

数组



CONTENTS

01 一维数组

02 二维数组

03 字符数组与字符串

04 数组与函数

05 数组程序设计举例



回顾一个例子

输入三个整数，并将
它们按从小到大顺序
输出。

Arraytest.c

```
#include<stdio.h>
int main(){
    int num1,num2,num3,t;
    printf("Please input 3 numbers:");
    scanf("%d%d%d", &num1, &num2, &num3);
    if(num1 > num2){
        t = num1;
        num1 = num2;
        num2 = t;
    }
    if(num1 > num3){
        t = num1;
        num1 = num3;
        num3 = t;
    }
    if(num2 > num3){
        t = num2;
        num2 = num3;
        num3 = t;
    }
    printf("Numbers in ascending order : %-5d%-5d%-5d\n",num1,num2,num3);
    return 0;
}
```

如果是10个数呢？

如果是100个数呢？



Please input 3 numbers: 56 78 23

Numbers in ascending order : 23 56 78



陕西科技大学

SHAANXI UNIVERSITY OF SCIENCE & TECHNOLOGY

数组是什么？

- 数组是**有顺序**数据的集合。
- 数组中的每一个元素都属于**同一个数据类型**。
- 数组用一个**统一的数组名**和**下标**来唯一地确定数组中的每一个元素。



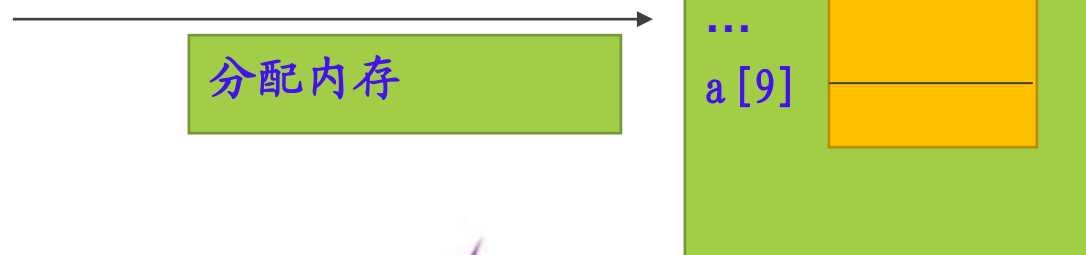
一维数组的定义

1. 类型说明符 数组名 [常量表达式];

例: `int a[10];`//定义一个含有10个整型变量的名为a的数组

`float b[20],c[30];`//定义一个含有20个实型变量的名为b的数组和一个含有30个实型变量的名为c的数组

2. 在内存分配若干连续空间给数组.



那100个数的问题

Arraytest.c *

```
#include<stdio.h>
int main(){
    //int num1,num2,num3,t;
    int num[100],t;
```

int num[100];



比int num1,num2.....num100; 方式要简单许多。



3. 注意事项:

- a) 数组类型为数组中每一个元素类型;
- b) 数组名命名规则与标识符相同;
- c) 方括号中只能写常量表达式表示数组长度;

/*以下是一个典型的
错误定义方式!*/

```
int n;  
scanf("%d",&n);  
int a[n];
```



一维数组的引用

C语言规定，只能引用单个数组元素，不能一次引用整个数组

引用数组元素的方法是：

数组名[下标]



下标是整型表达式
下标最小值为0



设有定义 `int a[10]; int i;`

`/*可以引用元素的从a[0]到a[9]*/`

`a[5]=6; a[7]=a[5]++;`

`a[6]=3; a[0]=a[5]+a[7]-a[2*3];`

`/*下标也可以使用变量*/`

`i = 2; a[i] = 0;`

`i = 8; a[i] = 12;`

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
10		0			6	3	6	12	



【例题1】输入10个整数，分别按顺序和逆序输出。

```
#include <stdio.h>
int main(){
    int i, a[10];
    printf("input 10 numbers:\n");
    for(i=0; i<10; i++)
        scanf("%d", &a[i]);
    printf("\n");
    for(i=0; i<=9; i++)
        printf("%d  ", a[i]);
    printf("\n");

    return 0;
}
```

```
input 10 numbers:
7
8
9
6
3
2
1
4
5
0

7 8 9 6 3 2 1 4 5 0
```



一维数组的初始化

- 1.初始化--在定义数组时对数组元素赋初值;
- 2.初始化方法:

数组类型 数组名 [数组长度]={初值列表};



