

使用字符串处理函数

输出字符串的函数

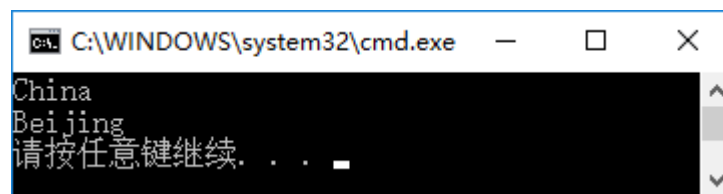
puts(字符数组)

作用：将一个字符串(以'\0'结束的字符序列)输出到终端。

用puts函数输出的字符串中可以包含转义字符。

在用puts输出时将字符串结束标志'\0'转换成'\n'，即输出完字符串后换行。

```
#include <stdio.h>
int main()
{
    char str[]={"China\nBeijing"};
    puts(str);
    return 0;
}
```



输入字符串的函数

gets(字符数组)

作用：从终端输入一个字符串到字符数组，并且得到一个函数值。该函数值是字符数组的起始地址。

```
gets(str);           //str是已定义的字符数组
```

如果从键盘输入：

Computer ↙

将输入的字符串“Computer”送给字符数组str（请注意，送给数组的共有9个字符，而不是8个字符），返回的函数值是字符数组str的第一个元素的地址。

注意

- 用puts和gets函数只能输出或输入一个字符串。



```
puts(str1, str2); 或 gets(str1, str2);
```

字符串连接函数

`strcat(字符串1, 字符串2)`

作用：把两个字符数组中的字符串连接起来，把字符串2接到字符串1的后面，结果放在字符数组1中，函数调用后得到一个函数值——字符数组1的地址。

字符数组1必须足够大，以便容纳连接后的新字符串。

连接前两个字符串的后面都有'\0'，连接时将字符串1后面的'\0'取消，只在新串最后保留'\0'。

```
char str1[30]={"People's Republic of "};  
char str2[]={"China"};  
printf("%s", strcat(str1, str2));
```

输出：People's Republic of China

连接前
str1:

P	e	o	p	l	e	'	s		R	e	p	u	b	l	i	c		o	f		\0	\0	\0	\0	\0	\0	\0	\0	\0
---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	--	---	---	--	----	----	----	----	----	----	----	----	----

str2:

C	h	i	n	a	
---	---	---	---	---	--

连接后
str1:

P	e	o	p	l	e	'	s		R	e	p	u	b	l	i	c		o	f		C	h	i	n	a	\0	\0	\0	\0
---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	--	---	---	--	---	---	---	---	---	----	----	----	----

字符串复制函数

strcpy(字符数组1, 字符串2)

```
char str1[10], str2[]="China";  
strcpy(str1, str2); 或 strcpy(str1, "China");
```

执行后, str1:

C	h	i	n	a	\0	\0	\0	\0	\0
---	---	---	---	---	----	----	----	----	----

作用：将字符串2复制到字符数组1中去。

字符数组1必须定义得足够大，以便容纳被复制的字符串2。字符数组1的长度不应小于字符串2的长度。

“字符数组1”必须写成数组名形式，“字符串2”可以是字符数组名，也可以是一个字符串常量。

若在复制前未对字符数组1初始化或赋值，则其各字节中的内容无法预知，复制时将字符串2和其后的‘\0’一起复制到字符数组1中，取代字符数组1中前面的字符，未被取代的字符保持原有内容。

不能用赋值语句将一个字符串常量或字符数组直接给一个字符数组。字符数组名是一个地址常量，它不能改变值，正如数值型数组名不能被赋值一样。



```
str1="China"; str1=str2;
```

可以用strncpy函数将字符串2中前面n个字符复制到字符数组1中去。

```
strncpy(str1, str2, 2);
```

将str2中最前面2个字符复制到str1中，取代str1中原有的最前面2个字符。但复制的字符个数n不应多于str1中原有的字符（不包括‘\0’）。

字符串比较函数

`strcmp(字符串1, 字符串2)`

```
strcmp(str1, str2);  
strcmp("China", "Korea");  
strcmp(str1, "Beijing");
```

