

Welcome

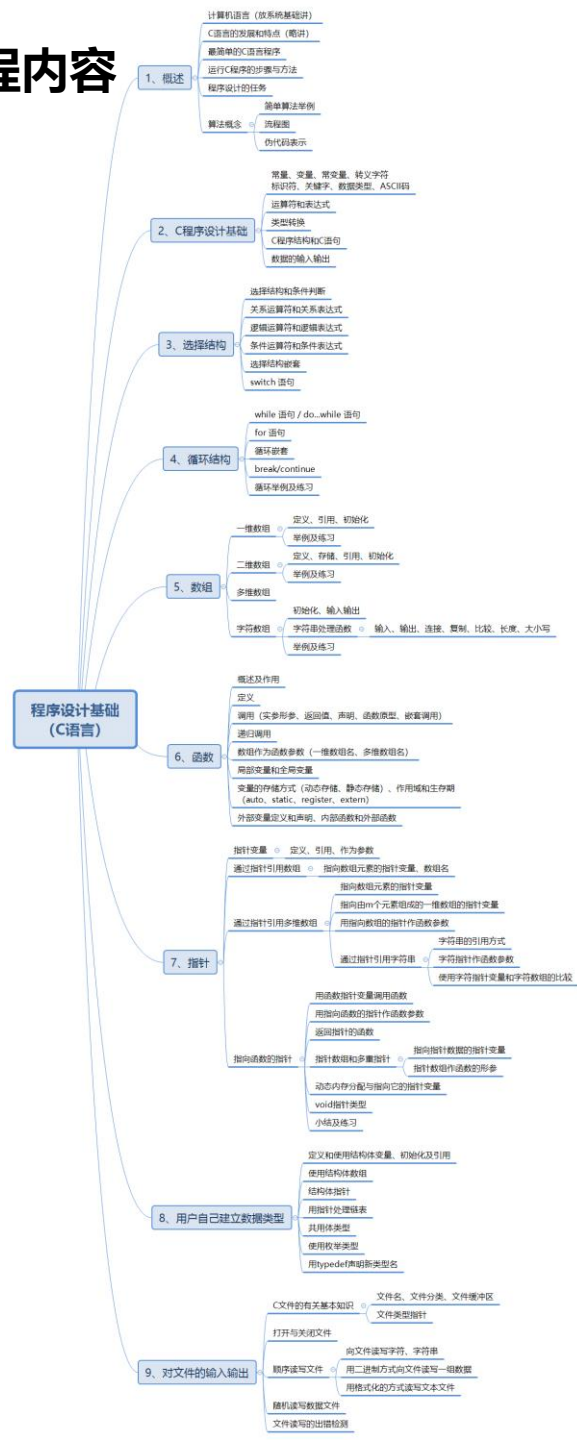
# 数据科学与大数据技术专业 程序设计基础(C语言)