①给全部元素赋初值

```
static int a[5]={0,1,2,3,4};
```

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

②可以只给一部分元素赋初值;

```
static float b[5]={ , ,1.4,7.2};
```

b[0]	b[1]	b[2]	b[3]	b[4]
0.0	0.0	1.4	7.2	0.0

③在对全部数组元素赋初值时，可以不指定数组长度。 `int a[3] = {1,2,3,4};`

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

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



注意无初始化时

static->0

auto->随机



【例题2】分析下面程序的运行结果。

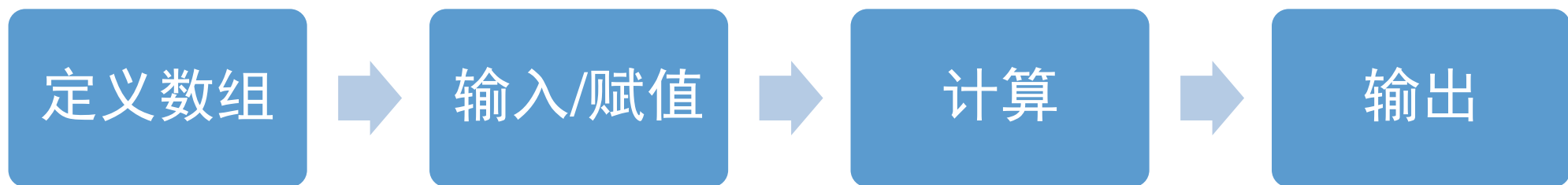
```
#include <stdio.h>
int main(){
    int i,a[5];
    static int c[5];
    for(i=0;i<5;i++)
        printf("%d  ",c[i]);
    printf("\n");

    return 0;
}
```

```
0 0 0 0 0
```



利用数组解决问题的一般解题思路



【例题3】利用数组，求斐波拉契数列的前10项。

- 第一项、第二项为1
- 前两项和为第三项



int fibo[10];
fibo[0] = fibo[1]=1; 或 int fibo[10]={1,1};



i 从2到9




fibo[i]=fibo[i-1]+fibo[i-2];

```
1 1 2 3 5 8 13 21 34 55
Press any key to continue
```

```
#include <stdio.h>
int main(){
    static int fibo[10]={1,1};
    int i;
    for(i=2;i<10;i++){
        fibo[i]=fibo[i-1]+fibo[i-2];
    }
    for(i=0;i<10;i++){
        printf("%d ",fibo[i]);
    }
    printf("\n");
    return 0;
}
```



【例题4】求若干个数(不多于10个)的最大数、最小数

- 不多于10个数  `#define N 10`
`float num[N];`
- 若干个数  `int n;`
- 最大数, 最小数  `float max,min;`

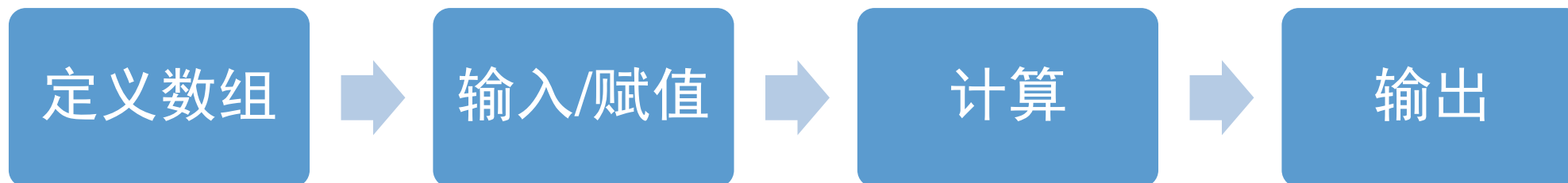
```
#define N 10
#include <stdio.h>
int main(){
    int i,n;
    float a[N],max,min;
    printf("input data numbers:");
    scanf("%d",&n);
    printf("enter numbers:");
    for(i=0;i<n;i++){
        scanf("%f",&a[i]);
    }
    max=min=a[0];
    for(i=1;i<n;i++){
        if(a[i]>max){
            max=a[i];
        }
        if(a[i]<min){
            min=a[i];
        }
    }
    printf("\nmax=%f,min=%f\n ",max,min);
}
```

```
input data numbers:8
enter numbers:8.7 79.4 4.7 -1.1 7.7 6.5 -7.9 500.0 3.6
max=500.000000,min=-7.900000
Press any key to continue
```



回到我们100个数的排序问题

1999



- 相邻两个数比较大小，把比较大的数放在后面
- 最大的那1个放在了正确的位置上。
- 剩下99个数

```
if(num[i-1]> num[i]){  
    t = num[i-1];  
    num[i-1] = num[i];  
    num[i] = t;  
}
```



输入100个整数，按照从小到大的顺序输出。

```
#include<stdio.h>
#define SIZE 100
int main(){
    int num[SIZE],t,i,count;

    printf("Please input 100 numbers:\n");
    for(i = 0; i<SIZE; i++){
        scanf("%d", &num[i]);
    }
    for(count = SIZE; count > 1; count--){
        for(i = 1; i < count; i++){
            if(num[i-1]> num[i]){
                t = num[i-1];
                num[i-1] = num[i];
                num[i] = t;
            }
        }
    }
    printf("Numbers in ascending order :\n");
    for(i = 0; i<SIZE; i++){
        printf("%-5d", num[i]);
        if(i%10==9)
            printf("\n");
    }
    return 0;
}
```

