

C 中string和string.h的作用和区别计算机二级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/582/2021\\_2022\\_C\\_\\_\\_E4\\_B8\\_ADstri\\_c97\\_582641.htm](https://www.100test.com/kao_ti2020/582/2021_2022_C___E4_B8_ADstri_c97_582641.htm) 2009年下半年全国计算机等级考试你准备好了没?考计算机等级考试的朋友,2009年下半年全国计算机等级考试时间是2009年9月19日至23日。更多优质资料尽在百考试题论坛 百考试题在线题库

```
#include <string>. void main() { string  
aaa = " abcsd d " . printf( " looking for abc from abcdec d %s\n " ,  
(strcmp(aaa, " abc " )) ? " Found " : " Not Found " ). } 不能正确执行，提示说是string类型没有定义 而下面：
```

```
#include <string>. using  
namespace std. void main() { string aaa = " abcsd d " . printf( "  
looking for abc from abcdec d %s\n " , (strcmp(aaa, " abc " )) ? "  
Found " : " Not Found " ). }
```

这里的string编译器就认识了，但是strcmp就不认识了呢？一般一个C的老的带“.h”扩展名的库文件，比如iostream.h，在新标准后的标准库中都有一个不带“.h”扩展名的相对应，区别除了后者的好多改进之外，还有一点就是后者的东东都塞进了“std”名字空间中。但唯独string特别。问题在于C++要兼容C的标准库，而C++的标准库里碰巧也已经有一个名字叫做“string.h”的头文件，包含一些常用的C字符串处理函数，比如楼主提到的strcmp. 这个头文件跟C++的string类半点关系也没有，所以g++并非g++的“升级版”，他们是毫无关系的两个头文件。要达到楼主的目，比如同时：

```
#include <string>. #include <string.h>. using namespace std. 或者  
#include <string>. #include <string.h>. 其中string.h是与C标准库的string.h相对应，但裹有std名字空间的版本。
```

笑谈（来自高质量）C++标准库很大。非常大。难以置信的大。怎么个大法？这么说吧：在C

标准中，关于标准库的规格说明占了密密麻麻300多页，这还不包括标准C库，后者只是“作为参考”（老实说，原文就是用的这个词）包含在C库中。当然，并非总是越大越好，但在现在的情况下，确实越大越好，因为大的库会包含大量的功能。标准库中的功能越多，开发自己的应用程序时能借助的功能就越多。C库并非提供了一切（很明显的是，没有提供并发和图形用户接口的支持），但确实提供了很多。几乎任何事你都可以求助于它。在归纳标准库中有些什么之前，需要介绍一下它是如何组织的。因为标准库中东西如此之多，你（或象你一样的其他什么人）所选择的类名或函数名就很有可能和标准库中的某个名字相同。为了避免这种情况所造成的名字冲突，实际上标准库中的一切都被放在名字空间std中（参见条款28）。但这带来了一个新问题。无数现有的C代码都依赖于使用了多年的伪标准库中的功能，例如，声明在`gt.`，`gt.`，`gt.`等头文件中的功能。现有软件没有针对使用名字空间而进行设计，如果用std来包装标准库导致现有代码不能用，将是一种可耻行为。（这种釜底抽薪的做法会让现有代码的程序员说出比“可耻”更难听的话）慑于被激怒的程序员会产生的破坏力，标准委员会决定为包装了std的那部分标准库构件创建新的头文件名。生成新头文件的方法仅仅是将现有C头文件名中的.h去掉，方法本身不重要，正如最后产生的结果不一致也并不重要一样。所以`gt.`变成了`gt.`，`gt.`变成了`gt.`，等等。对于C头文件，采用同样的方法，但在每个名字前还要添加一个c.所以C的`gt.`变成了`gt.`，`gt.`变成了`gt.`，等等。最后一点是，旧的C头文件是官方所反对使用的（即，明确列出不再支持），但旧的C头文件则没有（以保持对C的

兼容性)。实际上，编译器制造商不会停止对客户现有软件提供支持，所以可以预计，旧的C头文件在未来几年内还是会被支持。所以，实际来说，下面是C头文件的现状：旧的C头文件名如`gt.`将会继续被支持，尽管它们不在官方标准中。这些头文件的内容不在名字空间`std`中。新的C头文件如`gt.`包含的基本功能和对应的旧头文件相同，但头文件的内容在名字空间`std`中。（在标准化的过程中，库中有些部分的细节被修改了，所以旧头文件和新头文件中的实体不一定完全对应。）标准C头文件如`gt.`继续被支持。头文件的内容不在`std`中。具有C库功能的新C头文件具有如`gt.`这样的名字。它们提供的内容和相应的旧C头文件相同，只是内容在`std`中。所有这些初看有点怪，但不难习惯它。最大的挑战是把字符串头文件理清楚：`gt.`是旧的C头文件，对应的是基于`char*`的字符串处理函数；`gt.`是对应于旧C头文件的`std`版本；`gt.`是包装了`std`的C头文件，对应的是新的`string`类。如果能掌握这些（我相信你能），其余的也就容易了。特别推荐：2009年9月全国计算机等级考试时间及科目预告 2009年上半年全国计算机等级考试参考答案请进入计算机考试论坛 2009年全国计算机等级考试报名信息汇总 2009年NCRE考试有新变化 2009年全国计算机等级考试大纲 2009年上半年全国计算机二级考试试题及答案 2009年上半年全国计算机等级考试试题答案汇总 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)