上海体育学院经济管理学院

Wu Ying



# 课程内容



## 第5周 程序设计概述

### 计算机语言及C语言

#### 最简单的C语言程序

#### 运行C程序的步骤和方法

#### 程序设计的任务

#### 算法概念及简单算法

#### 算法表示

#### 流程图

#### 伪代码

## 第6周 C程序设计基础

#### 常量、变量及数据类型

#### 常量变量常量变量转义字符

#### 标识符关键字数据类型

#### 运算符和表达式及类型转换

#### C程序结构和C语句

#### 数据的输入输出

## 第7周 选择结构

#### 选择结构和条件判断

#### 运算符和表达式

#### 关系运算符和关系表达式

#### 逻辑运算符和逻辑表达式

#### 条件运算符和条件表达式

#### 选择结构嵌套

#### switch 语句

## 第8周 循环结构

#### while语句和do...while语句

#### for语句

#### 循环嵌套, break, continue

#### 循环练习

## 第9周 数组 (一)

#### 一维数组定义、引用、初始化

#### 一维数组练习

#### 二维数组定义引用初始化

#### 多维数组及练习

## 第10周 数组 (二) 及函数 (一)

#### 字符数组初始化及输入输出

#### 字符串处理函数

#### 函数定义与调用

#### 函数返回值、声明、原型、嵌套调用

## 第11周 函数 (二)

#### 递归调用

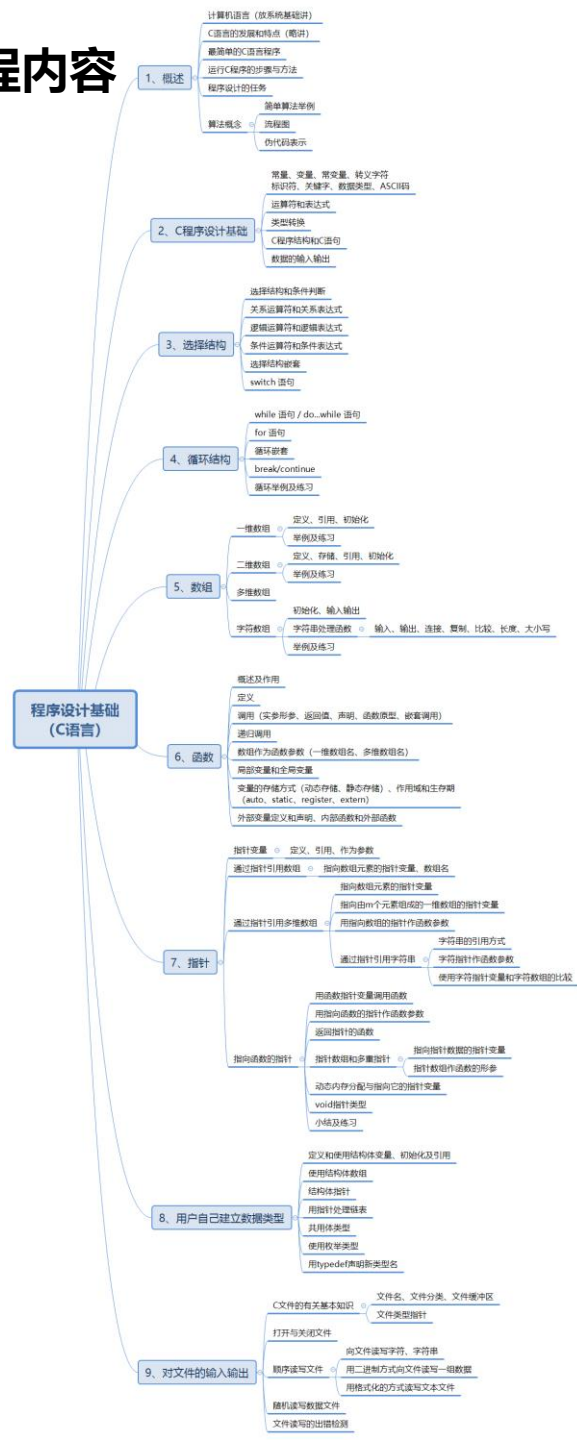
