



## §. 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、**5月10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

附：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

```
demo.cpp
demo-cpp (全局范围) main()
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "Hello, 同济!" << endl;
6     return 0;
7 }
8
```

100 % 未找到相关问题 行: 4 字符: 2 空格 SYS CR

输出

显示输出来源(S): 生成

生成开始于 22:23...

1>—— 已启动生成: 项目: demo-cpp, 配置: Debug Win32 ——

1>demo.cpp

1>D:\WorkSpace\VS2022-demo\demo-cpp\demo.cpp(1,1): warning C4335: 检测到 Mac 文件格式: 请将源文件转换为 DOS 格式或 UNIX 格式

1>D:\WorkSpace\VS2022-demo\demo-cpp\demo.cpp(1,10): warning C4067: 预处理器指令后有意外标记 - 应输入换行符

1>MSVCRTD.lib(exe\_main.obj) : error LNK2019: 无法解析的外部符号 \_main, 函数 "int \_\_cdecl invoke\_main(void)" (?invoke\_main@YAHKZ) 中引用了该符号

1>D:\WorkSpace\VS2022-demo\Debug\demo-cpp.exe : fatal error LNK1120: 1 个无法解析的外部命令

1>已完成生成项目 "demo-cpp.vcxproj" 的操作 - 失败。

生成: 0 成功, 1 失败, 0 最新, 0 已跳过

生成于 22:23 完成, 耗时 01.132 秒

错误列表 输出



## §. 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

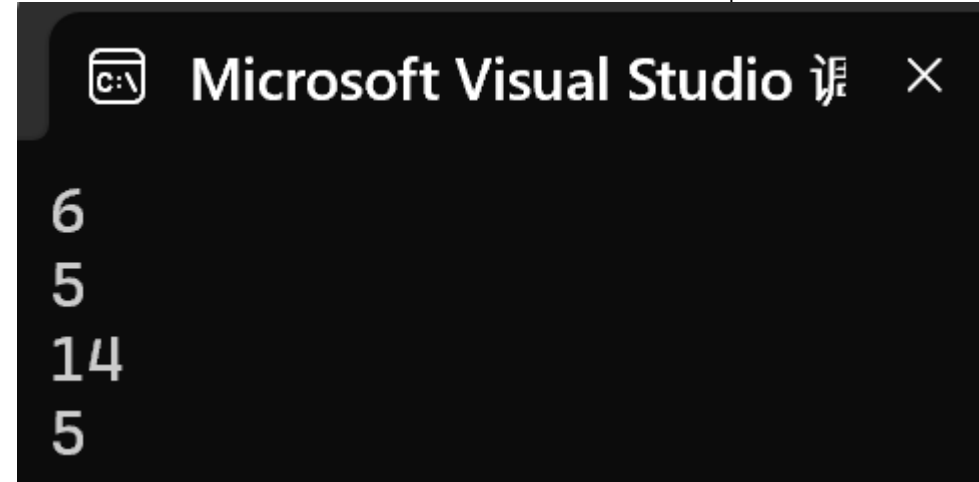
```
#include <iostream>
#include <cstring>
using namespace std;

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
```

//给出程序的运行结果



问题:

- 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\0
- 2、当含有多个\0(显式/隐式)时, 字符串长度计算到\_\_\_\_第一个\_\_\_\_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

```
#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;
}
```

//给出程序的运行结果

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)

```
#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;
}
```

//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\_\_\_\_\_两字符串总长+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)

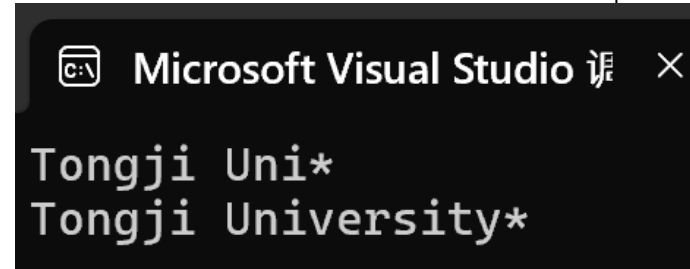
例4: 字符串连接前n个字符

```
//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;
}
```

//给出程序的运行结果



```
Microsoft Visual Studio 16.0
Tongji Uni*
Tongji University*
```

问题:

当n超过src表示的字符串的长度时, 连接规则是\_\_把能复制的全部接到后面去\_\_。

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 //VS需要
#include <iostream>
#include <cstring>
using namespace std;

