



《计算概论A》课程 程序设计部分

C++程序中的数组

李 戈

北京大学 信息科学技术学院 软件研究所

lige@sei.pku.edu.cn



北京大学



回顾：数组的定义

数组的定义

类型说明符 数组名 [常量表达式]

例： `float sheep[10];`
 `int a2001[1000];`

说明

- 1.数组名的第一个字符应为英文字母；
- 2.用方括号将常量表达式括起；
- 3.常量表达式定义了数组元素的个数；

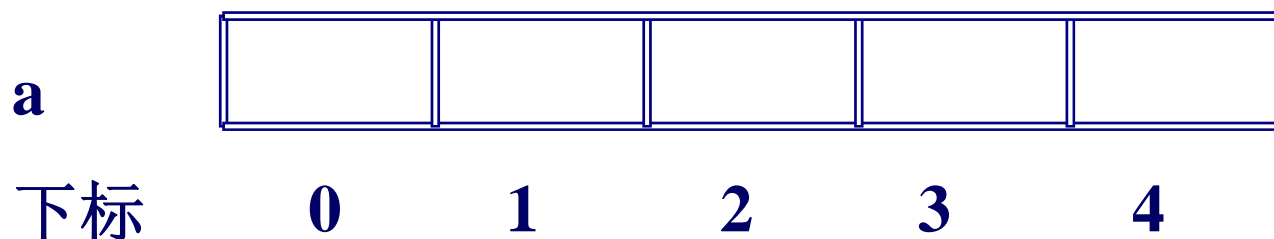


北京大学



回顾：数组的定义

- 4. 数组下标从0开始。如果定义5个元素，是从第0个元素至第4个元素；



例如 `int a[5]` 定义了5个数组元素如下：

`a[0]`, `a[1]`, `a[2]`, `a[3]`, `a[4]`

这是5个带下标的变量，这5个变量的类型是相同的

- 5. 常量表达式中不允许包含变量；

例如 `int n;`
 `n = 5;`
 `int a[n];`

不合法！



北京大学



回顾：数组的定义

```
#include <iostream>
using namespace std;
void main()
{
    const int i = 4;
    int a[i]={1, 2, 3, 4};
    cout<<"a[0]="<<a[0]
        <<" "; a[1]="<<a[1]
        <<" "; a[2]="<<a[2]
        <<" "; a[3]="<<a[3]
        <<endl;
}
```



北京大学



回顾：数组的定义

```
#include <iostream>
using namespace std;
#define I 4
void main()
{
    int a[I]={1, 2, 3, 4};
    cout<<"a[0]="<<a[0]
        <<" "; a[1]="<<a[1]
        <<" "; a[2]="<<a[2]
        <<" "; a[3]="<<a[3]
        <<endl;
}
```



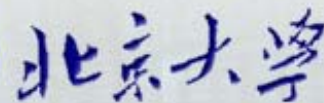
北京大学



```
1. #include <iostream>  
using namespace std;  
void main()  
{  
    int a[4];  
    cout<<"a[0]="<<a[0]<<"a[1]="<<a[1]<<"  
    a[2]="<<a[2]<<"a[3]="<<a[3]<<endl;  
}
```

2. 其他不变，改变声明项为

```
int a[ ] = { 0, 1, 2, 3 };
```





回顾：数组的初始化

3. 其他不变，改变声明项为

```
int a[4] = { 3, 8 };
```

4. 其他不变，改变声明项为

```
int a[4] = { 2, 4, 6, 8, 10 };
```

5. 其他不变，改变声明项为

```
int a[4] = { 2, 4, 6, d };
```

6. 其他不变，改变声明项为

```
int n=4;
```

```
int a[n] = { 0, 1, 2, 3 };
```



北京大学



数组的作用

■ 数组的作用

- ◆ 当你有一些数据要进行存储时：
 - 数组可以用于存放一系列数据类型相同的数据；
- ◆ 当你的处理对象是连续的数字时：
 - 数组中的数据与数组下标之间建立了对应关系；
 - 可以利用对应关系，进行记录计算状态；



北京大学



(一) 利用“对应关系”解题



北京大学



例一：找出素数（1）

■ 问题：

◆ 请编写程序，输出100以内的所有素数；



北京大学



例一：找出素数（1）

■ 问题：

- ◆ 请编写程序，输出100以内的所有素数；

■ 典型的解决方法：

◆ 循环（i 从 2 至 100）

- 设置一个标识 $p = 0$;

- 循环（j 从 2 至 $i-1$ ）

- ◆ 如果 $(i \text{ mode } j == 0)$ $p = 1$

- 如果 $p = 0$ ，输出 i



北京大学



例一：找出素数（1）

■ 问题：

◆ 请编写程序，输出100以内的所有素数；

■ 筛法求素数：

◆ 埃拉托斯特尼（Eratosthenes，约公元前274～194年）发明，又称埃拉托斯特尼筛子。

◆ “挑选”与“筛选”

