



清华大学

腾讯会议 ID: 543 395 962/ 2021
Tsinghua University

计算机程序设计基础

第6讲 数组

沈瑜 (010-62782951)

shenyu@tsinghua.edu.cn

清华大学电机系

2021.10.19





主要内容

- 数组C程序设计举例
- 一维数组
 - 一维数组的定义与初始化
 - 一维数组的引用与基本操作
- 字符数组与字符串
 - 字符数组的定义与初始化
 - 字符数组的输入输出
 - 常用字符串函数

《C程序设计（第五版）》参考教材第6章

《C语言程序设计:现代方法(第2版)》第8章





7.1 数组C程序设计举例

● 例1: 幸运字母程序（采用数组实现）

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char name[100]; // 字符数组
6     int len = 0; // 定义姓名拼音字符数变量
7     int sum = 0; // 定义字符累加变量
8
9     printf("请输入您的姓名拼音:");
10    scanf("%s", name); // 输入姓名拼音 (不含空格)
11
12    len = strlen(name); // 求姓名拼音字符数
13    for(int i=0; i<len; i++) // 大写转换为小写
14        if(name[i] >= 'A' && name[i] <= 'Z')
15            name[i] = name[i] - 'A' + 'a';
16
17    for(int i=0; i<len; i++) // 字符加和
18        sum += name[i];
19
20    printf("您的姓名包含%d个字符, 幸运字母是%c\n", len, sum/len);
21    return 0;
22 }
```

需要用到字符串函数

C:\wi
请输入您的姓名拼音: ShenYu
您的姓名包含6个字符, 幸运字母是o
请按任意键继续. . .



●例2:成绩分级程序（采用数组实现）

```
1 #include <stdio.h>
2 int main()
3 {
4     int score;//成绩
5     int score_line[4]={93,85,77,70}; //成绩分级线
6     char grade[5]={'A','B','C','D','F'}; //成绩分级代码
7     int index=0; //成绩分级号, 初始对应'A'
8
9     printf("你好! 你的《计算机程序设计基础》成绩预计得多少分? (请输入0-100整数) \n ");
10    scanf( "%d", &score );    //输入成绩
11
12    if(score>=0 && score<=100)
13    {
14        for(int i=0;i<4;i++)
15            if(score<score_line[i])
16                index=i+1; //例如判断score<93, 则暂时对应'B';以此类推
17        printf("你的预计成绩等级为: %c \n", grade[index]); //输出等级
18    }
19    else
20        printf("你的预计成绩输入有误! " ); //输出等级
21
22    return 0;
23 }
```

长度为4整数数组

长度为5的字符数组

```
C:\Windows\system32\cmd.exe
你好! 你的《计算机程序设计基础》成绩预计得多少分? (请输入0-100整数)
85
你的预计成绩等级为: B
请按任意键继续. . . .
```





●例3:统计阶乘（采用数组实现）

```
1 #include <stdio.h>
2 int main()
3 {
4     int num; //最大阶乘的基数
5     int fac[11]; //阶乘结果的数组
6     printf("请输入一个不大于10的整数:");
7     scanf("%d",&num);
8
9     if( num>10 || num<0 ) //重要:判断输入数据的正确性
10    {
11        printf("输入有误!\n");
12        return -1;
13    }
14
15    fac[0] = 1;
16    for(int i=1;i<=num;i++)//求各次阶乘
17    {
18        fac[i] = fac[i-1] * i;
19    }
20
21    for(int i=0;i<=num;i++)
22    {
23        printf("%d!=%d\n",i,fac[i]); //输出各次阶乘
24    }
25
26    return 0;
27 }
```

长度为11的整数数组

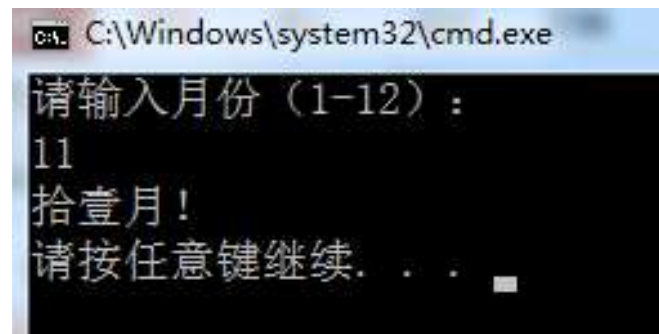
思考:
若不检查输入数据,
当输入-1时会怎样?
当输入100时又会怎样?

要点: 数组下标一定不能越界!



● 例4: 翻译月份（采用数组实现）

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     int nMonth;
6     printf("请输入月份 ( 1-12 ) : \n");
7     scanf("%d", &nMonth);
8     char month[12][20]={"壹月","贰月","叁月","肆月",
9                          "伍月","陆月","柒月","捌月",
10                         "玖月","拾月","拾壹月","拾贰月"};
11     if(nMonth>=1 && nMonth<=12 )
12         printf("%s ! \n",month[nMonth-1]);
13     else
14         printf("我不懂是几月啊 ? ! \n");
15
16     return 0;
17 }
```