Please input 100 numbers:

```
6 28 22 233 164 244 124 63 223 285
52 43 265 249 36 195 169 201 292 141
25 172 254 36 290 267 131 133 79 158
26 228 67 9 152 286 202 34 4 76
98 295 251 94 10 243 183 244 116 246
215 158 62 36 147 16 65 73 232 204
213 187 23 174 246 76 144 36 82 100
26 253 227 134 14 64 61 198 51 48
77 268 152 289 202 283 255 33 248 259
231 247 186 118 162 94 273 297 52 269
```

Numbers in ascending order :

```
4    6    9    10   14   16   22   23   25   26
26   28   33   34   36   36   36   36   43   48
51   52   52   61   62   63   64   65   67   73
76   76   77   79   82   94   94   98   100  116
118  124  131  133  134  141  144  147  152  152
158  158  162  164  169  172  174  183  186  187
195  198  201  202  202  204  213  215  223  227
228  231  232  233  243  244  244  246  246  247
248  249  251  253  254  255  259  265  267  268
269  273  283  285  286  289  290  292  295  297
```

Press any key to continue



陕西科技大学

SHAANXI UNIVERSITY OF SCIENCE & TECHNOLOGY

【例题5】对10个人的考试成绩进行分段统计，考试成绩放在a数组中，各分数段的人数存到b数组中：成绩为60到69的人数存到b[0]中，成绩为70到79的人数存到b[1]，成绩为80到89的人数存到b[2]，成绩为90到99的人数存到b[3]，成绩为100的人数存到b[4]，成绩为60分以下的人数存到b[5]中。

```
#include <stdio.h>
int main(){
    int i,a[10];
    static int b[6]={0,0,0,0,0,0};
    printf("enter the score:\n");
    for (i=0; i<10; i++){
        scanf("%d", &a[i]);
        switch(a[i]/10){
            case 6:b[0]++;break;
            case 7:b[1]++;break;
            case 8:b[2]++;break;
            case 9:b[3]++;break;
            case 10:b[4]++;break;
            default:b[5]++;
        }
    }
    printf("the result is: ");
    for (i=0; i<6; i++)
        printf("%5d", b[i]);
    return 0;
}
```

```
enter the score:
100
98
70
88
87
85
70
35
66
81
the result is: 1 2 4 1 1 1Press any key to continue
```

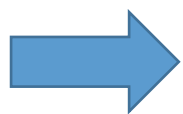


二维数组

```
int a[5];
```

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

数组元素为
整型变量



数组元素为
数组

aa[0]

aa[1]

aa[2]



二维数组的定义

1.定义方式:

类型说明符 数组名 [常量表达式1] [常量表达式2];

行数

列数

例: `float fa[3][4],fb[5][10];`



2. 说明

- 元素在内存排列顺序为按行存放;

```
int a[4][2];
```



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

二维数组在内存中的样子

- 多维数组定义亦类似于二维数组,如: **float a[2][3][4];**



二维数组的引用

数组名[行下标][列下标];

例：a[2][3]

数组元素是变量，因此一切
可以使用变量的地方都可以
使用数组元素

	0	1	2	3	4
0					
1					
2					
3					

int a[4][5];



二维数组初始化

1. 分行初始化。如：

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

2. 按数组排列的顺序将所有数据写在一个大括号内，
如：

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



3. 部分元素初始化。如：

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

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



4. 如果对全部元素都赋初值，则定义数组时可以不指定**第一维**的长度，如：

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

1	2	3	4
5	6	7	8
9	10	11	12



二维数组元素值的输入与输出

二维数组的输入与输出是通过对二维数组的每个元素的输入与输出实现的。

【例题6】二维数组输入与输出方法示例。

```
#include <stdio.h>
int main(){
    int row,col,a[3][4];
    printf("input array numbers:\n");
    for(row=0;row<3;row++)
        for(col=0;col<4;col++)
            scanf("%d",&a[row][col]);
    printf("output array numbers:\n");
    for(row=0;row<3;row++){
        for(col=0;col<4;col++)
            printf("%d  ",a[row][col]);
        printf("\n");
    }
    return 0;
}
```

```
input array numbers:
1 2 3 4
5 6 7 8
9 10 11 12
output array numbers:
1  2  3  4
5  6  7  8
9  10 11 12
Press any key to continue
```

}



陕西科技大学

SHAANXI UNIVERSITY OF SCIENCE & TECHNOLOGY

【例题7】在N行M列的二维数组x中，找出数组的最大值以及此最大值所在的行、列下标。

```
#define N 2
#define M 3
#include <stdio.h>
int main(){
    int i,j,x[N][M],max,line,col;
    printf("input array numbers:\n");
    for(i=0;i<N;i++)
        for(j=0;j<M;j++)
            scanf("%d",&x[i][j]);
    max=x[0][0];
    line=col=0;
    for(i=0;i<N;i++)
        for(j=0;j<M;j++)
            if(max<x[i][j]){
                max=x[i][j];
                line=i;
                col=j;
            }
    printf("\nmax=%d\tline=%d\tcol=%d\n",max,line,col);
}
```

```
input array numbers:
9 7 6
-8 82 49

max=82 line=1 col=1
Press any key to continue
```