● 不是挑选出所有素数，而是筛掉所有的合数；

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	...	a[99]
1	1	0	0	1	0		1



北京大学



例一：找出素数（2）

■ 如何找到所有的合数？

- ◆ 合数都可以分解为某两个数的乘积，也就是说，
合数可以看做是由某个数相加多次而得到的；
- ◆ 因此，可以让 $2, 3, 4, 5, \dots, c$ 中的每个数自我相加多次，来获得100之内的所有合数；
- ◆ c 如何确定？

- 根据初等数论，若 n 为合数，则 n 的最小正因数 c 满足：

$$1 < c \leq \sqrt{n}$$



北京大学



例一：找出素数 (3)

■ 解题思路：

- ◆ 将数组 $a[n]$ ($0 \leq n < 100$) 所有元素设置为0;
- ◆ 筛出所有合数：
 - 分别计算2, 3, 4, 5, ..., c自我相加多次的数值;
 - ◆ 每次计算得到的结果都是一个合数，在数组中标记该数字被“筛掉”；
 - ◆ 每次计算过程中，只要相加结果没有达到100就继续自我相加；
- ◆ 根据标记输出所有没有被筛掉的数字；

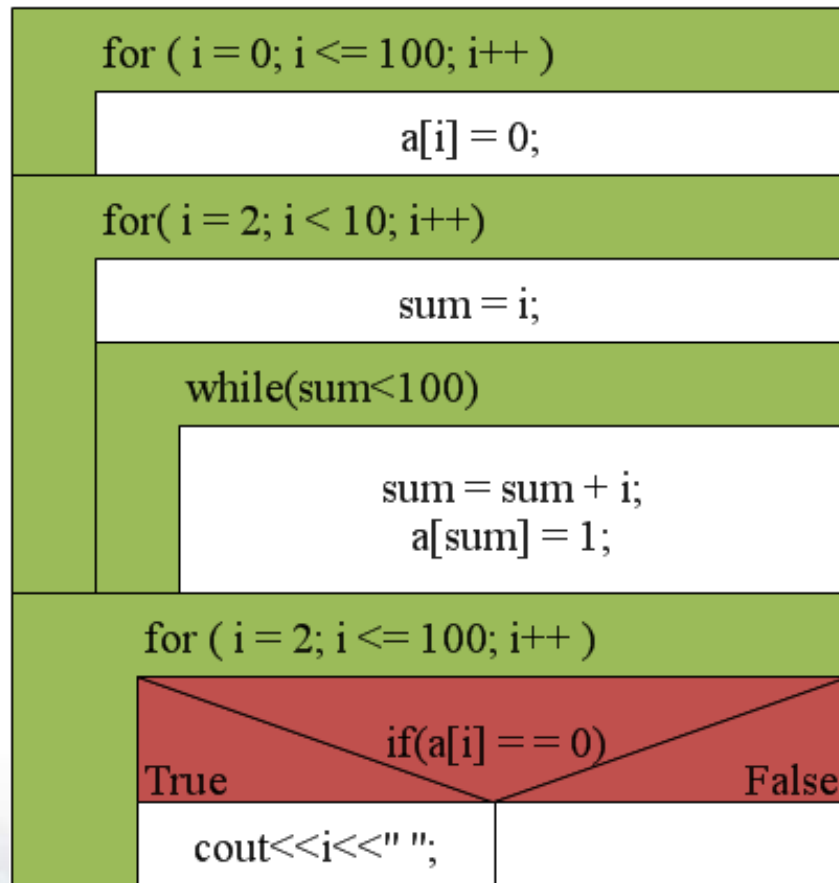


北京大学

```
int main( )
{
    int a[100];
    int sum = 0;
    for(int i = 0; i < 100; i++)
        a[i] = 0;
    for(int i = 2; i < sqrt(100.0); i++)
    {
        sum = i;
        while(sum < 100)
        {
            sum = sum + i;
            a[sum] = 1;
        }
    }
    for(int i = 0; i < 100; i++)
        if(a[i] == 0)
            cout << i << " ";
    return 0;
}
```




N-S图与程序



```
#include<iostream>
#include<cmath>
using namespace std;
int main( )
{
    int a[100];
    int sum = 0;
    for(int i = 0; i < 100; i++)
        a[i] = 0;
    for(int i = 2; i < sqrt(100.0); i++)
    {
        sum = i;
        while(sum<100)
        {
            sum = sum + i;
            a[sum] = 1;
        }
    }
    for(int i = 0; i < 100; i++)
        if(a[i] == 0)
            cout<<i<<" ";

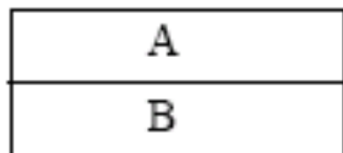
    return 0;
}
```



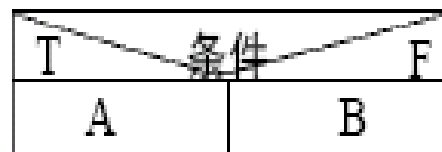
用N-S图描述算法

■ N-S图

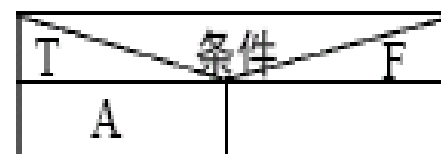
- ◆ 一种算法表示法，由美国人I. Nassi和B. Shneiderman共同提出，是算法的一种结构化描述方法。