6.2 一维数组

□ 数组的定义

按序排列的同类数据元素的集合称为数组 **array**

□ 数组的分类

- 按数组的类型分，有数值数组、字符数组等
- 按数组的维数分，有一维数组、二维数组等

只有一个下标变量的数组，称为一维数组



1. 一维数组定义

一般形式为:

类型符 **数组名** [**常量表达式**],;

- **类型符**可为任意基本数据类型
- **数组名**是用户定义的数组标识符
- **常量表达式**表示数据元素的个数，也称数组长度

例如:

int a[10]; //定义整型数组**a**，有**10**个元素

float b[10],c[20]; //定义实型数组**b**和**c**

char c[20]; //定义字符数组**c**



说明:

(1) 对同一个数组, 其所有元素的**数据类型**相同

(2) **数组名**不能与同一函数中其它变量同名

例如:

```
int a;
```

```
float a[10]; //是错误的
```

(3) **C语言**中规定数组的**下标从0开始**

例如:

```
int a[5] ; //5个元素分别为a[0],a[1],a[2],a[3],a[4]
```



(4) 不能在方括号中用变量来表示元素的个数，但是可以是符号常数或常量表达式。

合法的定义方式：

```
#define N 5  
int a[3+2],b[N+7];
```

错误的定义方式：

```
int n=5;  
int a[n];
```

(5) 数组中各元素在内存中按下标顺序连续存储



思考：

- 数组定义与变量定义有何区别？
- 形式上
 - 数组定义除指明变量类型和变量名外，还需指明长度
 - `int var;`
 - `int a[10];`
- 作用上
 - 变量定义：为单个数据分配空间
 - 数组定义：为多个数据分配空间



2.一维数组的初始化

(1) 数组的初始化是指在数组定义时给数组元素赋初值

(2) 数组初始化的一般形式为:

类型符 数组名[常量表达式]={值, 值.....值};

其中: 在{ }中的各数据值即为各元素的初值, 各值之间用逗号间隔。

例如:

```
int a[10]={ 0,1,2,3,4,5,6,7,8,9 };
```

相当于

```
a[0]=0; a[1]=1;... a[9]=9;
```



C 语言对数组初始化的几点规定：

(1) 可以只给部分元素赋初值。当{ }中值的个数少于元素个数时，只给前面部分元素赋值。

例如： **int a[10]={0,1,2,3,4};**

//表示给前5个元素赋值，后5个元素自动赋0值

(2) 只能给元素逐个赋值，不能给数组整体赋值。

例如：将数组的10个元素全部赋值为1

int a[10]={1,1,1,1,1,1,1,1,1,1};//正确

int a[10]=1;//错误



(3) 若给全部元素赋值，则可不写明数组长度

例如: `int a[5]={1,2,3,4,5};`

也可写为:

`int a[]={1,2,3,4,5};` ✓

不写明长度则必须初始化:

`int a[];` ✗

(4) {}中值的个数不可多于元素个数

例如:

`int a[5]={1,2,3,4,5,1 };` ✗



3.数组元素的引用

数组元素一般引用形式为：**数组名[下标]**

其中的**下标**只能为**整型常量**或**整型表达式**

例如：

```
int a[5], b[5];
```

```
int i=2, int j=3;
```

下标为整型常量或整型表达式：

a[i]=b[j]; //将数组b的第**4**个元素赋值给a的第**3**个元素

a[i+1]=b[j]; /将数组b的第**4**元素赋值给数组b的第**4**元素



注意:

(1) 数组的定义和数组元素中下标的差异

- 数组定义中的下标表示数组长度，即元素个数；
- 数组元素中的下标表示该元素在数组中的索引
- 前者只能是常量，后者可以是常量、变量或表达式

例如:

```
int a[3]; // 定义一个长度为3的数组
```

```
a[2] = 1; // 将数组a的第3个元素赋值为1
```

```
a[3] = 1; // 越界错误，数组a最后一个元素为a[2]
```



(2) 只能通过下标变量引用数组元素，而不能一次引用整个数组

例如：将数组**b**的各元素值依次赋给数组**a**的对应元素
可以用**for**循环逐一赋值

```
int a[10], b[10];  
for(i=0; i<10; i++)  
    a[i] = b[i];
```

不能写成如下形式：

a=b



6.3 字符数组与字符串

1. 字符数组的定义和初始化

□ 字符数组的定义：与数值数组相似，只是类型不同

例如： **char c[10];** //定义有10个字符元素的数组

□ 字符数组的初始化：与数值数组相似，但当初始化数据少于数组长度时，其余的元素为“空”（'\0'）

例如： **char c[10]={ 'C', ' ', 'P', 'r', 'o', 'g', 'R', 'a', 'm', '\0' };**

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]	c[9]
C		P	r	o	g	R	a	m	\0

c[9]，由系统自动初始化为空字符 '\0'



说明:

(1) 字符数组可以用字符串来初始化

例如:

```
char c[]={'C', ' ', 'P','r','o','g', 'r','a','m', '\0'};
```

可写为:

```
char c[]={"C Program"};
```

或去掉{ }写为:

```
char c[]="C Program";
```



(2) 字符数组**不能**用字符串赋值！

例如：

```
char c[10];
```

不能用：

```
c="C Program";
```

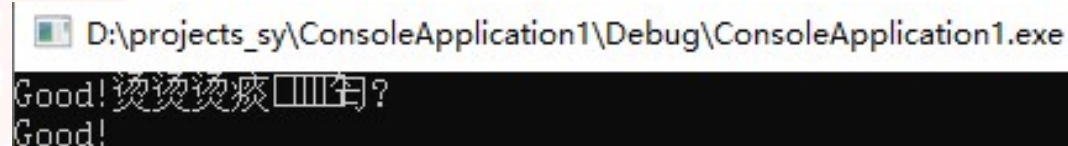


(3) 字符串在存储时，系统自动在其后加上结束标志‘\0’（占1字节），但字符数组最后一个元素不一定是‘\0’

例如：

```
#include "stdio.h"
int main()
{
    char c1[5]={'G','o','o','d','!'};
    char c2[]={"Good!"};
    printf("%s\n",c1);
    printf("%s\n",c2);
    return 0;
}
```

Good! 烫烫烫烫



D:\projects_sy\ConsoleApplication1\Debug\ConsoleApplication1.exe
Good! 烫烫烫烫
Good!

c1与c2并不等价，字符数组c1不能当字符串使用，因为其最后一个元素不是结束标志‘\0’。

字符串与字符数组的比较

在 C 语言中没有专门的字符串变量，通常用一个字符数组来存放一个字符串。

字符串总是以 ‘\0’ 作为串的结束符。因此当把一个字符串存入一个数组时，也把结束符 ‘\0’ 存入数组，并以此作为该字符串是否结束的标志。有了 ‘\0’ 标志后，就不必再用字符数组的长度来判断字符串的长度了。