作用：比较字符串1和字符串2。

字符串比较的**规则**是：将两个字符串自左至右逐个字符相比(按ASCII码值大小比较)，直到出现不同的字符或遇到'\0'为止。

- (1) 如全部字符相同，则认为两个字符串相等；
- (2) 若出现不相同的字符，则以第1对不相同的字符的比较结果为准。

比较的**结果**由函数值带回。

- (1) 如果字符串1与字符串2相同，则函数值为0。
- (2) 如果字符串1>字符串2，则函数值为一个正整数。
- (3) 如果字符串1<字符串2，则函数值为一个负整数。

注意

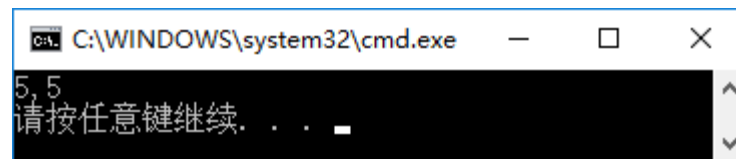
- 对两个字符串比较不能直接用`str1>str2`进行比较，因为`str1`和`str2`代表地址而不代表数组中全部元素，而只能用`(strcmp(str1, str2)>0)`实现，系统分别找到两个字符数组的第一个元素，然后顺序比较数组中各个元素的值。

测字符串长度的函数

strlen(字符数组)

作用：测试字符串长度的函数。函数的值为字符串中的实际长度(不包括'\0'在内)。

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[10]="China";
    printf("%d,%d\n",strlen(str),strlen("China"));
}
```



转换为大小写的函数

strlwr(字符串)

作用：将字符串中大写字母换成小写字母。

strupr(字符串)

作用：将字符串中小写字母换成大写字母。

注意

- 以上介绍了常用的8种字符串处理函数，它们属于**库函数**。库函数并非C语言本身的一部分，而是C语言编译系统为方便用户使用而提供的公共函数。不同的编译系统提供的函数数量和函数名、函数功能都不尽相同，使用时要小心，必要时查一下库函数手册。
- 在使用字符串处理函数时，应当在程序文件的开头用**#include <string.h>**把string.h文件包含到本文件中。

字符数组应用举例

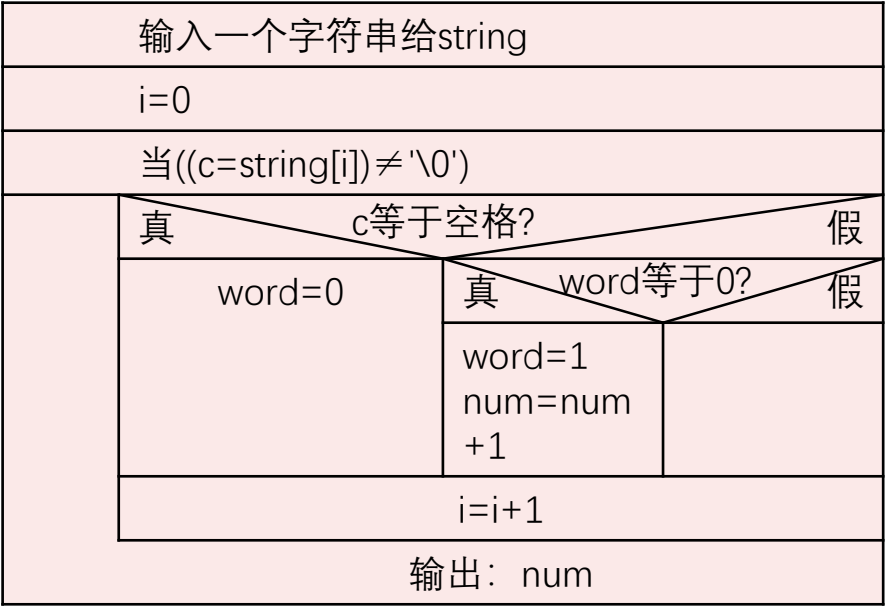
【例6.8】输入一行字符，统计其中有多少个单词，单词之间用空格分隔开。

string：用于存放字符串。

i：计数器，用于遍历字符串中的每个字符。

word：用于判断是否开始了一个新单词的标志。若word=0表示未出现新单词，如出现了新单词，就把word置成1。

num：用于统计单词数。

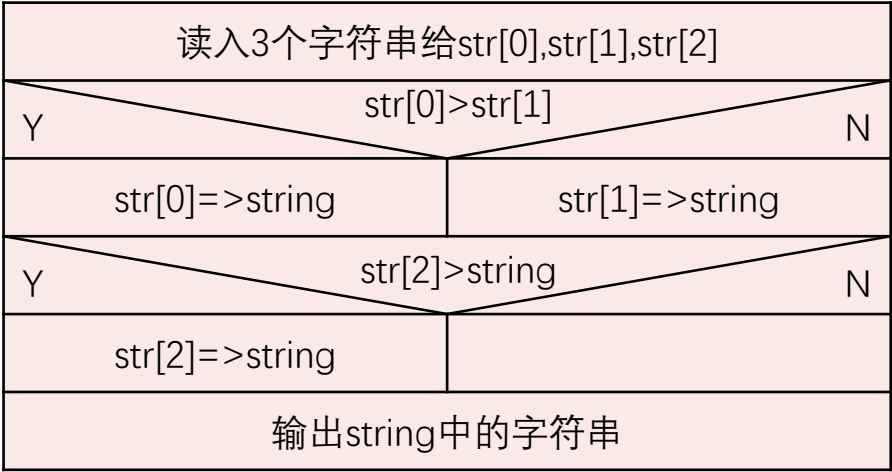