顺序结构

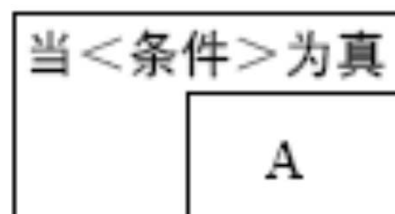


a)



b)

选择结构的N-S图



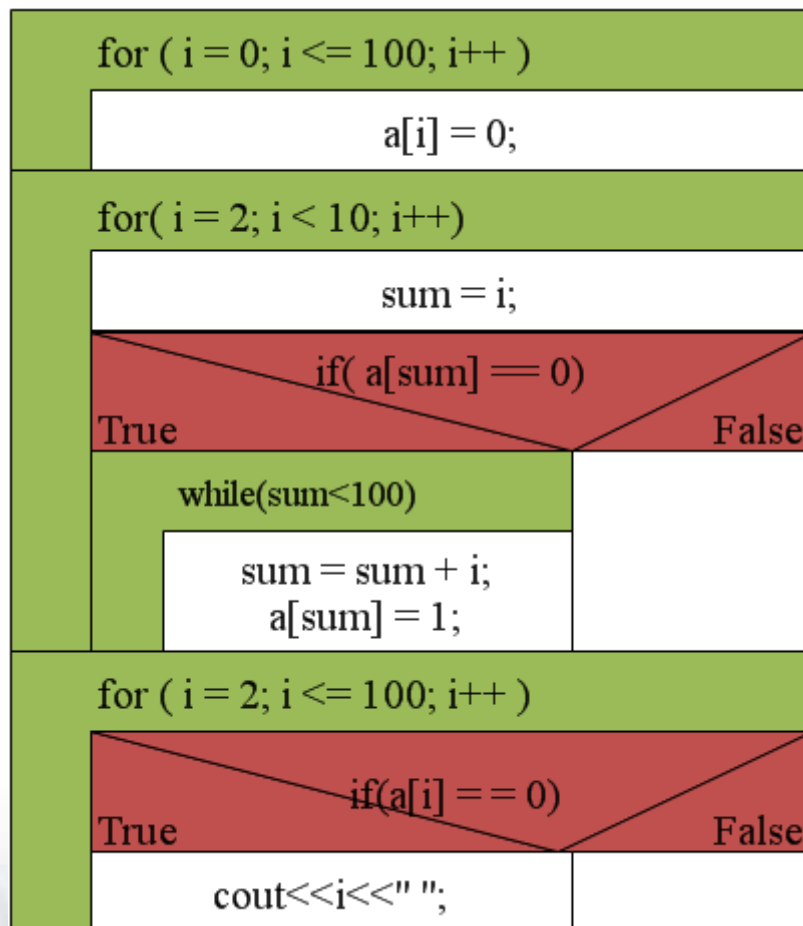
循环结构的N-S图



北京大学



优化的程序



```
#include<iostream>
#include<cmath>
using namespace std;
int main( )
{
    bool a[100] = {0};
    int sum = 0;
    for(int i = 0; i < 100; i++)
        a[i] = 0;
    for(int i = 2; i < sqrt(100.0); i++)
    {
        sum = i;
        if(a[sum]==0)
        {
            while(sum<100)
            {
                sum = sum + i;
                a[sum] = 1;
            }
        }
    }
    for(int i = 0; i < 100; i++)
        if(a[i] == 0)
            cout<<i<<" ";

    return 0;
}
```



例二：数字统计（1）

■ 问题：

◆ 输入1000个0~50的整数，请你统计每个数在输入数列中出现的次数。

■ 现在你能给出几种解决方案？



北京大学



例二：数字统计 (2)

■ 方法一：

```
for (i = 1; i <= 1000; i++)  
{  
    cin >> num;  
    for (j = 0; j < 50; j++)  
    {  
        if (num==j)  
            count[j]++;  
    }  
}
```



例二：数字统计 (3)

■ 方法二：

```
for (i=1;i<=1000;i++)  
{  
    cin >> num;  
    switch(num)  
    {  
        case 0 : count[0]++;break;  
        case 1 : count[1]++;break;  
        case 2 : count[2]++;break;  
        ...  
    }  
}
```





例二：数字统计（4）

■ 更简单的办法：

```
for (i = 0; i < 1000; i++)  
{  
    cin >> num;  
    count[num]++;  
}
```