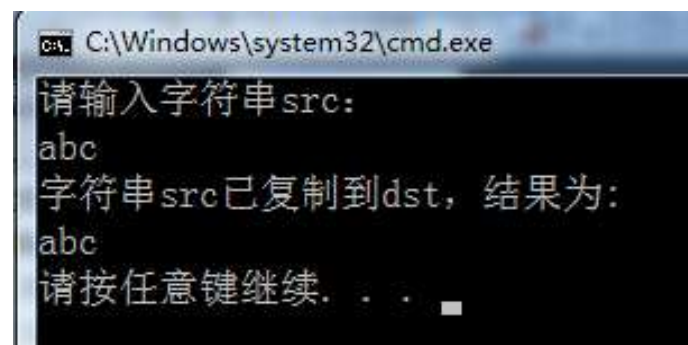
实际上，字符串就是一种字符型数组，并且这个数组的最后一个单元是一个字符结束标志 ‘\0’

字符串是一种以 ‘\0’ 结尾的字符数组！



例：字符数组的复制（strcpy函数的核心）

```
1 #include <stdio.h>
2 #define N 100
3
4 int main()
5 {
6     char src[N] = ""; //初始化为空字符串, src[0]存'\0'
7     char dst[N] = ""; //初始化为空字符串, dst[0]存'\0'
8
9     printf("请输入字符串src: \n");
10    scanf("%s", src); //不能含空格, 否则空格后的无效。如果想输入空格, 后续想办法
11
12    for(int i=0; i<N; i++)
13    {
14        if(src[i] == '\0')
15            break; //遇字符串结尾则处理结束
16        else
17            dst[i] = src[i];
18    }
19    printf("字符串src已复制到dst, 结果为: \n%s\n", dst);
20    return 0;
21 }
```

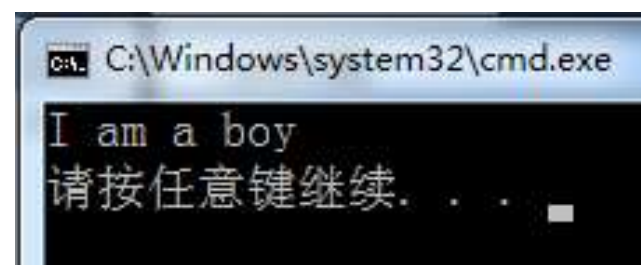


```
C:\Windows\system32\cmd.exe
请输入字符串src:
abc
字符串src已复制到dst, 结果为:
abc
请按任意键继续. . .
```

2. 字符数组的引用

引用字符数组一个元素，得到一个字符。其引用形式与数值数组相同

```
1 #include <stdio.h>
2 int main()
3 {
4     char c[10]={'I',' ','a','m',' ','a',' ','b','o','y'};
5     int i;
6     for(i=0;i<10;i++)
7         printf("%c",c[i]);
8     printf("\n");
9     return 0;
10 }
```



字符数组，未必是字符串哦~~~

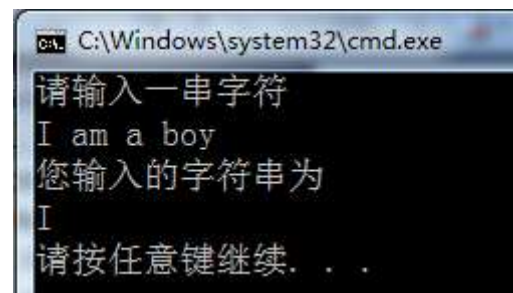
3.字符数组的输入输出

两种方式

- 用“%c”格式符逐个输入输出
- 用“%s”格式符按字符串输入输出

```
1 #include "stdio.h"
2 int main()
3 {
4     char c[100];
5     printf("请输入一串字符\n");
6     scanf("%s",c);
7     printf("您输入的字符串为\n%s\n",c);
8     return 0;
9 }
```

输入 “I am a boy”
输出是什么？



```
C:\Windows\system32\cmd.exe
请输入一串字符
I am a boy
您输入的字符串为
I
请按任意键继续. . .
```

说明:

- (1) 字符数组输出时遇'\0'结束, 且输出不包含'\0'。
- (2) “%s”格式输出字符串时, **printf()**函数的输出项是字符数组名, 而不是元素名。