```
#include <stdio.h>
int main()
{
    char string[81];
    int i,num=0,word=0;
    char c;
    gets(string); //输入一个字符串给字符数组string
    for(i=0;(c=string[i])!='\0';i++) //只要字符不是'\0'就循环
        if(c==' ') word=0; //若是空格字符，使word置0
    else if(word==0) //如果不是空格字符且word原值为0
    {
        word=1; //使word置1
        num++; //num累加1，表示增加一个单词
    }
    printf("There are %d words in this line.\n",num); //输出单词数
    return 0;
}
```

字符数组应用举例

【例6.9】有3个字符串,要求找出其中“最大”者。

str[0]:	H	o	l	l	a	n	d	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0
str[1]:	C	h	i	n	a	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0
str[2]:	A	m	e	r	i	c	a	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

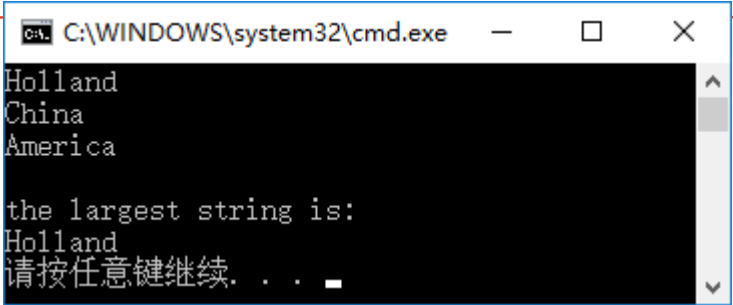


```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[3][20];    //定义二维字符数组
    char string[20];
    //定义一维字符数组，作为交换字符串时的临时字符数组
    int i;
    for(i=0;i<3;i++)
        gets(str[i]);    //读入3个字符串，分别给str[0],str[1],str[2]
```

```
    if(strcmp(str[0],str[1])>0)    //若str[0]大于str[1]
        strcpy(string,str[0]);    //把str[0]的字符串赋给字符数组string
    else                            //若str[0]小于等于str[1]
        strcpy(string,str[1]);    //把str[1]的字符串赋给字符数组string
    if(strcmp(str[2],string)>0)    //若str[2]大于string
        strcpy(string,str[2]);    //把str[2]的字符串赋给字符数组string
    printf("\nthe largest string is:\n%s\n",string);    //输出string
    return 0;
}
```



- (1) 流程图和程序注释中的“大于”是指两个字符串的比较中的“大于”。
- (2) str[0], str[1], str[2]和string是一维字符数组，其中可以存放一个字符串。
- (3) strcpy函数在将str[0], str[1]或str[2]复制到string时，最后都有一个'\0'。因此，最后用%s格式输出string时，遇到string中第一个'\0'即结束输出，并不是把string中的全部字符输出。



- JAVA 中数组与C类似，有String 类型

```
String str1 = "Hello World";  
String str2 = "Hello World";
```

```
str1.compareTo(str2)
```

- Python Numpy array and List

```
list1 = ['physics', 'chemistry', 1997, 2000]  
list2 = [1, 2, 3, 4, 5 ]  
list3 = ["a", "b", "c", "d"]
```

```
list = [2, 3, 4]  
for num in list:  
    print (num)
```