北京大学



(二) 利用数组完成数列排序



北京大学



典型问题之一：数列排序

- 编写一个程序，该程序接受一系列数字作为输入，如：
 - ◆ 1, 8, 3, 2, 4, 9
- 要求该程序完成将该数列按照从大到小的顺序排列，并输出。



北京大学

问题：采用冒泡排序法，将几个数从大到小排序并输出

	i=1	i=2	i=3	i=4	i=5	i=6
	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
初始值	1	8	3	2	4	9
1<8; 1, 8互换	1 ↔ 8	3	2	4	9	
1<3; 1, 3互换	8	1 ↔ 3	2	4	9	
1<2; 1, 2互换	8	3	1 ↔ 2	4	9	
1<4; 1, 4互换	8	3	2	1 ↔ 4	9	
1<9; 1, 9互换	8	3	2	4	1 ↔ 9	
1到达位置	8	3	2	4	9	1
8>3; 顺序不动	8	3	2	4	9	1
3>2; 顺序不动	8	3	2	4	9	1
2<4; 2, 4互换	8	3	2 ↔ 4	9	1	
2<9; 2, 9互换	8	3	4	2 ↔ 9	1	
2到达位置	8	3	4	9	2	1

j=1

j=2

	i=1	i=2	i=3	i=4	i=5	i=6
	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
中间结果	8	3	4	9	2	1
8>3; 顺序不动	8	3	4	9	2	1
3<4; 3, 4互换	8	3 ↔ 4		9	2	1
3<9; 3, 9互换	8	4	3 ↔ 9		2	1
3到达位置	8	4	9	3	2	1
8>4; 顺序不动	8	4	9	3	2	1
4<9; 4, 9互换	8	4 ↔ 9		3	2	1
4到达位置	8	9	4	3	2	1
8<9; 8, 9互换	8 ↔ 9		4	3	2	1
8到达位置	9	8	4	3	2	1

j=3

j=4

j=5



冒泡排序算法分析 (1)

- 总共冒了几次泡？（外层循环） **6-1次**

`for(j = 1; j <= 6-1; j++)`

	i=1	i=2	i=3	i=4	i=5	i=6	
	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	
中间结果	8	3	4	9	2	1	
8>3; 顺序不动	8	3	4	9	2	1	j=3
3<4; 3, 4互换	8	3↔4	9	2	1		
3<9; 3, 9互换	8	4	3↔9	2	1		
3到达位置	8	4	9	3	2	1	
8>4; 顺序不动	8	4	9	3	2	1	j=4
4<9; 4, 9互换	8	4↔9	3	2	1		
4到达位置	8	9	4	3	2	1	
8<9; 8, 9互换	8↔9	4	3	2	1		j=5
8到达位置	9	8	4	3	2	1	



冒泡排序算法分析 (2)

- 每次冒泡作了几次比较？（内层循环） **6-j**次

```
for( j = 1; j <= 6-1; j++)
```

```
for( i = 1; i <= 6-j; i++)
```

	i=1	i=2	i=3	i=4	i=5	i=6
	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
初始值	1	8	3	2	4	9
1<8: 1, 8互换	1 ↔ 8	3	2	4	9	
1<3: 1, 3互换	8	1 ↔ 3	2	4	9	
1<2: 1, 2互换	8	3	1 ↔ 2	4	9	
1<4: 1, 4互换	8	3	2	1 ↔ 4	9	
1<9: 1, 9互换	8	3	2	4	1 ↔ 9	
1到达位置	8	3	2	4	9	1
8>3: 顺序不动	8	3	2	4	9	1
3>2: 顺序不动	8	3	2	4	9	1
2<4: 2, 4互换	8	3	2 ↔ 4	9	1	
2<9: 2, 9互换	8	3	4	2 ↔ 9	1	
2到达位置	8	3	4	9	2	1

