



§. 基础知识题 – 常用的字符串处理函数

要求:

- 1、完成本文档中所有的题目并写出分析、运行结果，体会字符数组输入输出时不同用法的差异
- 2、题目明确指定编译器外，缺省使用VS2019即可
 - ★ 如果要换成其他编译器，可能需要自行修改头文件适配
 - ★ 部分代码编译时有warning，不影响概念理解，可以忽略
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，再拍照贴图
 - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
 - ★ 如果某题要求VS+Dev的，则如果两个编译器运行结果一致，贴VS的一张图即可，如果不一致，则两个图都要贴
- 4、转换为pdf后提交
- 5、**11月18日前**网上提交本次作业（在“文档作业”中提交）

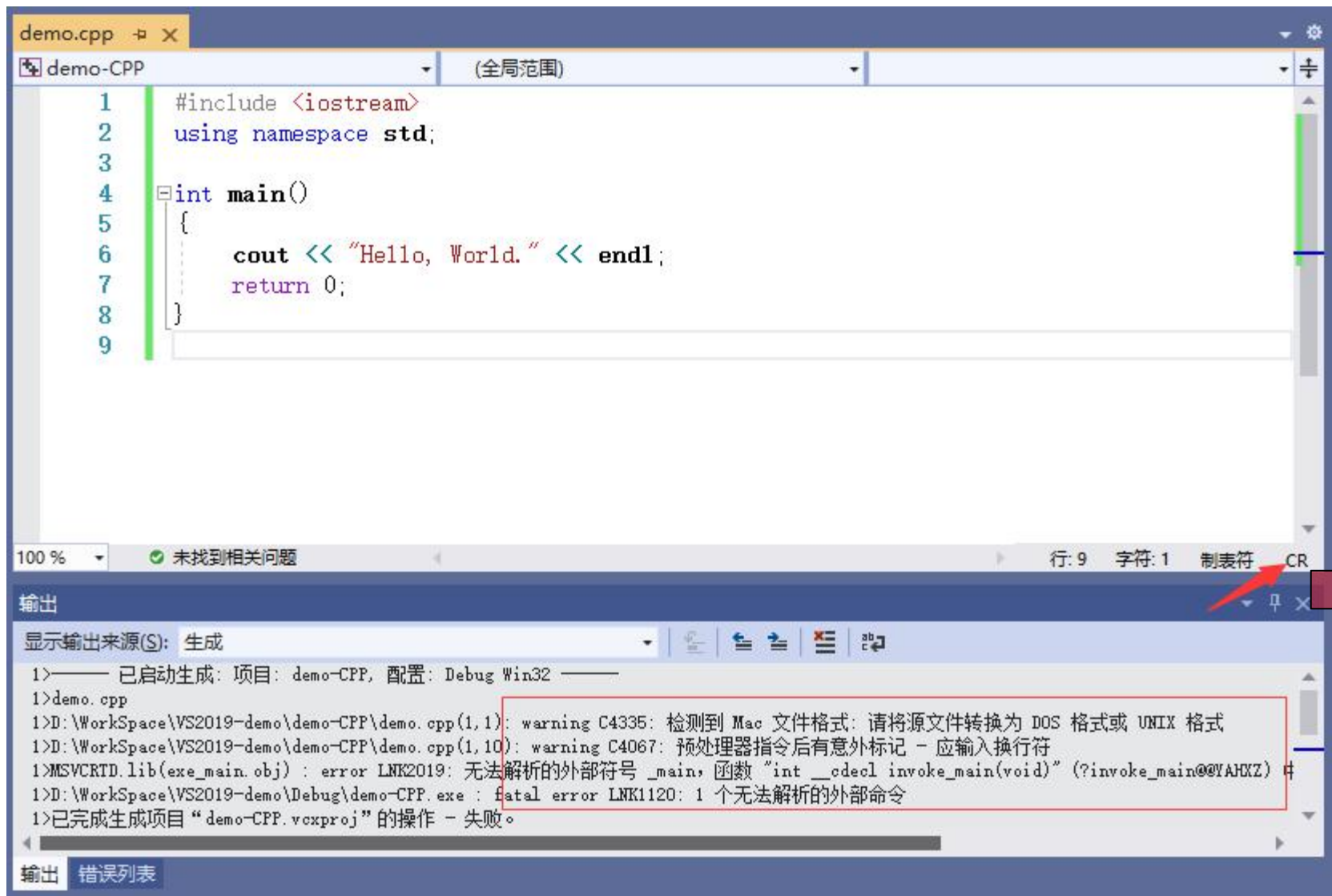


§. 基础知识题 - 常用的字符串处理函数

注意:

用WPS等其他第三方软件打开PPT, 将代码复制到VS2019中后, 如果出现类似下面的**编译报错**, 则观察源程序编辑窗口的

的右下角是否为CR, 如果是, 单击CR, 在弹出中选择CRLF, 再次CTRL+F5运行即可





§. 基础知识题 – 常用的字符串处理函数

总体知识:

★ 常用字符串处理函数

- ① `strlen (const char s[]);`
- ② `strcat (char dst[], const char src[]);`
- ③ `strncat(char dst[], const char src[], const unsigned int len);`
- ④ `strcpy (char dst[], const char src[]);`
- ⑤ `strncpy(char dst[], const char src[], const unsigned int len);`
- ⑥ `strcmp (const char s1[], const char s2[]);`
- ⑦ `strncmp(const char s1[], const char s2[], const unsigned int len);`
 - 更多的字符串处理函数通过作业完成并理解
 - 教材/参考资料中, 很多形式是 `const char *s`, 暂时忽略, 待学习指针后再进一步理解
 - 先不要考虑这些函数的返回值, 待学习指针后再进一步理解



§. 基础知识题 – 常用的字符串处理函数

1. `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

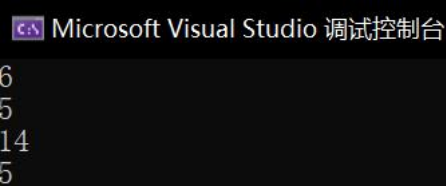
注意事项：返回第一个 `'\0'` 前的字符数量, 不含 `'\0'`



§. 基础知识题 – 常用的字符串处理函数

1. strlen(const char s[])

例1: 字符数组与字符串长度

<pre>#include <iostream> #include <cstring> using namespace std; int main() { char str1[]="Hello"; cout << sizeof(str1) << endl; cout << strlen(str1) << endl; char str2[]="china\0Hello\0\0"; cout << sizeof(str2) << endl; cout << strlen(str2) << endl; return 0; } //读操作, 不需要加_CRT_SECURE_NO_WARNINGS</pre>	<p>//给出程序的运行结果</p> 
--	--

<p>问题:</p> <ol style="list-style-type: none">1、求数组长度时, 无论是否有显式\0, 最后一定有隐式(显示/隐式)的\02、当含有多个\0(显式/隐式)时, 长度计算到从左往右数第一个\0为止
--

本页需填写答案



§. 基础知识题 – 常用的字符串处理函数

2. `strcat(char dst[], const char src[])`

功 能：将字符串src连接到字符串dst的尾部 (含尾零)

输入参数：存放字符串dst的字符数组dst

存放字符串src的字符数组src (只读)

返 回 值：改变后的字符数组dst

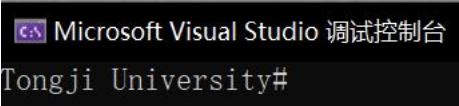
注意事项：字符数组dst要有足够的空间 (两串总长+1)



§. 基础知识题 – 常用的字符串处理函数

2.strcat(char dst[], const char src[])

例2：字符串连接

<pre>#define _CRT_SECURE_NO_WARNINGS //VS2019需要 #include <iostream> #include <cstring> using namespace std; int main() { char str1[30]="Tongji "; //不能缺省，至少18字节!!! char str2[]="University"; cout << strcat(str1, str2) << '#' << endl; //加#的目的? return 0; }</pre>	<p>//给出程序的运行结果</p> 
--	--

<p>问题：</p> <p>1、str2数组的默认长度是11。</p> <p>2、结合前面字符数组输入/输出的作业，strcat复制时包含（包含/不包含）src的\0。</p>
--