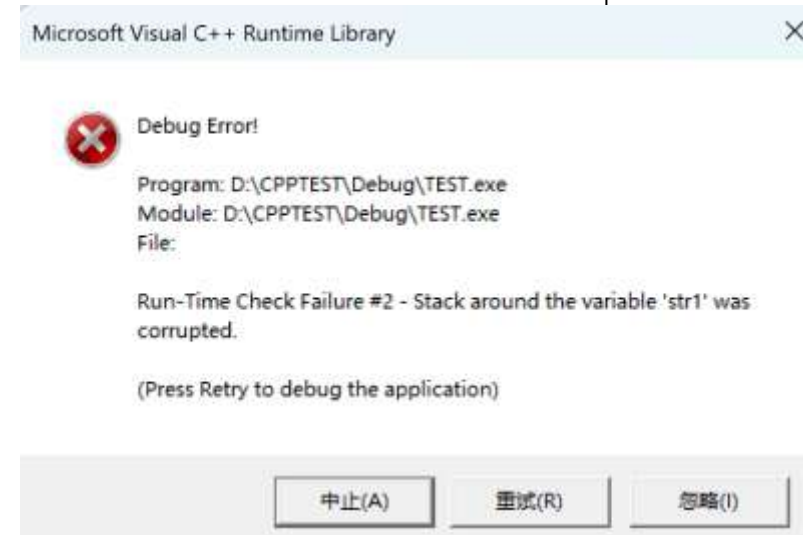
int main()
{
    char str1[]="Tongji ";

    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

弹窗提示越界访问



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\_\_\_\_原dst长度+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

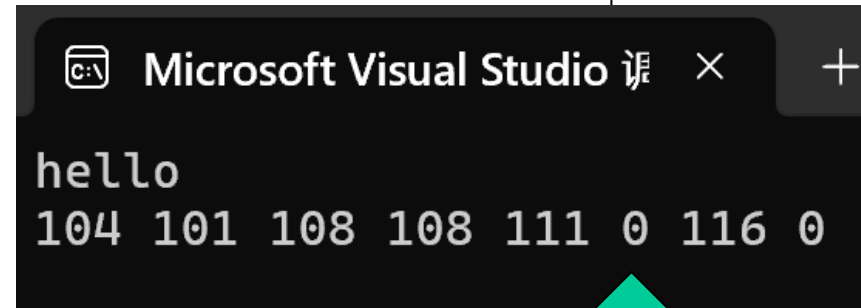
4. strcpy(char dst[], const char src[])

例6: 字符串拷贝

```
#define _CRT_SECURE_NO_WARNINGS //VS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;
}
```

//给出程序的运行结果



问题:

- 1、字符串复制时, 复制到src的\_\_\0\_\_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

```
#define _CRT_SECURE_NO_WARNINGS //VS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;
}
```

//给出程序的运行结果

Microsoft Visual Studio 运行 × +

hello  
104 101 108 108 111 0 116 0

问题:

- 1、a数组的默认大小时\_\_8\_\_, b数组的默认大小是\_\_\_\_12\_\_\_\_。
- 2、b数组的大小超过了a数组的大小, 为什么运行不出错? 因为函数只读到并复制到显式\0
- 3、本例中, 复制到b[\_5\_]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

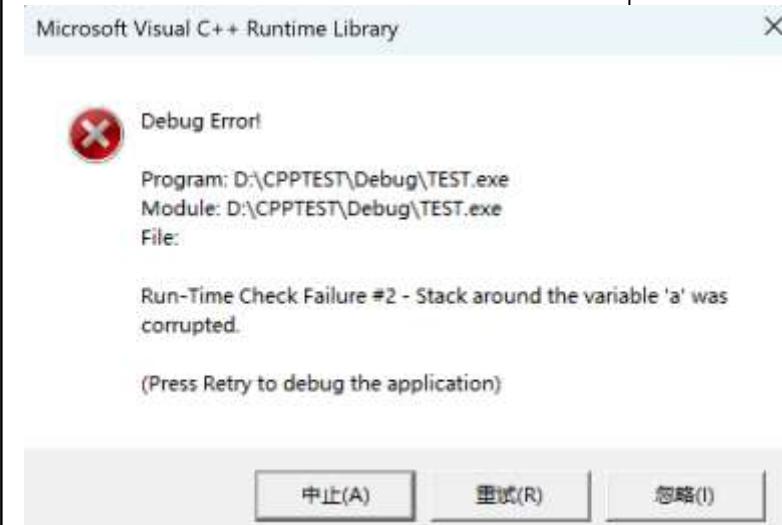
4. strcpy(char dst[], const char src[])

例8: 字符串拷贝 (有错)

```
#define _CRT_SECURE_NO_WARNINGS //VS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数组大小比a大, 复制过去会溢出
- 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\_\_\_\_\_src第一个\0前的字符数+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 ( **$\min(\text{串src长}, n) + 1$** )





## §. 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