字符数组与字符串

1. 字符数组的定义

- char 数组名[常量表达式];

2. 字符数组元素的引用

- 字符数组的每一个元素都是字符



3. 字符数组初始化

1) 逐个元素初始化，如：

```
static char c[10]={'h','a','p','p','y'};
```

2) 说明：

- 如果初值个数 > 数组长度，则作语法错误处理；
- 如果初值个数 < 数组长度，则只将这些字符赋给数组中前面那些元素，其余元素自动定为空字符（即' \0'）；
- 如果初值个数 = 数组长度，则在定义时可省略数组长度。



4. 用字符串来初始化字符数组

例如: `static char a[11] = { "I am a boy" };`

I		a	m		a		b	o	y	\0
---	--	---	---	--	---	--	---	---	---	----

例如: `char b[] = { "C program" };`

C		p	r	o	g	r	a	m	\0
---	--	---	---	---	---	---	---	---	----



字符数组与字符串的输入输出

假设有定义 `char a[10];`

- 1. 用格式符 “%c” 逐个字符输入输出:

```
scanf("%c",&a[0]);
```

```
printf("%c", a[0]); /*每次输入输出一个字符*/
```



2. 用格式符 “%s” 整个字符串输入输出：

```
scanf("%s", a);    /*注意此处用数组名a*/  
printf("%s", a);  /*注意此处用数组名a*/
```

用 “%s” 格式输入或输出字符数组时，数组名前不能再加 “&” 符号
数组名就是数组的起始地址。



用语句“scanf(“%s”, s); ” 空格键或回车键表示结束输入。
不包含空格键或回车键，而是在字符串末尾添加'\0'。

用一个scanf函数输入多个字符串以空格键或回车键作为字符串间的分隔。
%s的格式无法完整读入带有空格的字符串。

例如:

```
char s1[5], s2[5];
```

```
scanf("%s%s", s1, s2);
```

若输入数据:

C Prog ✓

则字符数组s1和s2的存储情况如下
图所示

C	\0			
P	r	o	g	\0



printf函数用"%s"格式输出字符数组时，则遇第一个'\0'时即结束输出。

例如：

```
char s3[5]={‘H’,’i’,’\0’,’C’,’\0’};
```

```
printf("%s", s3);
```

则输出为：

Hi

H	i	\0	C	\0
---	---	----	---	----



【例题8】用户从键盘输入一个字符串（字符中不包含空格），当输入回车时认为输入结束，统计输入字符串中小写英文字母、大写英文字母、数字字符、其他字符的个数。

```
#include "stdio.h"
int main(){
    int i,m,n,x,y;
    char s[80];
    printf("input a string:\n");
    scanf("%s",s);
    m=n=x=y=0;
    i=0;
    while(s[i]!='\0'){
        if(s[i]>='a'&& s[i]<='z') m++; //小写字母计数
        else if(s[i]>='A'&& s[i]<='Z') n++; //大写字母计数
        else if(s[i]>='0'&& s[i]<='9') x++; //数字计数
        else y++;
        i++;
    }
    printf("a~z:%d\nA~Z:%d\n0~9:%d\nothers:%d\n",m,n,x,y);
    return 0;
}
```

```
input a string:
abc1234!@#$%DEFGHI
a~z:3
A~Z:6
0~9:4
others:5
Press any key to continue
```



3.用字符串输入函数gets实现输入：

调用该函数时，应在程序中加入文件包含命令：`#include<stdio.h>`

❖函数原型：

`char *gets (char *s);`

以回车键作为输入结束标志；

❖调用形式：`gets(字符数组名)`； gets可以读入空格

❖举例：`char a[10];`

`gets(a);`

`printf(" %s" ,a);`



• 4. 用字符串输出函数puts实现输出

❖ 函数原型: `int puts (const char *s);`

❖ 调用形式: `puts(字符串/字符数组)`;

❖ 函数功能: 输出一个字符串。

❖ 应用举例: `char a[10];`

`gets(a);`

`puts(a);`

`puts("Hello C!");`



puts输出字符串后会再输出一个回车符



【例题9】将字符串s1从第m个字符开始剩余的所有字符，送入字符数组s2中。

```
#include "stdio.h"
int main(){
    int i,j,m;
    char s1[80],s2[80];
    printf("input a string:\n");
    gets(s1);
    printf("input start point:\n");
    scanf("%d",&m);
    i=m-1;
    j=0;
    while(s1[i]!='\0'){
        s2[j]=s1[i];
        i++;j++;
    }
    s2[j]='\0';
    puts(s2);
    return 0;
}
```

```
input a string:
This is a string.
input start point:
11
string.
Press any key to continue
```



❖何时自动添加 '\0'