#### 数组作为函数参数

#### 变量作用域和生存期

#### 变量存储方式、内部函数与外部函数



# 课程内容



## 第12周 指针 (一)

指针变量

通过指针引用数组

### 通过指针引用多维数组 (I)

指向数组元素的指针变量

指向一维数组的指针变量

## 第13周 指针 (二)

### 通过指针引用多维数组 (II)

用指向数组的指针作为函数参数

### 通过指针引用字符串

字符串的引用方式

字符指针作函数参数

使用字符指针变量和字符串数组的比较

## 第14周 指针 (三)

### 指向函数的指针 (I)

用函数指针变量调用函数

用指向函数的指针作函数参数

返回指向指针的函数

### 指向函数的指针 (II)

#### 指针数组和多重指针

指向指针数据的指针变量

指针数组作函数的形参

动态内存分配与指向它的指针变量

## 第15周 指针 (四) 及结构体 (一)

### 指向函数的指针 (IV)

void指针类型

小结及练习

### 结构体

定义和使用结构体变量

使用结构体数组、结构体指针

## 第16周 结构体 (二) 及文件 (一)

用指针处理链表、共用体类型

枚举类型、使用typedef声明新类型

C文件的基本知识

打开与关闭文件

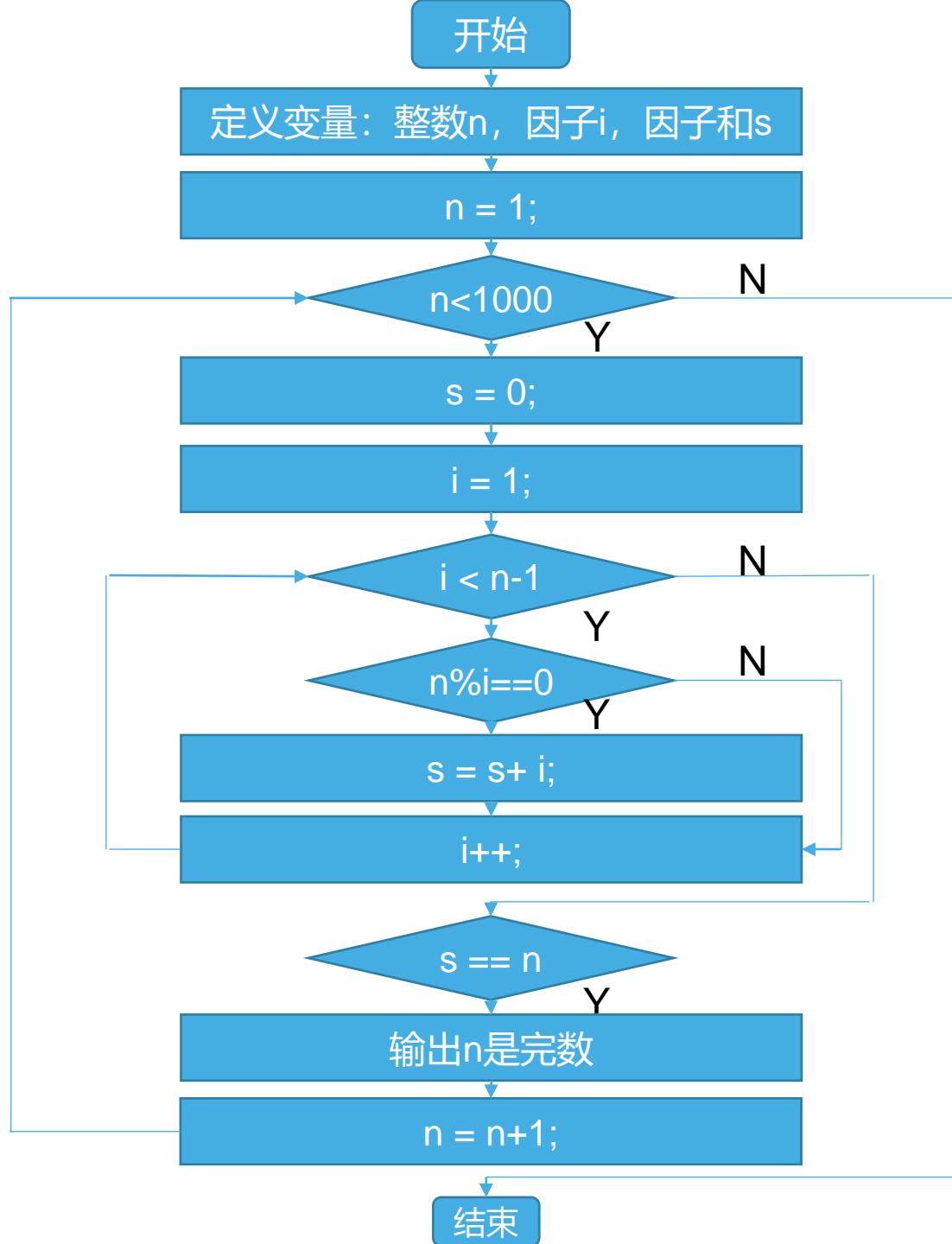
## 第17周 文件 (二)

顺序读写文件

随机读写数据文件

总复习练习

总复习练习



【9】一个数如果恰好等于它的因子之和,这个数就是“完数” 例如,6的因子是1,2,3,而 $6 = 1 + 2 + 3$ ,因此6是完数.请编写程序,求出1000内的所有完数.