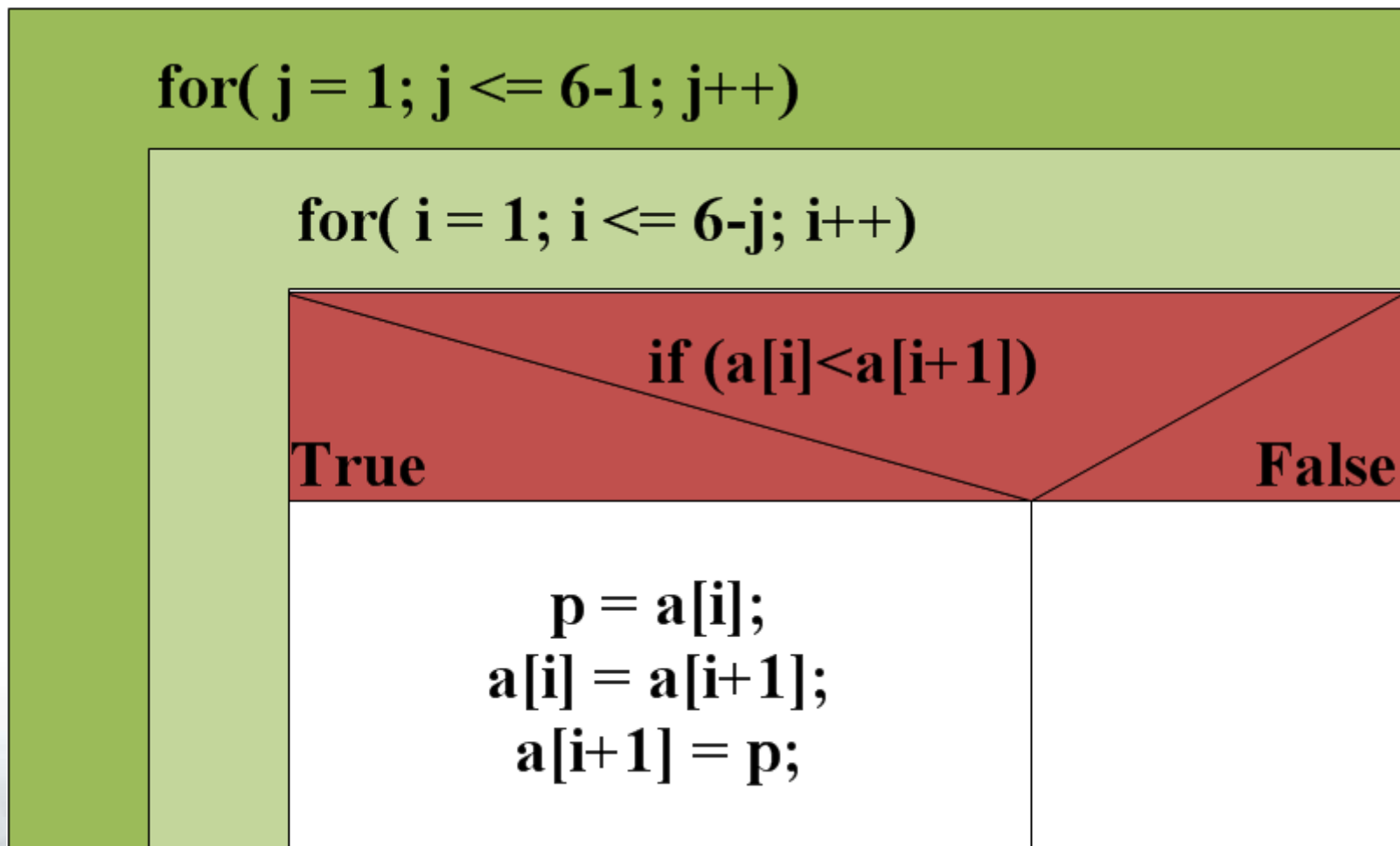
j=1

j=2



冒泡排序算法分析 (3)

■ 每次比较作了些什么？




```

#include <iostream>
using namespace std;
void main()
{
    int i, j, p, a[7];
    for (i = 1; i <= 6; i = i+1)
    {
        cout<<"请输入待排序的数a["<< i<<"] = ";
        cin>>a[i];
    }
    for ( j = 1; j <= 5; j = j+1)           // 外层循环
        for ( i = 1; i <= 6 - j; i = i + 1 ) // 内层循环
        {
            if ( a[i] < a[i+1] )
            {
                p = a[i];
                a[i] = a[i+1];
                a[i+1] = p;
            }
        }
    for ( i=1; i<=6; i=i+1)                // 输出排序结果
        cout<<a[i];
}

```



典型问题之二：数列插入

- 例: 给定一个包含20个元素的数组，请将数组排序，并将新给定的数插入到有序的数组中.



北京大学



典型问题之二：数列插入

- 例: 给定一个包含20个元素的数组，请将数组排序，并将新给定的数插入到有序的数组中.
- 五个步骤：
 - ◆ 建立数组
 - ◆ 排序
 - ◆ 找出被插入值所在的位置。
 - ◆ 从该位置起到最后的所有元素右移一位
 - ◆ 将被插入数值放入空出的位置





典型问题之二：数列插入

```
int main()
{ int a[20],i,p,k,num;
  cout<<"输入20个整数\n";
  for (i=0;i<20;i++)
    cin>>a[i]);           //输入
  for (i = 1; i <= 20-1; i++)      //排序
    for (j = 0; j < 20 - i ; j++)
      if a[j]>a[j+1]
        { t = a[j]; a[j] = a[j+1]; a[j+1] = t;}
```



北京大学



典型问题之二：数列插入

```
cout<<“输入要被插入的数\n”;
cin>>num;
p = 0;
while (p < 20)
{
    if (a[p] >= num) break;
    else p++;
}    //查找
for (k = 19;k >= p; k--)
    a[k+1]=a[k];    //右移一位
a[p] = num;    //插入
for (i = 0; i < 21; i++)
    cout<<setw(4)<<a[i]; //输出结果
}
```



北京大学



二维数组



北京大学

二维数组的定义

$a[0][0]$	$a[0][1]$...	$a[0][m]$
$a[1][0]$	$a[1][1]$...	$a[1][m]$
$a[2][0]$	$a[2][1]$...	$a[2][m]$
$a[3][0]$	$a[3][1]$...	$a[3][m]$
... ..			
$a[n-1][0]$	$a[n-1][1]$...	$a[n-1][m-1]$

	0	1	2	3	4	5	6	7	8
0	0	0	1	2	2	3	0	0	0
1	0	2	3	5	5	3	2	0	0
2	0	1	4	3	4	2	2	1	0
3	0	0	1	2	0	0	1	1	0
4	0	0	0	0	0	0	0	0	0
5	0	0	1	2	2	3	0	0	0
6	0	0	1	2	2	3	0	0	0
7	0	2	3	5	5	3	2	0	0
8	0	1	4	3	4	2	2	1	0
5									
0									



北京大学

二维数组的定义

定义：类型说明符 [常量表达式][常量表达式]