1)以字符串初始化时

如: `static char c[]={"I am happy"};`

或: `static char c[]="I am happy";`

2) 以 %s输入字符串时

`scanf("%s",c);`

3)用 gets接收字符串时



字符串处理函数

字符串有专属操作

使用字符串处理函数时，应在程序中加入头文件包含命令：

```
#include<string.h>
```



赋值

1. 字符串复制函数

❖ 函数原型:

- `char *strcpy(char *dest, const char *src);`
目标 源

❖ 使用格式:

```
strcpy( dest, src);
```

❖ 函数功能:将字符串src复制到字符数组dest中，返回被复制的字符串。

- 注意: src可以是字符串常量、字符数组名或字符指针变量。
- dest只能是字符数组名或字符指针变量。
- dest 存储空间要大于src



❖ 函数应用举例:

- char a[10]="abcdefghi", b[]="happy";

a	b	c	d	e	f	g	h	i	\0
---	---	---	---	---	---	---	---	---	----

h	a	p	p	y	\0
---	---	---	---	---	----

strcpy(a, b);

h	a	p	p	y	\0	g	h	i	\0
---	---	---	---	---	----	---	---	---	----

h	a	p	p	y	\0
---	---	---	---	---	----



【例题10】分析下面程序的运行结果。

```
#include <string.h>
#include <stdio.h>
int main(){
    static char c1[10]="abcdefghi",c2[]="china";
    strcpy(c1,c2);
    puts(c1);
    printf("%c\n",c1[7]);
    return 0;
}
```

a	b	c	d	e	f	g	h	i	\0
---	---	---	---	---	---	---	---	---	----

c	h	i	n	a	\0
---	---	---	---	---	----



加法

2.字符串连接函数

❖函数原型:

- `char *strcat(char *dest, const char *src);`

目标

源

❖函数功能:

删去字符串dest的结束标志'\0'，将字符串或字符数组src连同末尾的结束标记一起连接到字符数组dest尾部，返回连接以后的字符串。

注意：src可以是字符串常量、字符数组名或字符指针变量。

dest只能是字符数组名或字符指针变量。

dest 存储空间要大于src



❖ 应用举例

- static char a[20]="My name is ", b[10]="Li ming";

M	y		n	a	m	e		i	s		\0	\0	\0	\0	\0	\0	\0	\0
---	---	--	---	---	---	---	--	---	---	--	----	----	----	----	----	----	----	----

L	i		m	i	n	g	\0	\0	\0
---	---	--	---	---	---	---	----	----	----

strcat(a,b);

M	y		n	a	m	e		i	s		\0	\0	\0	\0	\0	\0	\0	\0
---	---	--	---	---	---	---	--	---	---	--	----	----	----	----	----	----	----	----

L	i		m	i	n	g	\0	\0	\0
---	---	--	---	---	---	---	----	----	----



【例题11】字符串的连接

```
#include<string.h>
#include<stdio.h>
int main(){
    char s1[10]="123",s2[7]="abcdef";
    printf("连接前\n");
    printf("s1=");
    puts(s1);
    printf("s2=");
    puts(s2);

    strcat(s1,s2);

    printf("连接后\n");
    printf("s1=");
    puts(s1);
    printf("s2=");
    puts(s2);

    return 0;
}
```

```
连接前
s1=123
s2=abcdef
连接后
s1=123abcdef
s2=abcdef
Press any key to continue
```



计数

3. 计算字符串长度的函数

❖ 函数原型:

```
unsigned int strlen(const char *str);
```

❖ 函数功能:

求出字符串或字符数组中实际字符个数（**不包括**'\0'）。



【例题12】写出程序的运行结果。

```
#include <string.h>
#include <stdio.h>
int main(){
    static char st[20]={"hello\0abce\0abc\0"};
    puts(st);
    printf("%d\n",strlen(st));
    return 0;
}
```

```
hello
5
Press any key to continue
```

h	e	l	l	o	\0	a	b	c	e	\0	a	b	c	\0
---	---	---	---	---	----	---	---	---	---	----	---	---	---	----

字符数组：可以存储很多字符

字符串：只包含起始字符到字符串结束符'\0'之前的所有字符



关系比较：大于、小于和等于

4. 字符串比较函数

字符串的大小是如何定义的呢？

❖ 函数原型：

```
int strcmp(const char *s1, const char *s2);
```

❖ 函数功能：

从左到右逐个比较两个字符串s1、s2中各个字符（用字符的ASCII码比较），直到出现不同字符或遇到结束标记为止。



❖ 函数返回值:

函数值=

0	当 串1=串2
负整数	当 串1<串2
正整数	当 串1>串2

❖ 函数应用举例

- 例如: `strcmp(" d ", " abc ")` 结果为正整数。
- `strcmp(" dog ", " door ")` 结果为负整数。

对三个字符串进行排序, Let's go!