例:

```
char c[] = "Good!";  
printf("%s", c); printf("%c", c[0]);  
printf("%s", c[0]);    /* 错误 */
```



(3) “%s”格式输出时，即使数组长度大于字符串长度，遇'\0'也结束。

例：

```
char c[10] = {"Good!"};  
printf("%s",c); /*只输出5个字符 */
```

(4) “%s”格式输出时，若数组中包含一个以上'\0'，遇第一个'\0'时结束。

例：

```
char c[] = {"Good!\0boy"};  
printf("%s",c); /*输出结果是： Good! */
```



(5) 一个scanf函数输入多个字符串，输入时以“空格”键作为字符串间的分隔（还记得 “I am a boy ” 吗？）
例如：

```
char str1[5],str2[5],str3[5];  
scanf("%s%s%s",str1,str2,str3);
```

输入数据： **How are you?**

str1、**str2**、**str3**获得的数据如下：

H	o	w	\0	
a	r	e	\0	
y	o	u	?	\0



(6) C语言中，数组名代表该数组的起始地址，因此，scanf()函数中不需要地址运算符&。

例：

```
char str[13];  
scanf("%s",str);  
scanf("%s",&str); /* 慎用，可能编译不通过*/
```

参考资料：对字符数组输入输出数组名是否加取地址符的一些理解

https://blog.csdn.net/qq_41950229/article/details/83515267



4.常用字符串函数

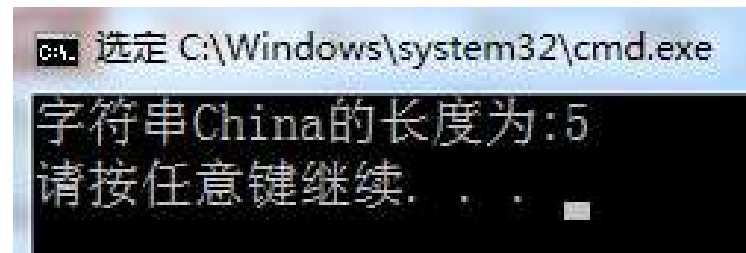
使用字符串函数应包含头文件"**string.h**"

(1) 测字符串长度函数**strlen**

格式: **strlen(str)**

功能: 测字符串的**实际长度**(不含字符串结束标志‘\0’) 并作为函数返回值。

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     int len;
6     char src[] = "China";
7     len = strlen(src);
8     printf("字符串%s的长度为:%d\n", src, len);
9     return 0;
10 }
```



cmd. 选定 C:\Windows\system32\cmd.exe
字符串China的长度为:5
请按任意键继续. . .

(2) 字符串输出函数 puts

格式: **puts (str);**

- 功能: 把字符数组中的字符串输出到显示器。即在屏幕上显示。
- 等价于: **printf("%s\n",str);**

(3) 字符串输入函数 gets

格式: **gets (str);**

- 功能: 从键盘上输入一个字符串直到回车键结束。将输入的字符串加上结束标志 ‘\0’, 存放到数组str。

(4) 字符串连接函数**strcat**

格式: **strcat (str1, str2)**

功能: 将字符数组**str2**中的字符串连接到**str1** 中字符串的后面, 并删去字符串1后的串标志 “\0”。

(5) 字符串拷贝函数**strcpy**

格式: **strcpy (str1, str2)**

功能: 把字符数组**str2**中的字符串拷贝到**str1**中。串结束标志 “\0”也一同拷贝。

(6) 字符大写转小写函数**strlwr**

格式: **strlwr(str)**

功能: 将字符串**str**中的大写字母转换为小写字母。



(7) 字符小写转大写函数**strupr**

格式: **strupr(str)**

功能: 将字符串**str**中的小写字母转换为大写字母。

(8) 字符串比较函数**strcmp**

格式: **strcmp(str1, str2)**

功能: 按照**ASCII**码顺序比较两个数组中的字符串, 并由函数返回值返回比较结果。

“<0” 表示 $s1 < s2$; 返回值 -1

“=0” 表示 $s1 = s2$; 返回值 0

“>0” 表示 $s1 > s2$ 。 返回值 1

若 $s1$ 与 $s2$ 的前 i 个字符一致, 则比较第 $i+1$ 个字符的大小。



设有数组定义：