**【6】  $1! + 2! + \dots + 20!$**

求阶乘和，数值很大，用double定义累加和

# 程序设计基础（C语言） 第9次



## 第6章利用数组处理批量数据

01

### 一维数组

- 定义、初始化以及引用
- 冒泡排序算法

02



### 二维数组

- 定义、初始化以及引用

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
10	6	8	33	24	115	900	7	80	95

# 定义和引用二维数组

## 例

有3个组，每组有6名队员，需把队员的工资按组保存备查。

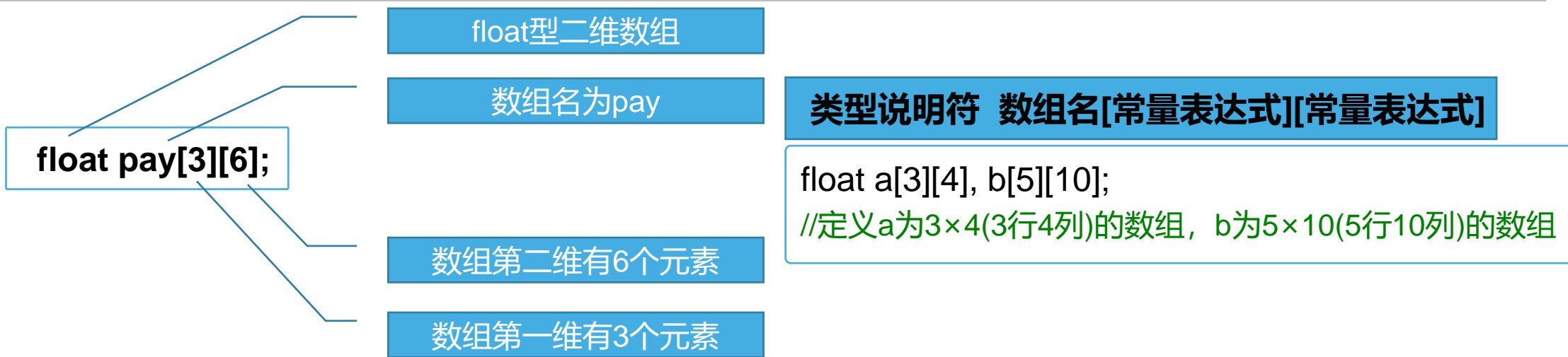
		队员1	队员2	队员3	队员4	队员5	队员6
		0	1	2	3	4	5
第1组	0	2456	1847	1243	1600	2346	2757
第2组	1	3045	2018	1725	2020	2458	1436
第3组	2	1427	1175	1046	1976	1477	2018

建立一个二维数组pay：第一维用来表示组别（3行 0...2）；第二维用来表示队员（6列 0...5）

例如用pay[2][3]表示第3组第4名队员的工资，它的值是1976

二维数组常称为**矩阵**(matrix)

# 定义二维数组



二维数组可被看作一种特殊的一维数组: 它的元素是一个一维数组。

```
float a[3][4];
```

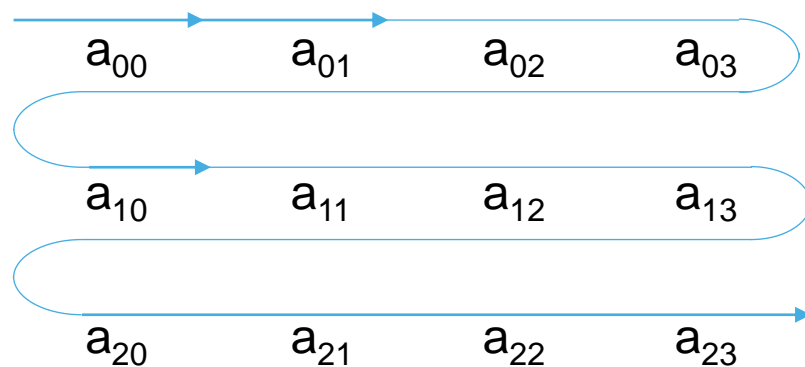
a[0]	<u>a</u> [0][0]	<u>a</u> [0][1]	<u>a</u> [0][2]	<u>a</u> [0][3]
a[1]	<u>a</u> [1][0]	<u>a</u> [1][1]	<u>a</u> [1][2]	<u>a</u> [1][3]
a[2]	<u>a</u> [2][0]	<u>a</u> [2][1]	<u>a</u> [2][2]	<u>a</u> [2][3]