```
#define _CRT_SECURE_NO_WARNINGS //VS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;
}
```

//给出程序的运行结果

```
Microsoft Visual Studio 调 × +
heudent
104 101 117 100 101 110 116 0
```

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

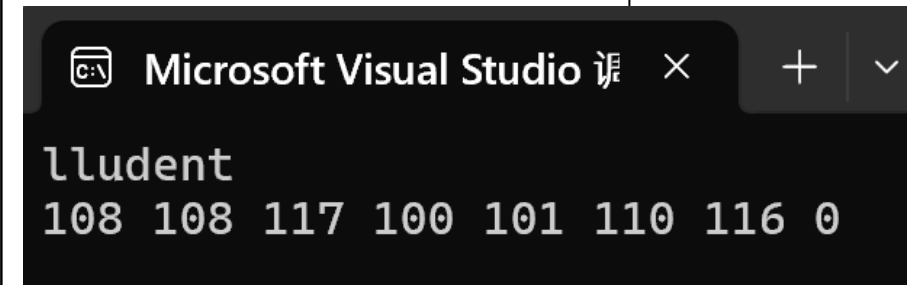
5. strncpy(char dst[], const char src[], unsigned int n)

例10: 字符串拷贝前n个字符

```
#define _CRT_SECURE_NO_WARNINGS //VS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;
}
```

//给出程序的运行结果



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 //VS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;

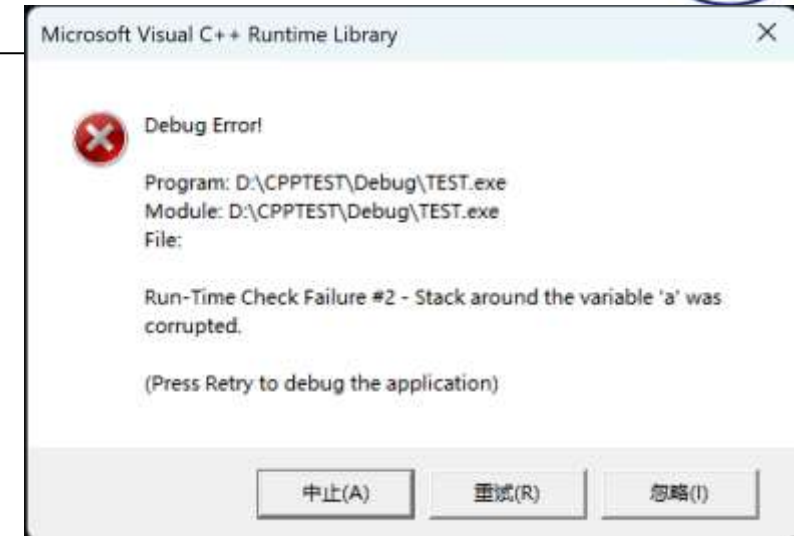
    strncpy(a, b, 200);
    cout << a << endl;

    for (i = 0; i < 12; i++) //12已越界，目的？
        cout << int(a[i]) << ' ';
    cout << endl;
    return 0;
}
```

//给出VS下程序的运行结果

//给出Dev下程序的运行结果

```
D:\桌面2\testDEV.exe
115 116 117 100 101 110 116 0 8 0 0 0
hello
104 101 108 108 111 0 0 0 8 0 0 0
```



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？）

N超src长度时，strncpy（）会复制到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

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 //VS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;

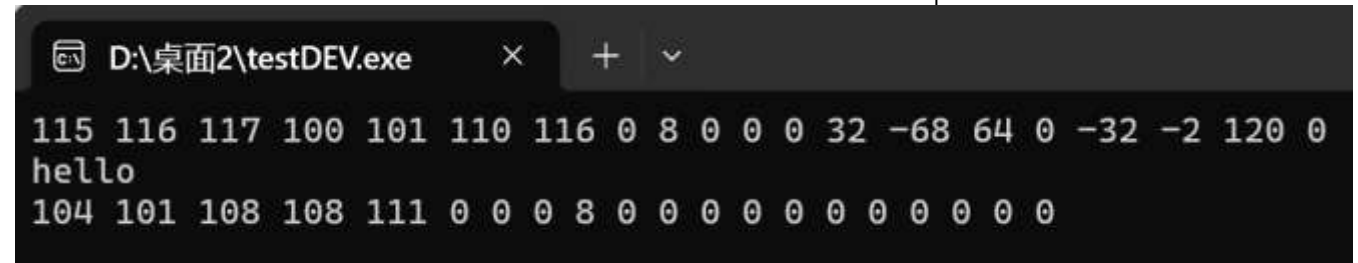
    strncpy(a, b, 200);
    cout << a << endl;

    for (i = 0; i < 20; i++) //20已越界, 目的?
        cout << int(a[i]) << ' ';
    cout << endl;
    return 0;
}
```

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\_\_strncpy () 会先复制到src的\0, 并用\0填充剩余合法空间\_\_。

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