```
char a[] = "china";
```

则a所占的空间大小为：

- ☐ A 4字节
- ☐ B 5字节
- ☒ C 6字节
- ☐ D 7字节

提交

6.4 综合示例：冒泡排序算法

问题

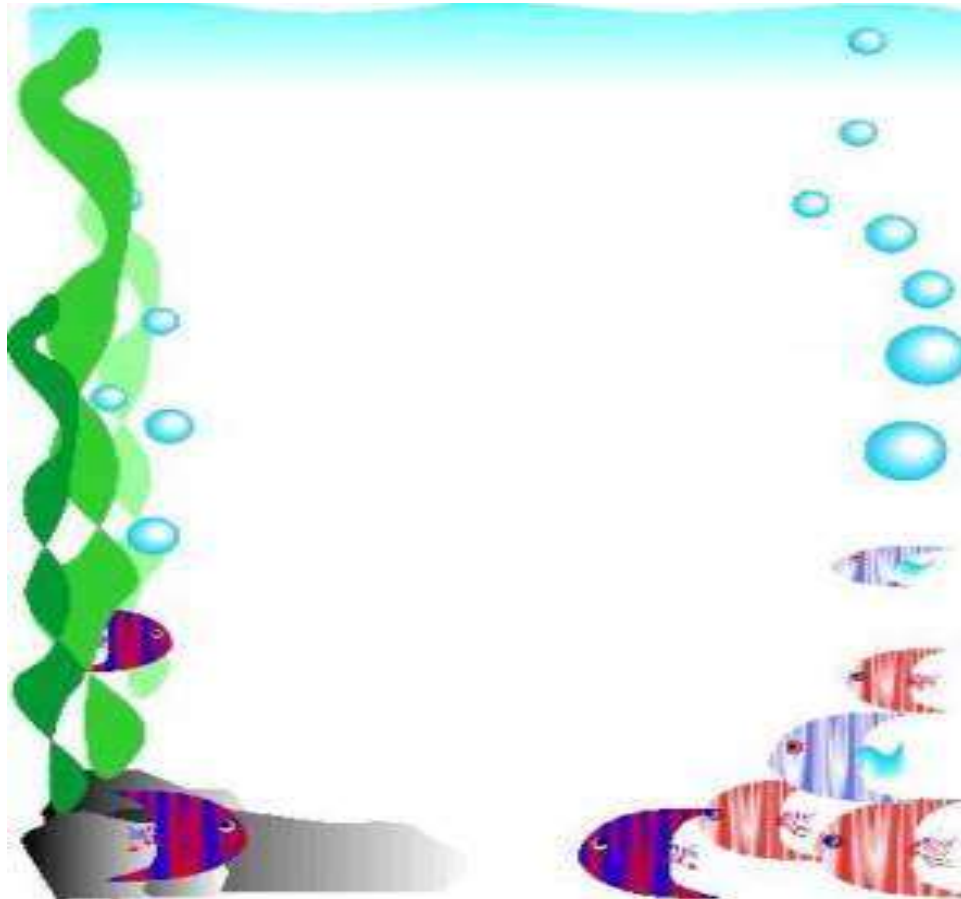
用冒泡排序的方法将下面一组无序数组排成从小到大的顺序。

{ 9, 8, 5, 2, 4, 0 }

{ 9, 8, 5, 4, 2, 0 }

{ 0, 2, 4, 5, 8, 9 }

1.冒泡排序原理



气泡大的在下面
气泡小的在上面



冒泡排序原理

数大的在下面
数小的在上面

9

8

5

4

2

0

- ◆ 对存放原始数据的数组，按从前往后（从上到下）的方向进行多次扫描，每次扫描称为一趟。
- ◆ 当发现相邻两个数据的次序与排序要求的大小次序不符合时，即将这两个数据进行互换。
- ◆ 这样，较小的数据就会逐个向前移动，好象气泡向上浮起一样。