本页需填写答案



§. 基础知识题 - 常用的字符串处理函数

2. strcat(char dst[], const char src[])

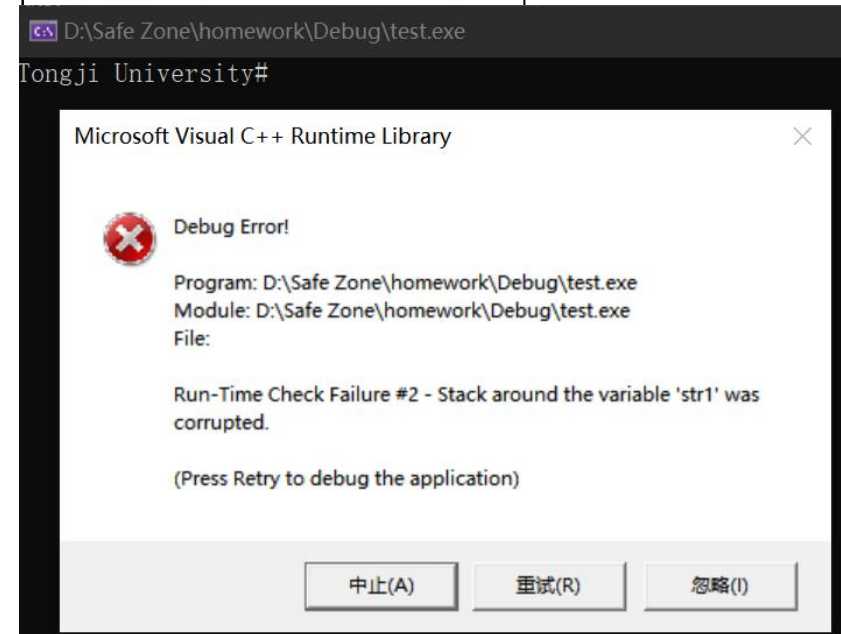
例3: 字符串连接 (错误)

```
#define _CRT_SECURE_NO_WARNINGS //VS2019需要
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char str1[]="Tongji ";
    char str2[]="University";
    cout << strcat(str1, str2) << '#' << endl; //加#的目的?

    return 0;
}
```

//给出程序的运行结果



问题:

- 1、str1数组的大小必须给出, 不能默认, 其最小长度是18 (针对本例的一个具体数字)。
- 2、dst数组的最小长度是字符串src、dst原长和+1 (通用规则) 才能保证正确。

本页需填写答案



§. 基础知识题 - 常用的字符串处理函数

3. `strncat(char dst[], const char src[], const unsigned int n)`

功 能：将字符串src的**前n个字符**连接到字符串dst的尾部

输入参数：存放字符串dst的字符数组dst

存放字符串src的字符数组src**(只读)**

要复制的长度n**(只读，如果n超过src长度，则只连接src个)**

返 回 值：改变后的字符数组dst

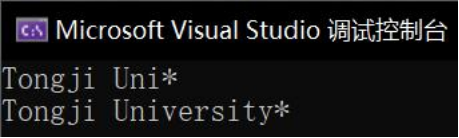
注意事项：字符数组dst要有足够的空间**(原dst长度+n+1)**



§. 基础知识题 - 常用的字符串处理函数

3.strncat(char dst[], const char src[], const unsigned int n)

例4：字符串连接前n个字符

<pre>//例：字符串连接前n个字符 #define _CRT_SECURE_NO_WARNINGS //VS2019需要 #include <iostream> #include <cstring> using namespace std; int main() { char str1[30]="Tongji "; char str2[30]="Tongji "; char str3[]="University"; cout << strncat(str1, str3, 3) << '*' << endl; cout << strncat(str2, str3, 300) << '*' << endl; return 0; }</pre>	<p>//给出程序的运行结果</p> 
---	--

问题：
但n超过src表示的字符串的长度时，连接规则是只连接src中的所有字符。

本页需填写答案



§. 基础知识题 - 常用的字符串处理函数

3. strncat(char dst[], const char src[], const unsigned int n)

