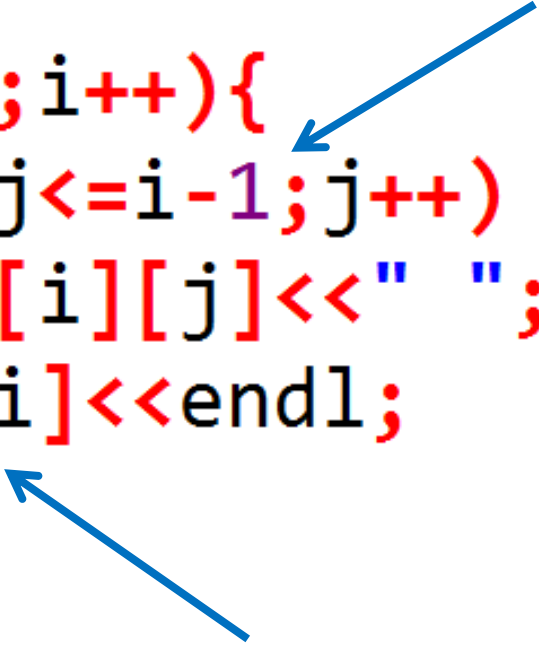
yh[i][j] 代表
杨辉三角形第i行第j列

全局数组自动清零

行末会有空格

若题目要求
行末没有空格
如何修改?

```
for(int i=1;i<=n;i++){  
    for(int j=1;j<=i-1;j++)  
        cout<<yh[i][j]<<" ";  
    cout<<yh[i][i]<<endl;  
}
```



枚举列号时
故意少枚举
最后一列

每行最后的元素
输出后直接换行

若题目要求
行末没有空格
如何修改?

眼观四路

小明能眼观四路。输入一个5*5的棋盘， '*'表示他在棋盘中的位置， 输出一个棋盘， '@'表示他能观察到的格子。

输入样例：

```
00000
00000
00*00
00000
00000
```

输出样例：

```
00000
00@00
0@*@0
00@00
00000
```

输入样例：

```
00000
00000
00000
0000*
00000
```

输出样例：

```
00000
00000
0000@
000@*
0000@
```

棋盘格四方向

	$(i-1, j)$	
$(i, j-1)$	(i, j)	$(i, j+1)$
	$(i+1, j)$	

i 代表行号
 j 代表列号

眼观四路

```
3 char a[7][7];  
4 int main(){  
5     for(int i=1;i<=5;i++)  
6         for(int j=1;j<=5;j++)  
7             cin>>a[i][j];
```

← a[i][j]代表棋盘i号行j号列的字符

使用1号到5号
不使用0号和6号

眼观四路

```
3 char a[7][7];
4 int main(){
5     for(int i=1;i<=5;i++)
6         for(int j=1;j<=5;j++)
7             cin>>a[i][j];
```

$a[i][j]$ 代表棋盘i号行j号列的字符

使用1号到5号
不使用0号和6号

```
8     for(int i=1;i<=5;i++)
9         for(int j=1;j<=5;j++)
10            if(a[i][j]=='*'){
```

行号i减1代表往上

行号i加1代表往下

列号j减1代表往左

列号j加1代表往右

```
    if(i-1>=1) a[i-1][j]='@';
```

```
    if(i+1<=5) a[i+1][j]='@';
```

```
    if(j-1>=1) a[i][j-1]='@';
```

```
    if(j+1<=5) a[i][j+1]='@';
```

```
    }
```

326

棋盘格八方向

$(i-1, j-1)$	$(i-1, j)$	$(i-1, j+1)$
$(i, j-1)$	(i, j)	$(i, j+1)$
$(i+1, j-1)$	$(i+1, j)$	$(i+1, j+1)$

i代表行号
j代表列号

```
1  /*
2  定义二维数组a 包含7行7列个char类型变量
3  a[i][j]代表棋盘i号行j号列的字符
4  */
5  #include<bits/stdc++.h>
6  using namespace std;
7  char a[7][7]; ←
8  int main(){
9      freopen("ear.in", "r", stdin);
10     freopen("ear.out", "w", stdout);
11     for(int i=1; i<=5; i++)
12         for(int j=1; j<=5; j++)
13             cin>>a[i][j];
```

使用1号到5号
不使用0号和6号

a[i][j]代表棋盘i号行j号列的字符

```
for(int i=1;i<=5;i++)  
    for(int j=1;j<=5;j++)  
        if(a[i][j]=='*'){
```

行号i减1代表往上

行号i加1代表往下

列号j减1代表往左

列号j加1代表往右

请识别
剩余的
4个方向

```
if(i-1>=1) a[i-1][j]='#';
```

```
if(i+1<=5) a[i+1][j]='#';
```

```
if(j-1>=1) a[i][j-1]='#';
```

```
if(j+1<=5) a[i][j+1]='#';
```

```
if(i-1>=1&& j-1>=1) a[i-1][j-1]='#';
```

```
if(i-1>=1&& j+1<=5)
```

```
if(i+1<=5&& j-1>=1)
```

```
if(i+1<=5&& j+1<=5)
```

```
}
```

二维数组 内存地址


```
1  #include<iostream>
2  using namespace std;
3  int f[3][2];
4  int main(){
5      cout<<&f[0][0]<<endl;
6      cout<<&f[0][1]<<endl;
7      cout<<&f[1][0]<<endl;
8      cout<<&f[1][1]<<endl;
9      cout<<&f[2][0]<<endl;
10     cout<<&f[2][1]<<endl;
11     return 0;
12 }
```

观察6个元素的地址

```
1 #include<iostream>
2 using namespace std;
3 int f[3][2];
4 int main(){
5     cout<<f[0]<<endl;
6     cout<<f[1]<<endl;
7     cout<<f[2]<<endl;
8     cout<<f<<endl;
9     return 0;
10 }
```

f[0]不是元素值
而是内存地址

f[0]和f是
同一个内存地址

太戈编程

99

326

1438

拓展题

6