```
for(i=0;i<5;i++)  
    if (a[i]>a[i+1])  
        { t=a[i];a[i]=a[i+1];a[i+1]=t; }
```

a[0]	9	8	8	8	8	8
a[1]	8	9	5	5	5	5
a[2]	5	5	9	4	4	4
a[3]	4	4	4	9	2	2
a[4]	2	2	2	2	9	0
a[5]	0	0	0	0	0	9

大数沉淀，小数起泡

```
for(i=0;i<4;i++)  
    if (a[i]>a[i+1])  
    { t=a[i];a[i]=a[i+1];a[i+1]=t; }
```

a[0]	8	5	5	5	5
a[1]	5	8	4	4	4
a[2]	4	4	8	2	2
a[3]	2	2	2	8	0
a[4]	0	0	0	0	8
a[5]	9	9	9	9	9

```
for(i=0;i<3;i++)  
    if (a[i]>a[i+1])  
    { t=a[i];a[i]=a[i+1];a[i+1]=t; }
```

a[0]	5	4	4	4
a[1]	4	5	2	2
a[2]	2	2	5	0
a[3]	0	0	0	5
a[4]	8	8	8	8
a[5]	9	9	9	9

```
for(i=0;i<2;i++)  
  if (a[i]>a[i+1])  
  { t=a[i];a[i]=a[i+1];a[i+1]=t; }
```

a[0]	4	2	2
a[1]	2	4	0
a[2]	0	0	4
a[3]	5	5	5
a[4]	8	8	8
a[5]	9	9	9


```
for(i=0;i<1;i++)  
    if (a[i]>a[i+1])  
    { t=a[i];a[i]=a[i+1];a[i+1]=t; }
```

a[0]

a[1]

a[2]

a[3]

a[4]

a[5]

2

0

0

2

4

4

5

5

8

8

9

9

```
for(i=0;i<5;i++)  
    if (a[i]>a[i+1])  
    { .....}
```

```
for(i=0;i<4;i++)  
    if (a[i]>a[i+1])  
    { .....}
```

.....

```
for(i=0;i<1;i++)  
    if (a[i]>a[i+1])  
    { .....}
```

```
for(j=0;j<5;j++)
```

```
    for(i=0;i<5-j;i++)  
        if (a[i]>a[i+1])  
        { .....}
```

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

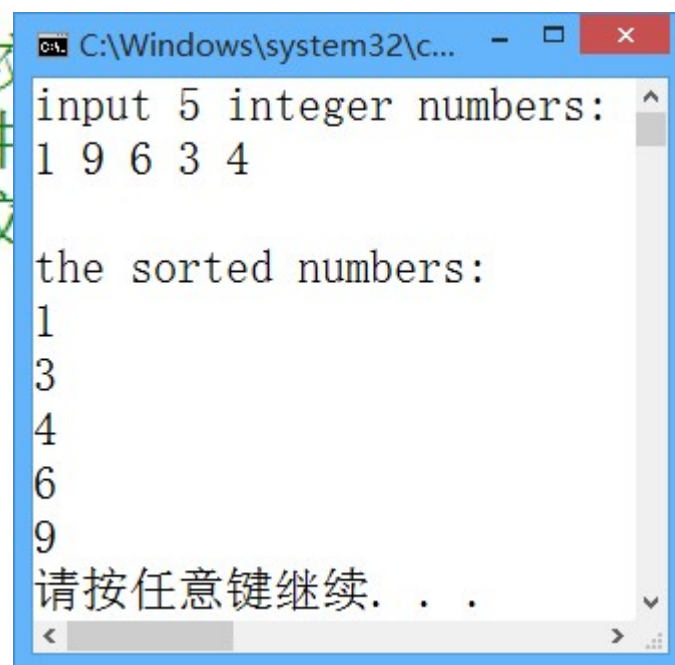
```
    for(i=0;i<6-j;i++)  
        if (a[i]>a[i+1])  
        { .....}
```