# 二维数组的存储

C语言中，二维数组的元素在内存中是按行连续存放的，是线性的。

```
float a[3][4]
```



2000	a[0][0]	第0行元素
2004	a[0][1]	
2008	a[0][2]	
2012	a[0][3]	
2016	a[1][0]	第1行元素
2020	a[1][1]	
2024	a[1][2]	
2028	a[1][3]	
2032	a[2][0]	第2行元素
2036	a[2][1]	
2040	a[2][2]	
2044	a[2][3]	

# 引用二维数组元素

## 数组名[下标][下标]

“下标”可以是整型常量或整型表达式，如a[2][3]

数组元素可以出现在表达式中，也可以被赋值，如：b[1][2]=a[2][3]/2;

## 注意

- 在引用数组元素时，下标值应在已定义的数组大小的范围内。

```
int a[3][4]; //定义a为3×4的二维数组
```

```
a[3][4]=3; //?
```

- 定义数组时用的a[3][4]，定义数组的维数和各维的大小
- 引用元素时的a[3][4]，其中的3和4是数组元素的下标值，代表行号为3、列号为4的元素（序号从0起），超出数组定义范围，越界。

# 二维数组的初始化

(1)分行给二维数组赋初值。（最清楚直观）

```
int a[3][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12}};
```

(2)可以将所有数据写在一个花括号内，按数组元素在内存中的排列顺序对各元素赋初值。

```
int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};
```

(3)可以对部分元素赋初值。

```
int a[3][4]={{1},{5},{9}};
```

①

```
int a[3][4]={{1},{0,6},{0,0,11}};
```

②

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

③

```
int a[3][4]={{1},{},{9}};
```

④

①	②	③	④
1 0 0 0	1 0 0 0	1 0 0 0	1 0 0 0
5 0 0 0	0 6 0 0	5 6 0 0	0 0 0 0
9 0 0 0	0 0 11 0	0 0 0 0	9 0 0 0

(4)如果对全部元素都赋初值(即提供全部初始数据)，则定义数组时对第1维的长度可以不指定，但第2维的长度不能省。

```
int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};
```

≡

```
int a[][4]={1,2,3,4,5,6,7,8,9,10,11,12};
```

在定义时也可以只对部分元素赋初值而省略第1维的长度，但应分行赋初值。

```
int a[][4]={{0,0,3},{},{0,10}};
```

## 二维数组程序举例

【例6.4】将一个二维数组行和列的元素互换，存到另一个二维数组中。

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \Rightarrow b = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

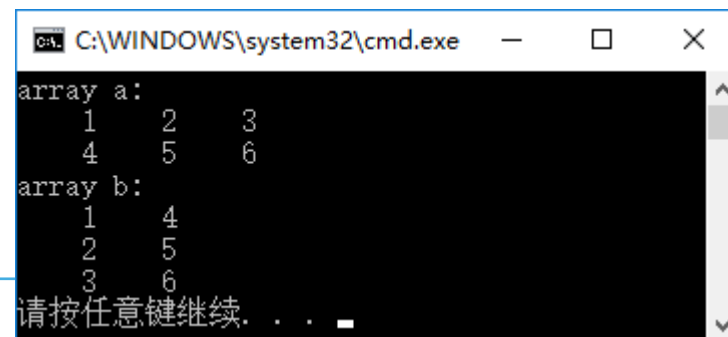
# 二维数组程序举例

【例6.4】将一个二维数组行和列的元素互换，存到另一个二维数组中。

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \Rightarrow b = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

```
#include <stdio.h>
int main()
{
    int a[2][3]={1,2,3},{4,5,6};
    int b[3][2],i,j;
    printf("array a:\n");
    for(i=0;i<=1;i++)           //处理a数组中的一行中各元素
    {
        for (j=0;j<=2;j++)      //处理a数组中某一列中各元素
        {
            printf("%5d",a[i][j]); //输出a数组的一个元素
            b[j][i]=a[i][j]; //将a数组元素的值赋给b数组相应元素
        }
    }
```