例如：float a[3][4]

a[0][0] ... a[0][3]

a[1][0] ... a[1][3]

a[2][0] ... a[2][3]

可以理解为三个一维数组即：

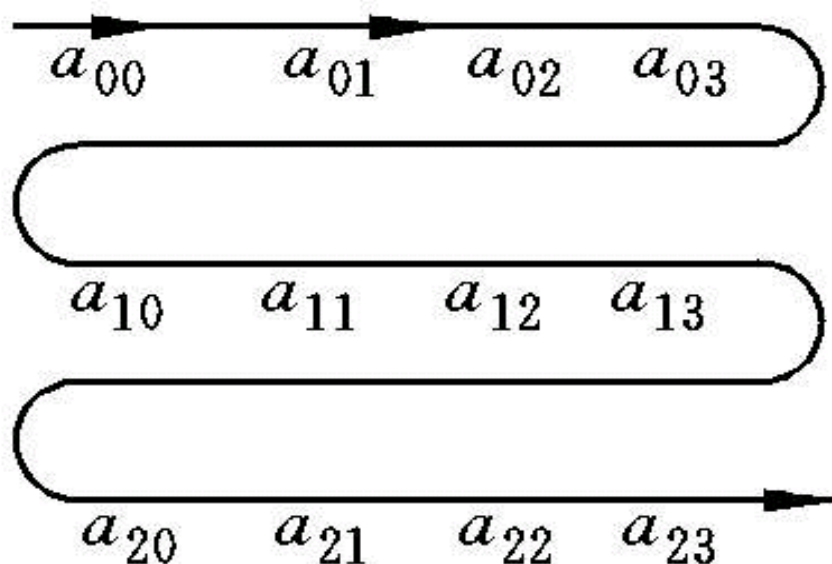
float a[0][4], a[1][4], a[2][4]

此处a[0], a[1], a[2]是一维数组名

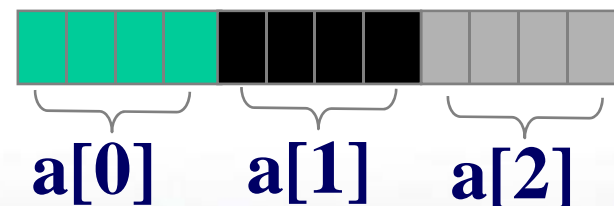
$$a \left\{ \begin{array}{l} a[0] \text{ -- } a[0][0] \ a[0][1] \ a[0][2] \ a[0][3] \\ a[1] \text{ -- } a[1][0] \ a[1][1] \ a[1][2] \ a[1][3] \\ a[2] \text{ -- } a[2][0] \ a[2][1] \ a[2][2] \ a[2][3] \end{array} \right.$$

二维数组元素的排列顺序

- C++ 语言中,二维数组中元素排列的顺序是:
按行存放,即在内存中先顺序存放第一行的元素,再存放第二行的元素。
 - ◆ 数组 $a[3][4]$ 的存放顺序



在内存中连续12
个浮点数单元



北京大学

二维数组的定义

■ 引用数组元素

◆ 程序中引用的是数组中一个元素，必须用下标来指定。

◆ 如：数组名[下标][下标]