【例题13】定义三个字符串，按英文字母序排列后输出。

```
#include <string.h>
#include <stdio.h>
int main(){
    char s1[10]="China",s2[10]="America",s3[10]="Britain",t[10];
    if(num1 > num2){
        t = num1;
        num1 = num2;
        num2 = t;
    }

    puts(s1);
    puts(s2);
    puts(s3);
    return 0;
}
```

```
ica
Britain
China
Press any key to continue
```



6.5 数组程序设计举例

【例题14】判断一个字符串是否为回文串（回文串指正读反读都一样的字符串，如：字符串“abc121cba”）

```
#include <string.h>
#include <stdio.h>
int main(){
    char x[20];
    int i,j,n;
    gets(x);
    n=strlen(x);
    i=0;
    j=n-1;
    while(x[i]==x[j]&& i<j){
        i++;
        j--;
    }
    if(i>=j)
        printf("The number is palindrome");
    else
        printf("The number is not palindrome");
    return 0;
}
```

11223344

The number is not palindrome
Press any key to continue

112232211

The number is palindrome
Press any key to continue



【例题15】比较用户输入的字符串是否为表示同意的yes（用户可以用大写或小写）

```
#include <string.h>
#include <stdio.h>
int main(){
    int k;
    char s1[5],s2[]="yes";
    printf("input a string:\n");
    gets(s1);

    if(strcmp(s1,s2)!=0)
        printf("st1!=st2\n");
    else
        printf("st1=st2\n");

    return 0;
}
```

```
input a string:
Five
st1!=st2
Press any key to continue
```

```
input a string:
yes
st1=st2
Press any key to continue
```

```
input a string:
YES
st1=st2
Press any key to continue
```



【例题16】输入5个数，用“起泡法”对这5个数由小到大排序。

寻找数组中最大的数，放在最后的位置

在剩下的数中再寻找最大数，放置在剩余数的最后位置

所谓起泡法，是指按照上述思想，采用从第一个数开始，将相邻两数进行比较，把较大的数放在后面，再与下一个数进行比较，直至比较完所有的数的方式寻找最大值。



7

-9

13

6

20

第1次

-9

7

13

6

20

第2次

-9

7

13

6

20

第3次

-9

7

13

20

第4次

-9

7

6

13

20

第一趟结果

第1趟处理



-9

7

6

13

20

第1次

-9

7

6

13

20

第2次

-9

6

7

13

20

第3次

-9

6

7

13

20

第二趟结果

第2趟处理



第3趟处理

-9

6

7

13

20

第1次

-9

6

7

13

20

第2次

-9

6

7

13

20

第三趟结果

第4趟处理

-9

6

7

13

20

第1次

-9

6

7

13

20

第四趟结果



陕西科技大学

SHAANXI UNIVERSITY OF SCIENCE & TECHNOLOGY

用起泡法为N个数排序的情况分析表

第i趟	待排序数据个数	第i趟比较元素	j的范围[0, N-i-1]	两两比较次数
1	N	a[0]与a[1], a[1]与a[2]...a[N-2]与a[N-1]	[0, N-2]	N-1
2	N-1	a[0]与a[1], a[1]与a[2]...a[N-3]与a[N-2]	[0, N-3]	N-2
3	N-2	a[0]与a[1], a[1]与a[2]...a[N-4]与a[N-3]	[0, N-4]	N-3
...
k	N-k+1	a[0]与a[1], a[1]与a[2]...a[N-k-1]与a[N-k]	[0, N-k-1]	N-k
...
N-1	2	a[0]与a[1]	[0]	1

i的取值范围是[1, N-1]

j的取值范围是[0, N-i-1]



气泡法排序

```
#define N 5
#include <stdio.h>
int main(){
    int a[N],temp,i,j;
    printf("input numbers:\n");
    for(i=0;i<N;i++){
        scanf("%d",&a[i]);
    }
    for(i=1;i<N;i++){
        for(j=0;j<N-i;j++){
            if(a[j]>a[j+1]){
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    for(i=0;i<N;i++){
        printf("%d  ",a[i]);
    }
    return 0;
}
```