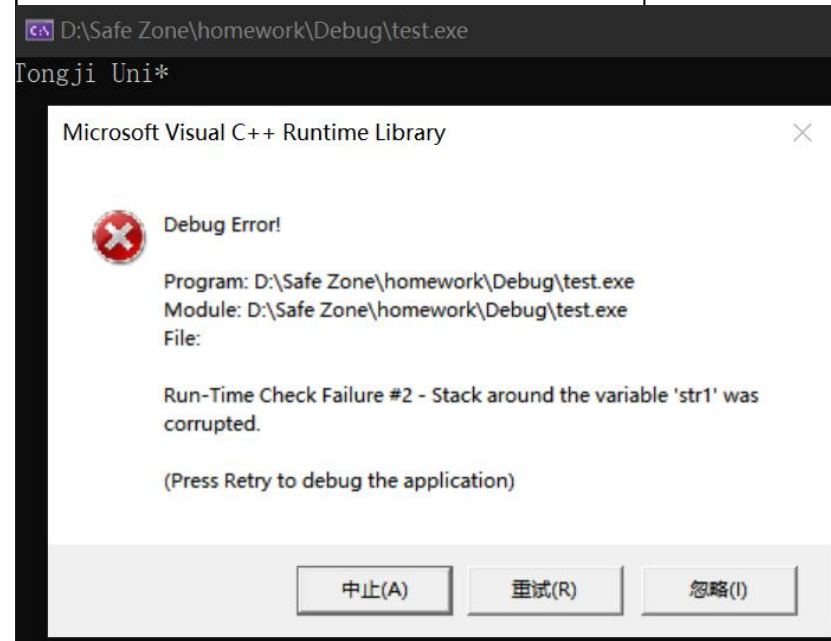
例5: 字符串连接前n个字符 (错误)

```
//例: 字符串连接前n个字符
#define _CRT_SECURE_NO_WARNINGS //VS2019需要
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char str1[]="Tongji ";
    char str3[]="University"; //缺省长度为11
    cout << strncat(str1, str3, 3) << '*' << endl;

    return 0;
}
```

//给出程序的运行结果



问题:

- 1、str1数组的大小必须给出, 不能默认, 其最小长度是11 (针对本例的一个具体数字)。
- 2、dst数组的最小长度是字符串dst原长+连接字符个数n+1 (通用规则) 才能保证正确。

本页需填写答案



§. 基础知识题 – 常用的字符串处理函数

4. strcpy(char dst[], const char src[])

功 能：将字符串src复制到字符串dst中, 覆盖原dst串

输入参数：存放字符串dst的字符数组dst

存放字符串src的字符数组src (只读)

返 回 值：改变后的字符数组dst


注意事项：字符数组dst要有足够的空间 (串src长+1)



§. 基础知识题 - 常用的字符串处理函数

4. strcpy(char dst[], const char src[])

例6: 字符串拷贝

<pre>#define _CRT_SECURE_NO_WARNINGS //VS2019需要 #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; }</pre>	<p>//给出程序的运行结果</p> 
---	--

<p>问题:</p> <ol style="list-style-type: none">1、字符串复制时, 复制到src的最后一个字符为止, 包含 (包含/不包含) \0, 之后的字符不再复制。2、在运行截图中用箭头指出证明结论1的位置
--

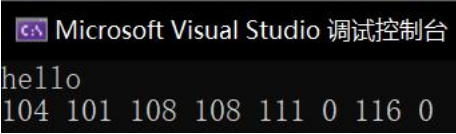
本页需填写答案



§. 基础知识题 – 常用的字符串处理函数

4. strcpy(char dst[], const char src[])

例7：字符串拷贝

<pre>#define _CRT_SECURE_NO_WARNINGS //VS2019需要 #include <iostream> #include <cstring> using namespace std; int main() { int i; char a[]="student", b[]="hello\0china"; strcpy(a,b); cout << a << endl; for(i=0;i<8;i++) cout << int(a[i]) << ' '; cout << endl; return 0; }</pre>	<p>//给出程序的运行结果</p> 
--	--