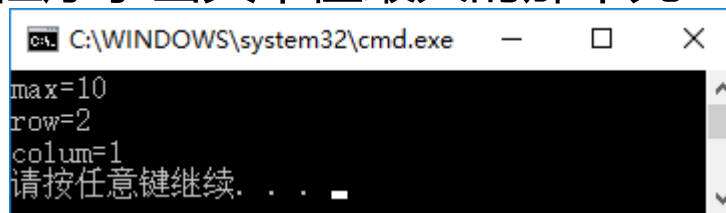
```
        printf("\n");
    }
    printf("array b:\n");
    for(i=0;i<=2;i++)           //输出b数组各元素
    {                             //处理b数组中一行中各元素
        for(j=0;j<=1;j++)       //处理b数组中一列中各元素
        {
            printf("%5d",b[i][j]); //输出b数组的一个元素
        }
        printf("\n");
    }
    return 0;
```



```
C:\WINDOWS\system32\cmd.exe
array a:
    1    2    3
    4    5    6
array b:
    1    4
    2    5
    3    6
请按任意键继续. . . .
```

## 二维数组程序举例

【例6.5】有一个 $3 \times 4$ 的矩阵，要求编程序求出其中值最大的那个元素的值，以及其所在的行号和列号。

A screenshot of a Windows command prompt window. The title bar shows 'C:\WINDOWS\system32\cmd.exe'. The command prompt contains the following text: 'max=10', 'row=2', 'column=1', and '请按任意键继续. . .'.

```
C:\WINDOWS\system32\cmd.exe
max=10
row=2
column=1
请按任意键继续. . .
```



先思考一下在打擂台时怎样确定最后的优胜者。先找出任一人站在台上，第2人上去与之比武，胜者留在台上。再上去第3人，与台上的人(即刚才的得胜者)比武，胜者留台上，败者下台。以后每一个人都是与当时留在台上的人比武。直到所有人都上台比过为止，最后留在台上的就是冠军。



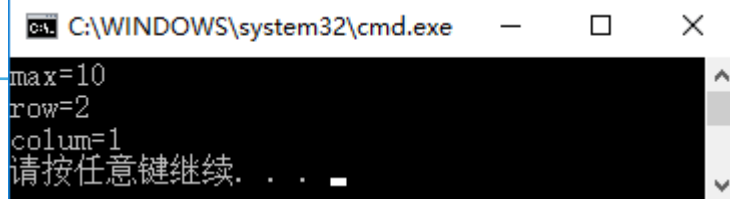
## 二维数组程序举例

算法

找最大最小值  
打擂台算法

【例6.5】有一个 $3 \times 4$ 的矩阵，要求编程序求出其中值最大的那个元素的值，以及其所在的行号和列号。

```
#include <stdio.h>
int main()
{   int i,j,row=0,column=0,max;
    int a[3][4]={{1,2,3,4},{9,8,7,6},{-10,10,-5,2}};    //定义数组并赋初值
    max=a[0][0];                                           //先认为a[0][0]最大
    for(i=0;i<=2;i++)
        for(j=0;j<=3;j++)
            if(a[i][j]>max)                                //如果某元素大于max, 就取代max的原值
            {   max=a[i][j];
                row=i;                                     //记下此元素的行号
                column=j;                                  //记下此元素的列号
            }
    printf("max=%d\nrow=%d\ncolumn=%d\n",max,row,column);
    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe
max=10
row=2
column=1
请按任意键继续. . .
```

先思考一下在打擂台时怎样确定最后的优胜者。先找出任一人站在台上，第2人上去与之比武，胜者留在台上。再上去第3人，与台上的人(即刚才的得胜者)比武，胜者留台上，败者下台。以后每一个人都是与当时留在台上的人比武。直到所有人都上台比过为止，最后留在台上的就是冠军。