```
#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;
}
```

//给出程序的运行结果

```
Microsoft Visual Studio 调 ×
1
-1
1
0
0
```

问题: 两个字符串相等的条件是?

尾零前的字符长度、内容相同

本页需填写答案



## §. 基础知识题 – 常用的字符串处理函数

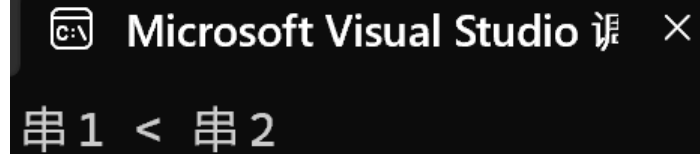
6. strcmp(const char s1[], const char s2[])

例14: 字符串比较 (另一种形式)

```
#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;
}
```

//给出程序的运行结果



串1 < 串2

问题: 给出两个字符串比较的执行过程

逐字符比较ASCII码, 如果相等则继续比较下一个字符, 直到遇见\0或者首次出现不相等的字符

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）

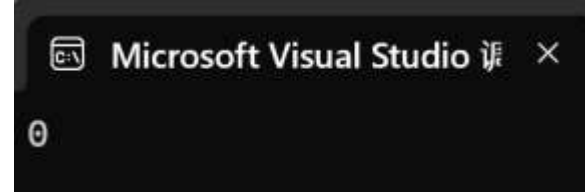
```
#include <iostream>
#include <cstring>
using namespace std;

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

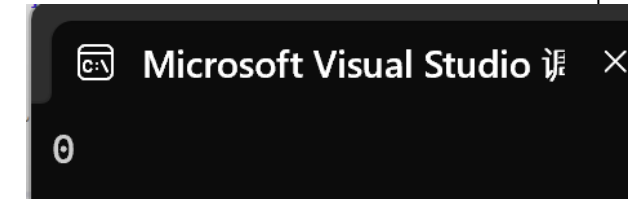
    k = strcmp(str1, str2);
    cout << k << endl;

    return 0;
}
```

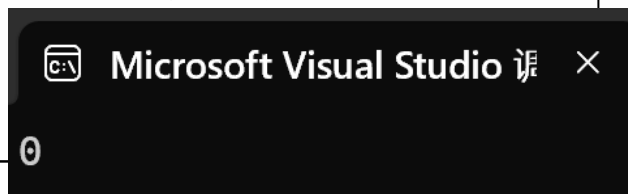
//给出程序的运行结果



//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\_\_\_\_\_内存地址\_\_\_\_\_进行比较。

本页需填写答案





## §. 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

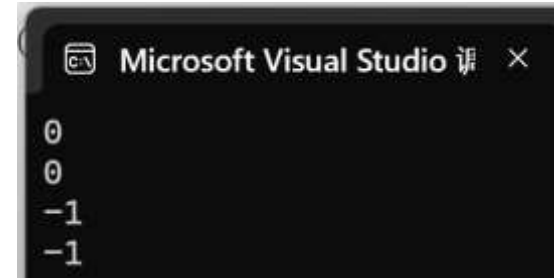
```
#include <iostream>
#include <cstring>
using namespace std;

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;

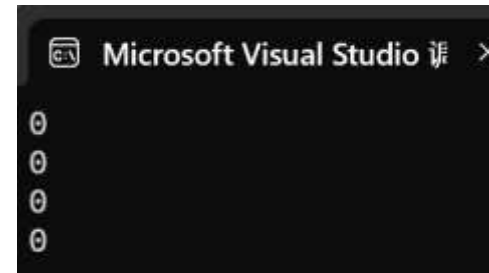
    return 0;
}
```

//给出程序的运行结果



```
0
0
-1
-1
```

//将str2也置为"abcd"  
给出程序的运行结果



```
0
0
0
0
```

问题:

- 1、当n小于短串长度时, 则比较到\_\_\_\_\_第n个\_\_\_\_\_。
- 2、当n大于等于短串长度时, 则比较到\_\_\_\_\_短串长度\_\_\_\_\_为止。
- 3、如果n超过长串的长度, 则比较到\_\_\_\_\_长串长度\_\_\_\_\_为止。

本页需填写答案