◆ `float a[4][4]`

`a[0][0] a[0][1] a[0][2] a[0][3]`

`a[1][0] a[1][1] a[1][2] a[1][3]`

`a[2][0] a[2][1] a[2][2] a[2][3]`

`a[3][0] a[3][1] a[3][2] a[3][3]`

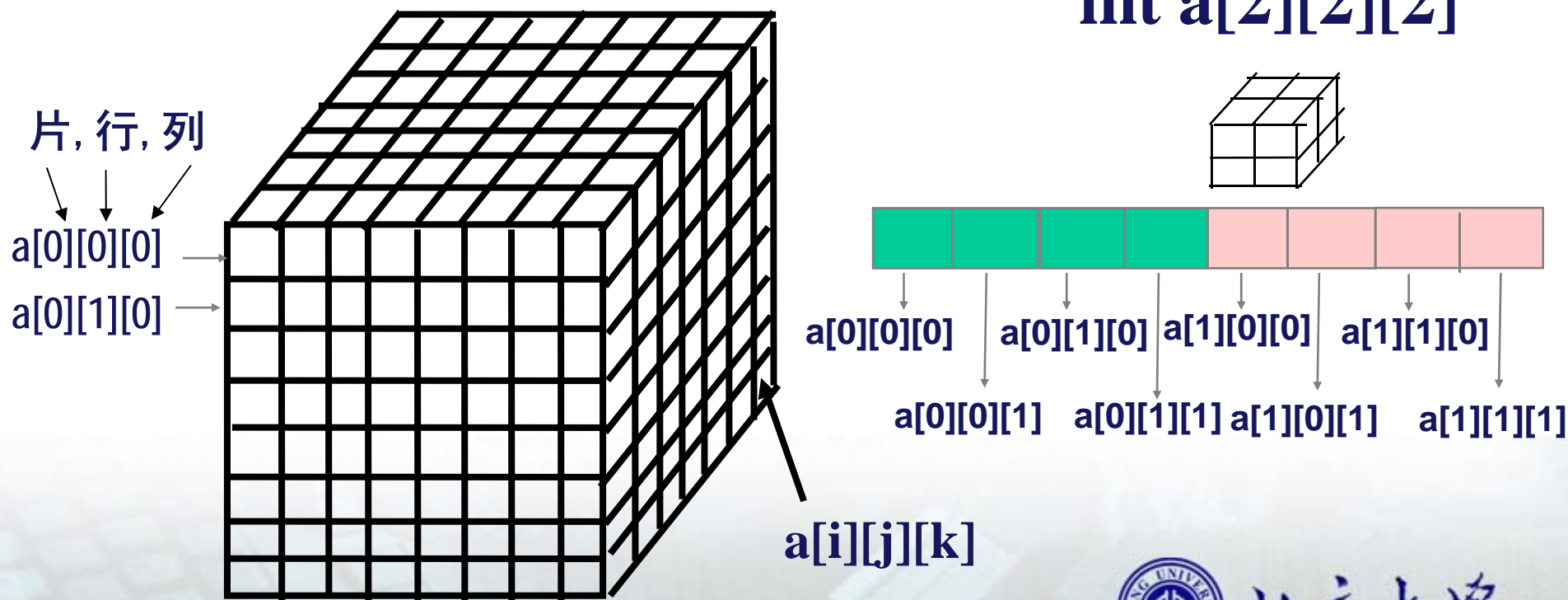
其中a[2][3]指的是第三行第四个元素。



北京大学

三维数组的定义

- 三维数组的定义与二维数组相同。
 - ◆ 如: `float a[2][3][4]` $2 \times 3 \times 4$ 共有24个元素。
 - ◆ `float a[100][100][100]` 有100 0000个元素



北京大学



二维数组的初始化

■ 二维数组初始化

(1) 分行给二维数组赋初值。

◆ 如: `int a[3][4]=`
`{ {1, 2, 3, 4}, {5, 6, 7, 8}, {50, 10, 11, 12} };`

(2) 可以将所有数据写在一个花括弧内,按数组排列的顺序对各元素赋初值。

◆ 如: `int a[3][4]=`
`{1, 2, 3, 4, 5, 6, 7, 8, 50, 10, 11, 12};`



北京大学



二维数组的初始化

■ 二维数组的初始化

(3) 可以对部分元素赋初值。

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

◆ 它的作用是只对各行前几列的元素赋初值, 其余元素值自动为0。赋值后的数组元素如下:

1	0	0	0
0	6	0	0
0	0	11	0

(4) 如果对全部元素都赋初值, 则定义数组时对第一维的长度可以不指定, 但第二维的长度不能省。

```
int a[ ][4] = { 1, 2, 3, 4, 5, 6, 7, 8, 50, 10, 11, 12};
```



北京大学

二维数组的赋值

■ 常用的数组赋值的方法

1. 用cin语句由用户输入

```
for (i=1;i<=3;i++)
```

```
    for (j=1;j<=4;j++)
```

```
        cin>>a[i][j];
```

2. 与循环变量之间有规律

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

```
for (i=0;i<4;i++)  
    (for j=0;j<4;j++)  
        a[i][j] = 4*i+j;
```



北京大学

赋值练习：

- 将一个二维数组行和列元素互换,存到另一个数组中.

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

```
for (i=0;i<3;i++)  
    for (j=0;j<2;j++)  
        b[i][j]=a[j][i]
```

```
for (j=0;j<2;j++)  
    for (i=0;i<3;i++)  
        b[i][j]=a[j][i]
```



北京大学

二维数组的应用举例

- 车队各车型一周行程表及耗油情况如下,计算车队一周的油费。

	一	二	三	四	五	六	七
大卡车	1600	2300	4000	7500	2400	2400	3200
小卡车	7500	6300	7200	55000	55000	6000	5500
小汽车	3000	25000	3100	2700	2700	3500	4000

- 元/公里
 - ◆ 大卡车 1.5
 - ◆ 小卡车 1.0
 - ◆ 小汽车 0.8



北京大学

二维数组的应用举例

$\text{dis}[0] = a[0][0] + a[0][1] + a[0][2] + \dots + a[0][6]$

$\text{dis}[1] = a[1][0] + a[1][1] + a[1][2] + \dots + a[1][6]$

$\text{dis}[2] = a[2][0] + a[2][1] + a[2][2] + \dots + a[2][6]$

每种车型的总里程

for (i=0;i<3;i++)

for (j=0;j<7;j++)

dis[i]=dis[i]+a[i][j];

每种车型的总消费

for (i=0;i<3;i++)

cost[i]=dis[i]*price[i]



北京大学

```
int main()
{ int dis[7]={0,0,0},
  float price[3]={1.5,1.0,0.8};
  int a[3][7]={ {1600,2300,4000,7500,2400,2400,3200},
                {7500,6300,7200,55000,55000,6000,5500},
                {3000,25000,3100,2700,2700,3500,4000}}
  int total = 0, i, j;
  for (i=0; i<3; i++)
  {
    for(j=0; j<7; j++)  dis[i] = dis[i] + a[i][j];
    total=total+dis[i]*price[i];
  }
  cout<<“本周总消耗为” <<total;
  return 0;
}
```