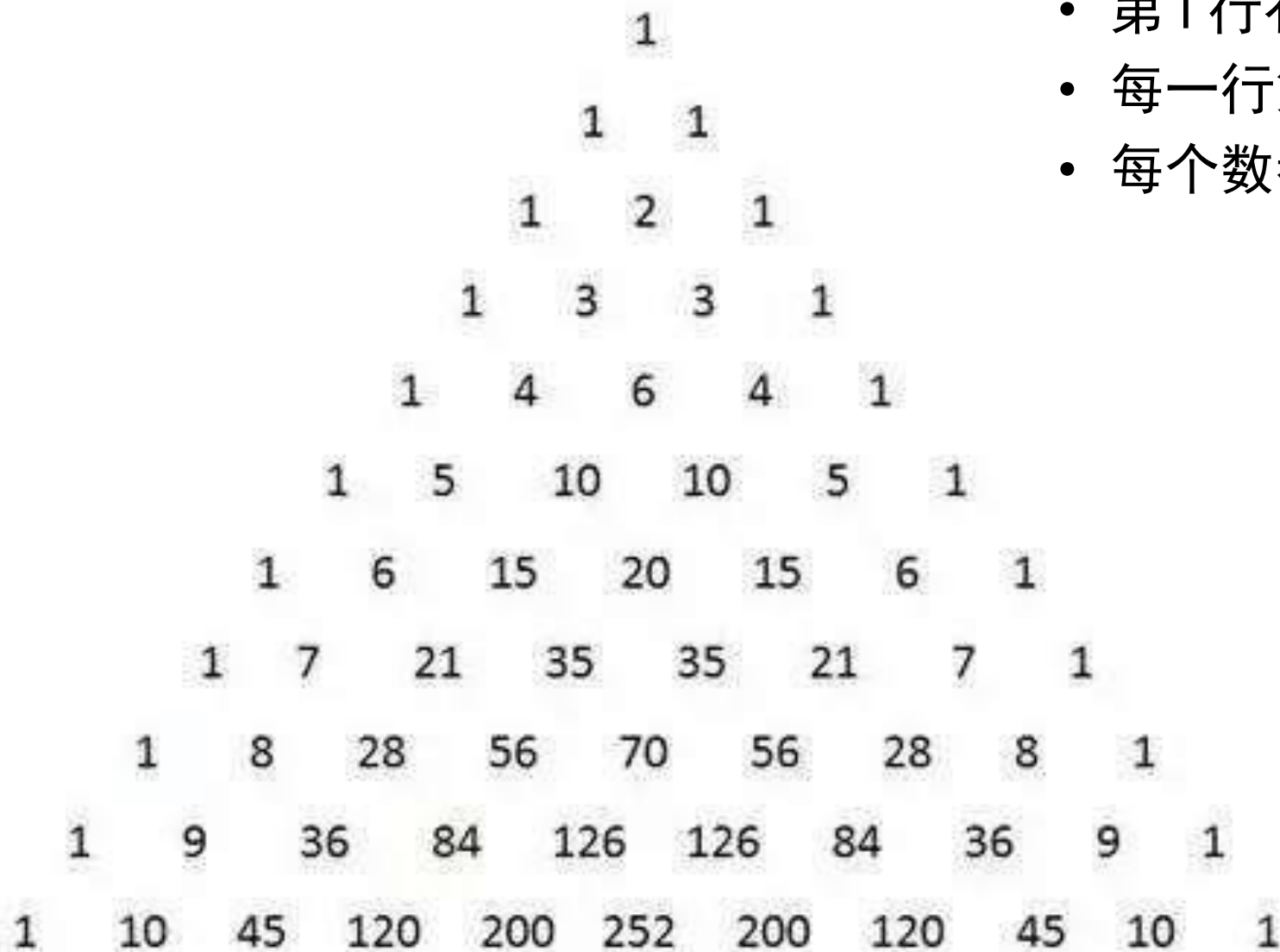
i 的取值范围是[1, N-1]

j 的取值范围是[0, N-1-i]

```
input numbers:
8 9 4 7 11
4 7 8 9 11 Press any key to continue
```



【例题17】输出杨辉三角形的前5行



- 第 i 行有 i 个数
- 每一行第一个数和最后一个数都是1
- 每个数都是上方两个数的和

【例题17】输出杨辉三角形的前5行

```
#define N 5
#include <stdio.h>
int main(){
    int i,j;

    int x[N][N];
    for(i=0;i<N;i++){
        for(j=0;j<=i;j++){
            if(j==0||i==j){
                x[i][j]=1;
            }
            else{
                x[i][j]=x[i-1][j]+x[i-1][j-1];
            }
        }
    }
    for(i=0;i<N;i++){
        for(j=0;j<=i;j++){
            printf("%d ",x[i][j]);
        }
        printf("\n");
    }
}
```

```
1
1 1
第2行有i个数
1 3 3 1
每一行第一个数和最后一个数都是1
1 4 6 4 1
每行的第j个数都是上一行第j-1个数
Press any key to continue
和第j个数的和
```

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
Press any key to continue
```



【例18】输入五个人的姓名，按字母顺序排列输出。

```
#include <string.h>
#include <stdio.h>
int main(){
    char st[20]          ;
```