- 问题：
- 1、a数组的默认大小是8，b数组的默认大小是11。
 - 2、b数组的大小超过了a数组的大小，为什么运行不出错？b的字符串只到hello，小于字符串a长度
 - 3、本例中，复制到b[6]就停止复制了

本页需填写答案



§. 基础知识题 - 常用的字符串处理函数

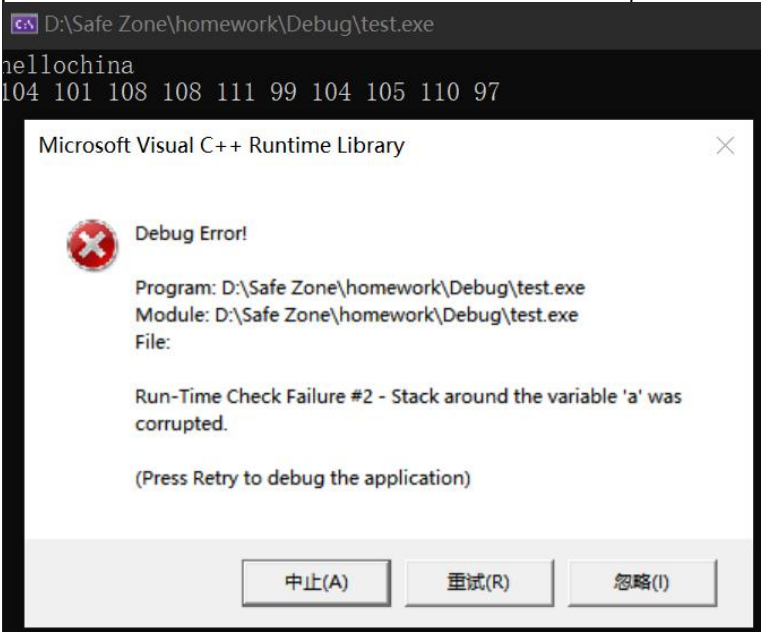
4. strcpy(char dst[], const char src[])

例8: 字符串拷贝 (有错)

```
#define _CRT_SECURE_NO_WARNINGS //VS2019需要
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    int i;
    char a[11]="student", b[]="hellochina";
    strcpy(a,b);
    cout << a << endl;
    for(i=0; a[i]!='\0'; i++)
        cout << int(a[i]) << ' ';
    cout << endl;

    return 0;
}
```

//给出程序的运行结果



- 问题:
- 1、本程序为什么会错? b的字符串长度大于a的字符串长度, b无法拷贝到字符数组a
 - 2、仅改a的定义使正确, 如何做? (直接上面的源程序中用红色写出修改内容即可)
 - 3、dst数组的最小长度是字符串src长+1 (通用规则) 才能保证正确。

本页需填写答案



§. 基础知识题 – 常用的字符串处理函数

5. strncpy(char dst[], const char src[], unsigned int n)

功 能：将字符串src的**前n个**复制到字符串dst中, 覆盖原dst串

输入参数：存放字符串dst的字符数组dst

存放字符串src的字符数组src (**只读**)

要复制的长度n (**只读, 如果n超过src长度, 则只复制src个**)

返 回 值：改变后的字符数组dst

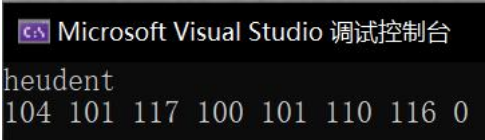
注意事项：字符数组dst要有足够的空间 (**min(串src长, n)+1**)



§. 基础知识题 – 常用的字符串处理函数

5. strncpy(char dst[], const char src[], unsigned int n)

例9：字符串拷贝前n个字符

<pre>#define _CRT_SECURE_NO_WARNINGS //VS2019需要 #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; }</pre>	<p>//给出程序的运行结果</p> 
--	--

问题：
本程序证明了strncpy复制时，不包含（包含/不包含）\0。

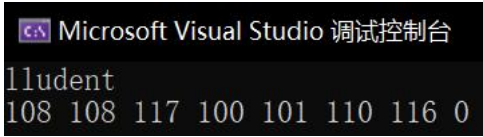
本页需填写答案



§. 基础知识题 – 常用的字符串处理函数

5. strncpy(char dst[], const char src[], unsigned int n)

