



§. 基础知识题 – 常用的字符串处理函数

要求:

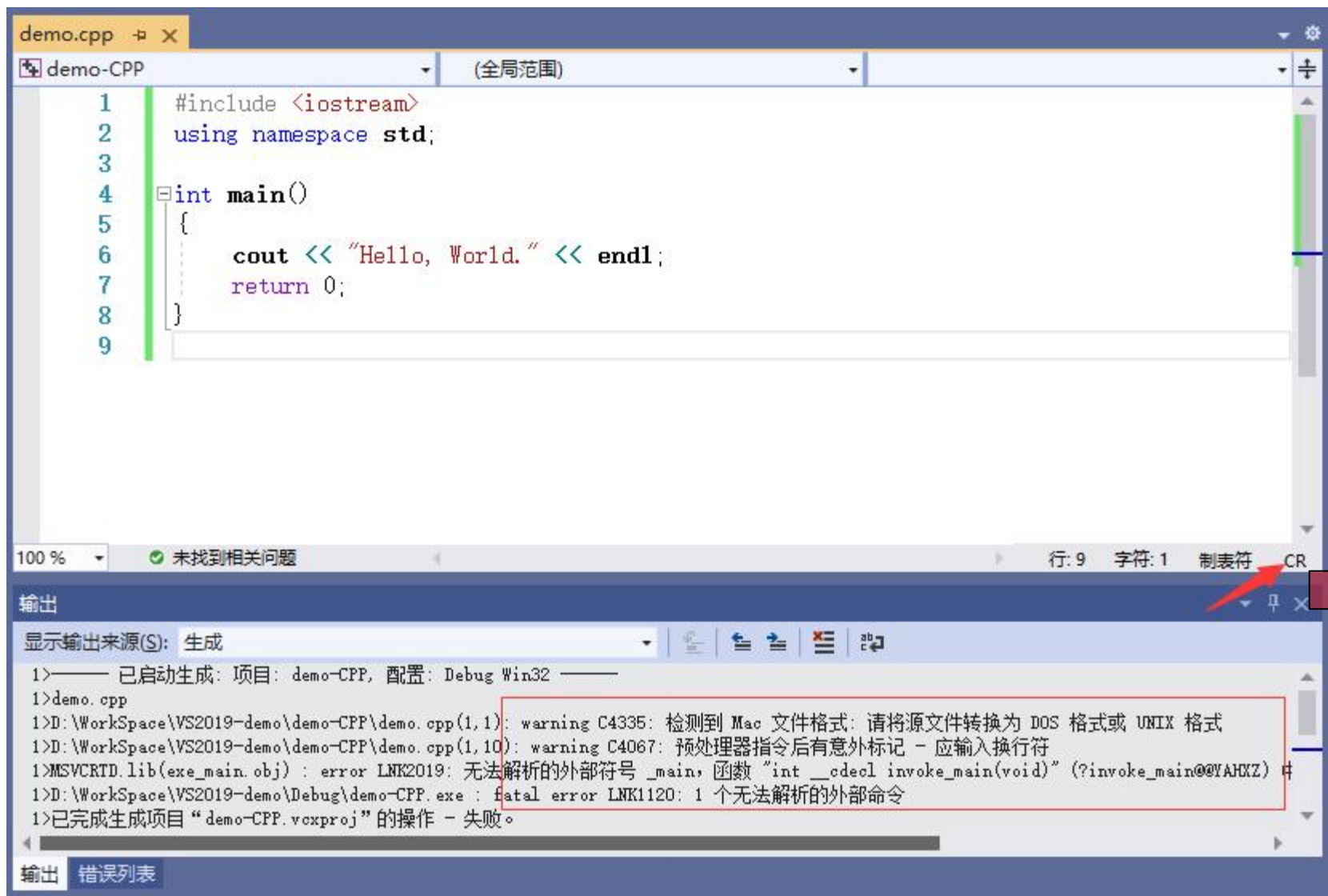
- 1、完成本文档中所有的题目并写出分析、运行结果，体会字符数组输入输出时不同用法的差异
- 2、题目明确指定编译器外，缺省使用VS2022即可
 - ★ 如果要换成其他编译器，可能需要自行修改头文件适配
 - ★ 部分代码编译时有warning，不影响概念理解，可以忽略
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，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14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VS2022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

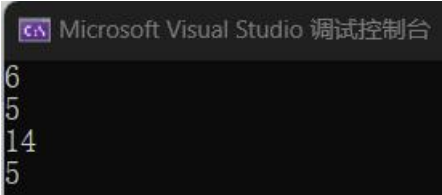
注意事项：返回第一个 `'\0'` 前的字符数量, 不含 `'\0'`



§. 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> <p>1、求数组长度时, 无论是否有显式\0, 最后一定有__<u>隐式</u>__ (显示/隐式)的\0</p> <p>2、当含有多个\0(显式/隐式)时, 字符串长度计算到__<u>第一</u>__个\0为止</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[])`

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 //VS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>
--	--------------------

问题:

1、str2数组的默认长度是 11。

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。

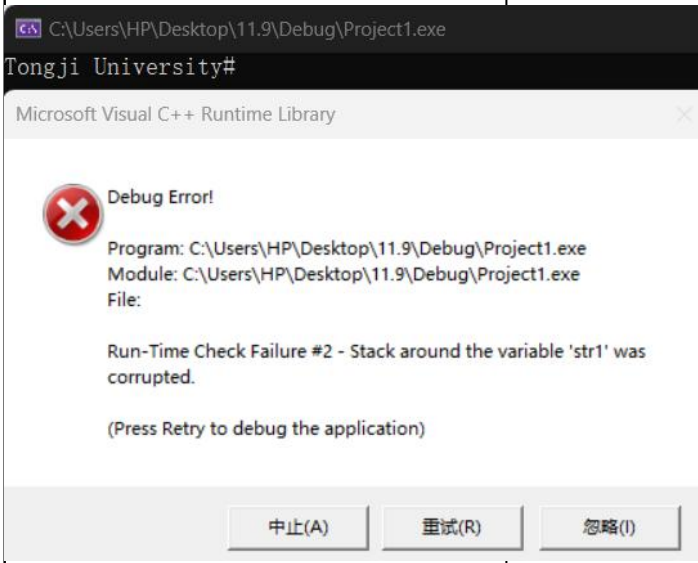
本页需填写答案



§. 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)

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

- 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 两个字符串总长+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

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

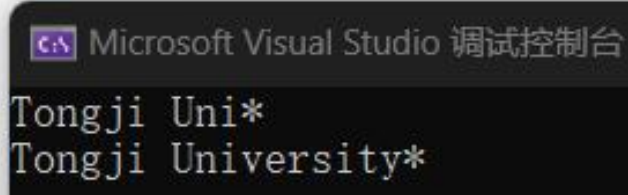
注意事项: 字符数组dst要有足够的空间 (**原dst长度+n+1**)



§. 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 //VS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>	<pre>//给出程序的运行结果</pre> 
---	--

问题：
但n超过src表示的字符串的长度时，连接规则是将src整个字符串连接到dst后。

本页需填写答案