```
    return 0;
```

```
}
```

```
input people's name:
```

```
Perch
```

```
Soucy
```

```
George
```

```
Pedro
```

```
Rebecca
```

```
George
```

```
Pedro
```

```
Perch
```

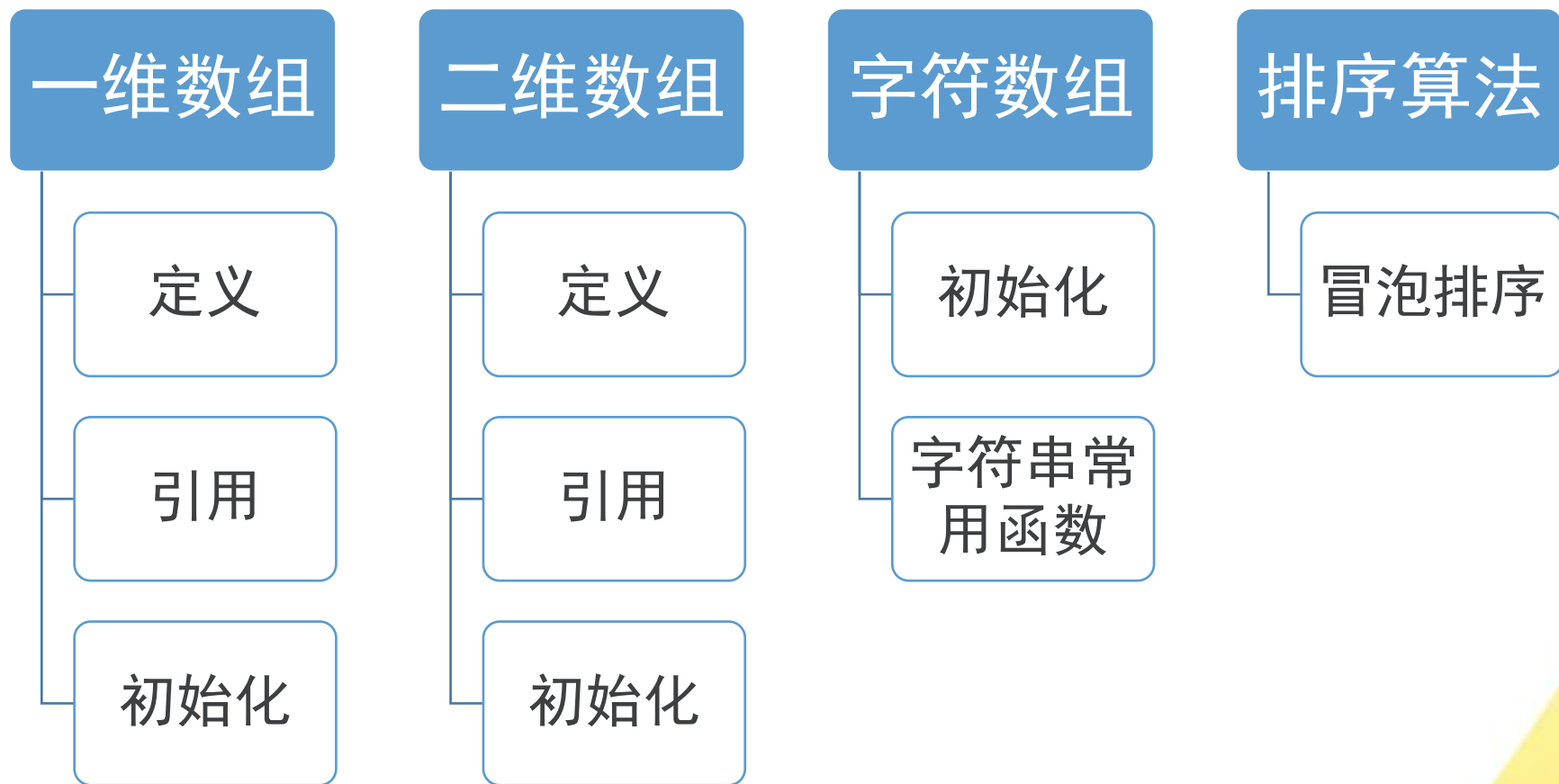
```
Rebecca
```

```
Soucy
```

```
Press any key to continue
```



总结



6.4 数组与函数

6.4.1 数组元素做函数参数

❖ 使用时与普通变量一样

❖ 值传递方式

实参

在函数调用时将值传递给

形参



【例6-16】一个数组中有10个整型元素，求数组中所有素数之和。

```
#include "stdio.h"
#include "math.h"
main()
{int a[10], i, sum=0;
 int prime(int x);
 printf("enter 10 numbers:\n");
 for(i=0; i<10; i++)
 {scanf("%d", &a[i]);
  if(prime(a[i])) sum+=a[i];
 }
 printf("sum=%d\n", sum);
}
```

```
int prime(int x)
{int f=1, k;
 if(x==1) f=0;
 for(k=2; k<=sqrt(x); k++)
 if(x%k==0) {f=0; break; }
 return(f);
}
```



6.4.2 数组名做函数参数

地址传递方式, 实参与形参共地址, 相当于对同一对象的不同名字。

实参

在函数调用时将其地址值传递给

形参



【例6-17】分析下面程序的运行结果。

```
#include "stdio.h"

int f(int b[], int m, int n)
{ int i, s=0;
  for(i=m; i<n; i=i+2) s=s+b[i];
  return s;
}

main()
{ int x, a[]={1, 2, 3, 4, 5, 6, 7, 8, 9};
  x=f(a, 3, 7);
  printf("%d\n", x);
}
```



【例6-11】 实参为字符指针的复制情况。

```
#include <string.h>
main()
{ char *s1="123",*s2="abcedf",*s3;
  s3=strcpy(s1,s2);
  printf("%u,%u,%u\n",s1,s2,s3);
  puts(s1);
  puts(s2);
  puts(s3);
}
```



THANKYOU