例10：字符串拷贝前n个字符

<pre>#define _CRT_SECURE_NO_WARNINGS //VS2019需要 #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; }</pre>	<p>//给出程序的运行结果</p> 
--	--

问题：

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）

（直接在源程序中修改____位置即可）

本页需填写答案



§. 基础知识题 – 常用的字符串处理函数

5. strncpy(char dst[], const char src[], unsigned int n)

例11：字符串拷贝前n个字符（深度讨论）

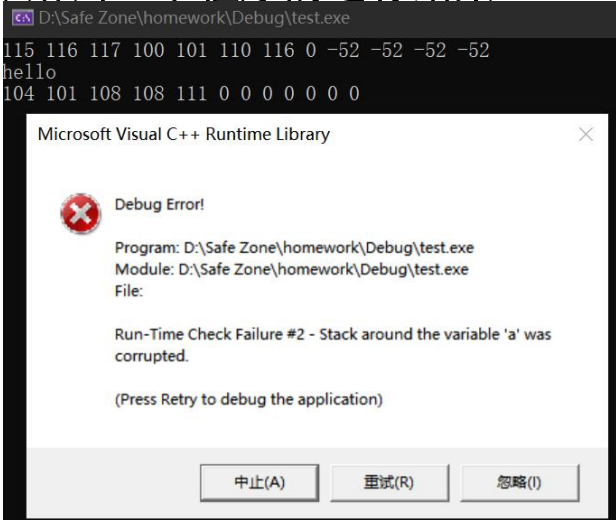
```
#define _CRT_SECURE_NO_WARNINGS //VS2019需要
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    int i;
    char a[] = "student", b[] = "hello";
    for (i = 0; i < 12; i++) //12已越界，目的？
        cout << int(a[i]) << ' ';
    cout << endl;

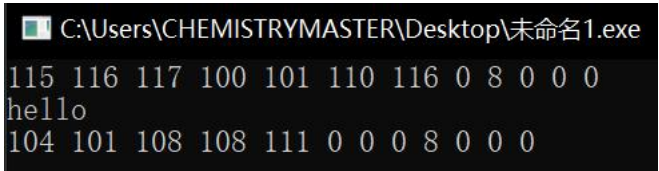
    strncpy(a, b, 200);
    cout << a << endl;

    for (i = 0; i < 12; i++) //12已越界，目的？
        cout << int(a[i]) << ' ';
    cout << endl;
    return 0;
} //越界目的：方便观察n超src、dst长度时，dst的变化
```

//给出VS下程序的运行结果



//给出Dev下程序的运行结果



问题：

观察两个for循环的后6个数字的输出，能得到什么结论？（提示：n超src长度是到\0为止吗）

答：n超src长度并不是到\0为止，而是把若干个\0也复制上，拷贝到dst上了

本页需填写答案



§. 基础知识题 – 常用的字符串处理函数

5. strncpy(char dst[], const char src[], unsigned int n)

例12: 字符串拷贝前n个字符（深度讨论）

```
#define _CRT_SECURE_NO_WARNINGS //VS2019需要
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    int i;
    char a[] = "student", b[] = "hello";
    for (i = 0; i < 20; i++) //20已越界，目的?
        cout << int(a[i]) << ' ';
    cout << endl;

    strncpy(a, b, 200);
    cout << a << endl;

    for (i = 0; i < 20; i++) //20已越界，目的?
        cout << int(a[i]) << ' ';
    cout << endl;
    return 0;
} //越界目的: 方便观察n超src、dst长度时，dst的变化
```

同上例，
数组越界到20

//给出VS下程序的运行结果



//给出Dev下程序的运行结果



问题:
如果n超过了src的长度, 则将src连同多个尾零拷贝到dst上, 而不只拷贝src。

本页需填写答案



§. 基础知识题 – 常用的字符串处理函数

6. strcmp(const char s1[], const char s2[])

功 能：比较字符串s1和字符串s2的大小

输入参数：存放字符串s1的字符数组s1 (只读)

存放字符串s2的字符数组s2 (只读)

