



补充：字符串处理函数

- ① `strcat(char s[], const char t[])`
- ② `strcpy(char s[], const char t[])`
- ③ `strcmp(const char s[], const char t[])`
- ④ `strlen(const char s[])`

使用前加 `#include <string.h>` //C方式

`#include <cstring>` //C++方式

VS下可以不加这个头文件，VS认为部分函数不够安全，使用前需要加

`#define _CRT_SECURE_NO_WARNINGS`



补充：字符串处理函数

- ① `strcat(char s[], const char t[])`
- 功 能：将字符串t连接到字符串s的尾部
- 输入参数：存放字符串s的字符数组s
- 存放字符串t的字符数组t(只读)
- 返 回 值：改变后的字符数组s
- 注意事项：字符数组s要有足够的空间(两串总长+1)



补充：字符串处理函数

- ① `strcat(char s[], const char t[])`

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str1[30]="People's Republic of ";
    char str2[]="China"; //缺省长度为6
    cout << strcat(str1, str2) << endl;
    return 0;
}
```

输出为：
People's Republic of China



补充：字符串处理函数

- ② strcpy(char s[], const char t[])
- 功 能：将字符串t复制到字符串s中,覆盖原串s
- 输入参数：存放字符串s的字符数组s
- 存放字符串t的字符数组t(只读)
- 返 回 值：改变后的字符数组s
- 注意事项：字符数组s要有足够的空间(串t长+1)



补充：字符串处理函数

• ② strcpy(char s[], const char t[])

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    int i;
    char a[]="student", b[]="hello";
    strcpy(a, b);
    cout << a << endl;
    for(i=0;i<8;i++)
        cout << (int)a[i] << ' ';
    cout << endl;
    return 0;
}
```

复制前a	s	t	u	d	e	n	t	\0
复制后a	h	e	l	l	o	\0	t	\0

↑
复制到\0为止
a[6]以后保持原值

//a/b数组的缺省长度8/6

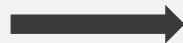
输出为:

hello

104 101 108 108 111 0 116 0



```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    int i;
    char a[]="student", b[]="hello";
    strcpy(a, b);
    cout << a << endl;
    for(i=0; i<8; i++)
        cout << (int)a[i] << ' ';
    cout << endl;
    return 0;
}
```



假设1:

char a[]="student", b[]="hello\0china";

则: a/b数组的缺省长度8/12

运行结果与本例完全相同

假设2:

char a[]="student", b[]="hellochina";

1、为什么错?

2、仅修改a的定义使正确, 如何做?



补充：字符串处理函数

- ② strcpy(char s[], const char t[])
- 字符串复制时包含'\0'，到'\0'为止

输出为：

heudent

104 101 117 100 101 110 116 0

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    int i;
    char a[]="student", b[]="hello";
    strncpy(a, b, 2);
    cout << a << endl;
    for(i=0;i<8;i++)
        cout << (int)a[i] << ' ' ;
    cout<<endl;
    return 0;
}
```



补充：字符串处理函数

- ② strcpy(char s[], const char t[])
- 字符串复制时包含'\0'，到'\0'为止

如果想从b[2]开始复制2个字符到a中，如何做？
即期望输出：lludent

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    int i;
    char a[]="student", b[]="hello";
    strncpy(a, &b[2], 2);
    cout << a << endl;
    for(i=0; i<8; i++)
        cout << (int)a[i] << ' ' ;
    cout<<endl;
    return 0;
}
```




补充：字符串处理函数

- ② strcpy(char s[], const char t[])
- 不能用直接赋值的方法进行字符串的复制

char str1[10]="china"; 正确，定义时赋初值

char str2[10];

str2="hello"; 编译报错

str2=str1; 编译报错



补充：字符串处理函数

- ③ `strcmp(const char s[], const char t[])`
- 功 能：比较字符串s和字符串t的大小
- 输入参数：存放字符串s的字符数组s (只读)
- 存放字符串t的字符数组t (只读)
- 返 回 值：整型值(0:相等 >0:串1大 <0:串1小)



补充：字符串处理函数

- ③ strcmp(const char s[], const char t[])

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str1[] = "house", str2[] = "horse";
    char str3[] = "abcd", str4[] = "abcde";
    char str5[] = "abcd", str6[] = "abc";
    char str7[] = "abcd", str8[] = "abcd";
    char str9[] = "abcd", str10[] = "abcd\0efgh";
    cout << strcmp(str1, str2) << endl;
    cout << strcmp(str3, str4) << endl;
    cout << strcmp(str5, str6) << endl;
    cout << strcmp(str7, str8) << endl;
    cout << strcmp(str9, str10) << endl;
    return 0;
} //仅比较无赋值，因此不需加_CRT_SECURE_NO_WARNINGS
```

1	串1>串2
-1	<
1	>
0	==
0	==



补充：字符串处理函数

- ③ strcmp(const char s[], const char t[])

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str1[]="abcd", str2[]="abcde";
    int k = strcmp(str1, str2);
    if (k==0)
        cout << "串1 = 串2" << endl;
    else if (k<0)
        cout << "串1 < 串2" << endl;
    else
        cout << "串1 > 串2" << endl;
    return 0;
}
```

另一种输出形式：

串1 < 串2



补充：字符串处理函数

- ③ `strcmp(const char s[], const char t[])`
- 两串相等的条件是长度相等且对应位置字符的ASCII码值相等
- 字符串的比较流程如下：

两串首字符对应比较，若**不等则返回非0**，若相等则继续比较下一个字符，重复到两串对应字符均为'\0'则结束比较，**返回0(相等)**

```
strcmp("house", "horse"); 到第3个字符结束  
strcmp("abcd", "abcde"); 到第5个字符结束  
strcmp("abcd", "abc"); 到第4个字符结束  
strcmp("abcd", "abcd"); 到第5个字符结束
```



补充：字符串处理函数

- ③ `strcmp(const char s[], const char t[])`
- 不相等返回时，有些系统返回-1/1，有些系统返回第一个不相等字符的ASCII差值，因此一般不比较具体值，而只是判断 >0 / <0 / $==0$

`strcmp("house", "horse");` VS编译器返回1
 某些编译器返回3



补充：字符串处理函数

- ③ `strcmp(const char s[], const char t[])`
- 不能直接用比较运算符比较字符串的大小

(但直接用比较运算符语法不错，只是含义不同)

```
char str1[]="horse", str2[]="house";  
int k;  
k = str1 < str2;  
cout << k << endl; //输出为0
```

数组名代表数组的首地址，因此两串直接用比较运算符表示比较存放两串的内存地址的大小，而不是比较串的大小

- 用`strncmp(s, t, n)`可比较两个字符串的前`n`个字符的大小



补充：字符串处理函数

- ④ `strlen(const char s[])`
- 功 能：求字符串的长度
- 输入参数：存放字符串的字符数组
- 返 回 值：整型值表示的长度
- 注意事项：返回第一个'\0'前的字符数量，不含'\0'



补充：字符串处理函数

• ④ strlen(const char s[])

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str[]="Hello";
    cout << sizeof(str) << endl;
    cout << strlen(str) << endl;
    return 0;
}
```

输出为:

6

5

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str[]="china\0Hello\0\0";
    cout << sizeof(str) << endl;
    cout << strlen(str) << endl;
    return 0;
}
```

输出为:

14

5

//同strcmp, 不需要加 _CRT_SECURE_NO_WARNINGS