```

#define N 5
int a[100]; /* 定义大一些, 但只用前N个元素 */
int i, j;
printf("input %d integer numbers:\n", N);
for(i=0; i<N; i++)
    scanf("%d", &a[i]); /* 输入N个整数 */
printf("\n");
for(j=1; j<N; j++) /* 第j趟比较
    for(i=0; i<N-j; i++) /* 第j趟中
        if (a[i] > a[i+1]) /* 交
        {
            int temp;
            temp = a[i];
            a[i] = a[i+1];
            a[i+1] = temp;
        };

```



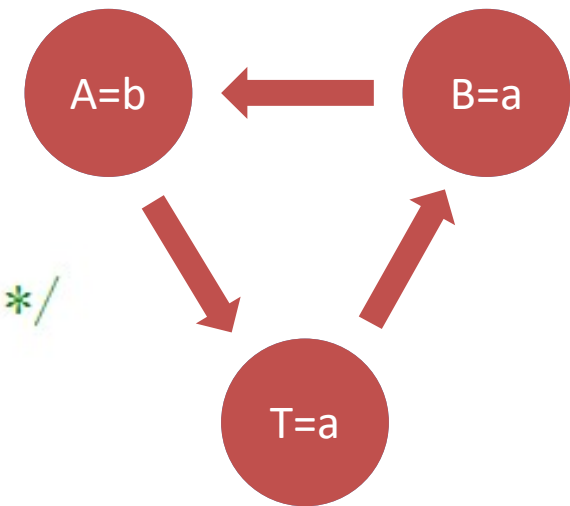
```

C:\Windows\system32\c...
input 5 integer numbers:
1 9 6 3 4
the sorted numbers:
1
3
4
6
9
请按任意键继续. . .

```

*算法的艺术

```
#define N 5
int a[100]; /* 定义大一些, 但只用前N个元素 */
int i, j;
printf("input %d integer numbers:\n", N);
for(i=0; i<N; i++)
    scanf("%d", &a[i]); /* 输入N个整数 */
printf("\n");
for(j=1; j<N; j++) /* 第j趟比较 */
    for(i=0; i<N-j; i++) /* 第j趟中两两比较N-j次 */
        if (a[i] > a[i+1]) /* 交换大小 */
        {
            int temp;
            temp = a[i];
            a[i] = a[i+1];
            a[i+1] = temp;
        };
```



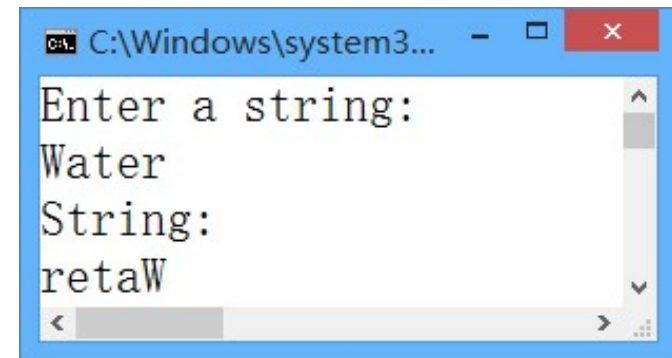
思考:
temp的作用?
是否可以不用temp?



*算法的艺术

- 思考下面程序的实现功能

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[80];
    int temp, i, j;
    printf("Enter a string:\n");
    scanf("%s", str);
    for(i=0, j=strlen(str)-1; i<j; i++, j--)
    {
        temp=str[i];
        str[i]=str[j];
        str[j]=temp;
    }
    printf("\String: \n%s\n", str);
    return 0;
}
```



单选题 1分



```
char str[]="SSSWLIA",c; int k;  
for(int k=2;((c=str[k])!='\0');k++){  
    switch(c)  
    {  
        case 'l': ++k;break;  
        case 'L':continue;  
        default: putchar(c);continue;  
    }  
    putchar('*');  
}
```

运行的结果是？

break 语句跳出switch而不是for循环

- ☐ A SSW ☒ B SW* ☐ C SW*A ☐ D SW

提交