分数段统计 (1)

■ 问题

- ◆ 期中考试结束后，需要按 <60 ， $60\sim 69$ ， $70\sim 79$ ， $80\sim 89$ ， $90\sim 99$ ， 100 各分数段统计全班**50**名学生在各分数段的人数。
- ◆ 要求：依次输入学生成绩，打印出各分数段的人数



北京大学



分数段统计 (2)

■ 思路:

- ◆ 利用数组的对应关系解题;

■ 方法:

- ◆ 将分数段编成数组下标索引 $k = \text{num} / 10$
- ◆ 60以下的分数为一个档次 $\text{if } (k < 6) \ k = 5$
- ◆ 将下标与数组对应 $k = k - 5$
- ◆ 相应的数组值加1



北京大学



分数段统计 (3)

```
int main()
{  int a[6]={0,0,0,0,0,0};
    int i,k,num;
    cout<<"输入50个学生的成绩\n";
    for (i=0;i<50;i++)
    {   cin>>num; k = num/10;
        if (k<6) k=5;
        k = k-5; //数组下标从0到5分别表示6个分数段
        a[k]++;
    }
    for (i= 0;i<6;i++) cout<<setw(4)<<a[i];
    return 0;
}
```



北京大学



二维数组应用举例

■ 问题

- ◆ 某学校有**1000**位老师，分布在**20**个不同的学院中，每个学院最多有**12**个系，请你编写一个程序，输入每位老师的所在院、系的编号（院编号**1-20**，系编号**1-12**），打印出各个系老师的数量。

■ 分析

- ◆ 你的解决方案？



北京大学

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{  int teacher[21][13];
    int school, department;
    int i,j;
    char name[30];

    for (i=1;i<21;i++)
        for(j=1;j<13;j++)
            teacher[i][j]=0;

    for(i=0;i<1000;i++)
    {
        cin>>name>>school>>department;
        teacher[school][department]++;
    }

    for (i=1;i<21;i++)
        for(j=1;j<13;j++)
            cout << setw(4) << teacher[i][j] ;
    cout << endl;
    return 0;
}
```



整数奇偶排序

■ Description

输入10个0~100之间的不同整数，彼此以空格分隔 重新排序以后输出(也按空格分隔)，要求：

- 1.先输出其中的奇数,并按从大到小排列；
- 2.然后输出其中的偶数,并按从小到大排列。

■ Input

任意排序的10个整数（0~100），彼此以空格分隔

■ Output

按照要求排序后输出，由空格分隔



北京大学

```
#include<iostream>
using namespace std;
int main(){
    int a[10],b[10],c[10],i,j,k,m,n,t;
    while(cin>>a[0]>>a[1]>>a[2]>>a[3]>>a[4]>>a[5]>>a[6]>>a[7]>>a[8]>>a[9]){
        j=0;
        k=0;
        for(i=0;i<=9;i++){
            if(a[i]%2!=0) b[j++]=a[i];
            if(a[i]%2==0) c[k++]=a[i];
        }
        for(i=0;i<j;i++){
            for(m=j-1;m>=1+i;m--){
                if(b[m]>b[m-1]){
                    t=b[m-1];
                    b[m-1]=b[m];
                    b[m]=t;
                }
            }
        }
    }
}
```

```
    for(i=0;i<k-1;i++){
        for(n=0;n<k-1-i;n++){
            if(c[n]>c[n+1]){
                t=c[n];
                c[n]=c[n+1];
                c[n+1]=t;
            }
        }
    }
    for(i=0;i<j;i++){
        cout<<b[i]<<" ";
    }
    for(i=0;i<k;i++){
        cout<<c[i]<<" ";
    }
    cout<<endl;
}
return 0;
}
```

```
#include <iostream>
using namespace std;
int main( ){
    int a[10], i, j;
    while (cin>>a[0]>>a[1]>>a[2]>>a[3]>>a[4]>>a[5]>>a[6]>>a[7]>>a[8]>>a[9]){
        for (i=99; i>=1; i=i-2){
            for (j=0; j<=9; j++){
                if (a[j]==i)
                    cout<<a[j]<<" ";
            }
        }
        for (i=0; i<=100; i=i+2){
            for (j=0; j<=9; j++){
                if (a[j]==i)
                    cout<<a[j]<<" ";
            }
        }
        cout<<endl;
    }
    return 0;
}
```



程序应写成什么样？



北京大学



写程序注意事项

- 变量命名
 - ◆ 匈牙利命名法 or 自己的命名方法;
- 注释！注释！注释！
 - ◆ 变量定义后
 - ◆ 函数定义前
 - ◆ 重要的逻辑判断之后
 - ◆ 输入输出之后
- 缩进！



北京大学



写程序注意事项

■ 最容易犯的错误

◆ `if (x = 1)`

◆ `int sum;`

`for(i = 0; i < 100; i++)`

`sum++;`

◆ `cout<<sum;`

◆ `int a[100][100]`

`... a[i][j]...`

谨慎!



北京大学



好好想想,有没有问题?

谢谢!



北京大学