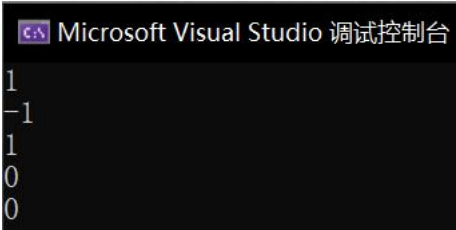
返 回 值：整型值 (0:相等 >0:串1大 <0:串1小)



§. 基础知识题 – 常用的字符串处理函数

6. strcmp(const char s1[], const char s2[])

例13: 字符串比较

<pre>#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; }</pre>	<p>//给出程序的运行结果</p> 
--	--

问题：两个字符串相等的条件是？
答：尾零前各字符相等、字符个数相等

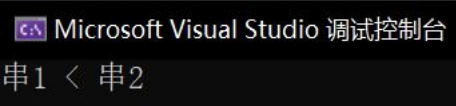
本页需填写答案



§. 基础知识题 – 常用的字符串处理函数

6. strcmp(const char s1[], const char s2[])

例14：字符串比较（另一种形式）

<pre>#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; }</pre>	<p>//给出程序的运行结果</p> 
---	--

问题：给出两个字符串比较的执行过程

答：先逐个对比两个字符串对应位置的字符，直到有不同字符为止，不同字符中ASCII大的字符串判为大
若遇见任意一个字符的尾零前仍未比出大小，而该位置一个字符串为尾零，另一个不为尾零，则判后者大
若遇见任意一个字符的尾零前仍未比出大小，而该位置两个字符串都为尾零，则判两个字符串相等



§ . 基础知识题 – 常用的字符串处理函数

6. strcmp(const char s1[], const char s2[])

例15：字符串比较（编译不错，但运行结果与期望不符合）

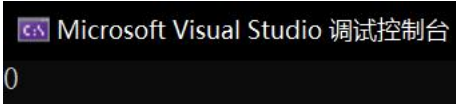
```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char str1[]="house", str2[]="horse";
    int k;

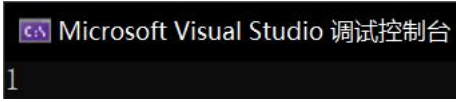
    k = strcmp(str1, str2);
    cout << k << endl;

    return 0;
}
```

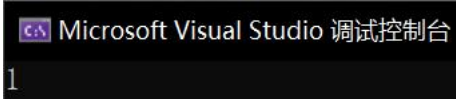
//给出程序的运行结果



//将str1和str2的内容互换，给出运行结果



//将str1和str2都置为"house"，给出运行结果



问题：

这个程序的运行结果是表示str1和str2的对应位置字符的ASCII大小进行比较。



§. 基础知识题 – 常用的字符串处理函数

7. `strncmp(const char s1[], const char s2[], const unsigned int n)`

功 能：比较字符串s1和字符串s2的前n个字符的大小

输入参数：存放字符串s1的字符数组s1 (只读)

存放字符串s2的字符数组s2 (只读)

要比较的长度n (只读)

返 回 值：整型值(0:相等 >0:串1大 <0:串1小)



§. 基础知识题 - 常用的字符串处理函数

7. strcmp(const char s1[], const char s2[], const unsigned int n)

例16: 字符串比较前n个字符

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char str1[] = "abcd", str2[] = "abcde";

    cout << strcmp(str1, str2, 3) << endl;
    cout << strcmp(str1, str2, 4) << endl;
    cout << strcmp(str1, str2, 5) << endl;
    cout << strcmp(str1, str2, 100) << endl;

    return 0;
}
```

//给出程序的运行结果

```
Microsoft Visual Studio 调试控制台
0
0
-1
-1
```

//将str2也置为"abcd"
给出程序的运行结果

```
Microsoft Visual Studio 调试控制台
0
0
0
0
```

问题:

- 1、当n小于短串长度时, 则比较到第n个字符。
- 2、当n大于等于短串长度时, 则比较到短串的尾零为止。
- 3、如果n超过长串的长度, 则比较到短串的尾零为止。

本页需填写答案