

C与C 中标准输入实现方式上的一点区别 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022_C_E4_B8_8E_C__E4_B8_AD_c97_253815.htm 刚开始学C语言时遇到一个问题，如下代码：以下是引用片段：

```
#include int main() ...{ char a,b. printf("Please input the first character:"). scanf("%c", amp.b). printf("The two characters are %c, %c", a, b). return 0. }
```

程序运行结果如下：而下面这段程序却可以正常运行：以下是引用片段：

```
#include int main() ...{ char a. int b. printf("Please input the first character:"). scanf("%c", amp.b). printf("The two characters are %c, %d", a, b). return 0. }
```

运行结果如下：当时对这个问题很困惑，时间一长就慢慢淡忘了。昨天一个同学问到C运算符重载的问题，如下代码，其中的while循环起什么作用，好像注释掉后程序运行没什么差别：以下是引用片段：

```
istream& is, b& amp. is.get() != ) ...{ continue. } return is. }
```

其中，b为一个自定义的类型，其中含有一个char str[b::MAX]的私有成员，b::MAX为一个代表字符串长度的宏定义。可见

，当通过>>输入字符串的长度超过b::MAX长度时，while循环就起作用了，它会丢弃超过b::MAX长度的输入缓冲区的内容

，直到换行符出现为止，实际上就是一个清缓冲区的功能，

以待下一次重新输入字符串。为了验证我的分析，做了如下测试：以下是引用片段：

```
b b1, b2. cin >> b1. cin >> b2.
```

```
b1.show(). b2.show().
```

果然，当输入字符串小于b::MAX时一切正常，当超过b::MAX时，当输入完第一个字符串回车后，没

等输入第二个字符串，程序就把b1和b2的内容都显示出来了

，而b2的内容正是输入的字符串超过b::MAX长度的部分，说

明前面的分析是正确的。今天又想起这个问题，突然联想到以前C语言的两个scanf连用产生的问题，灵机一动，难道也是输入缓冲区没有自动清空产生的问题？当连续用scanf输入两个字符的时候，输入第一个字符后回车，结果换行符被当作第二个字符输入了，scanf可不会区分什么换行符还是普通字符，而当输入一个字符，再输入一个整数时，因为类型不匹配的原因，scanf会将换行符忽略而等待输入正确的整数后才返回，所以得到了正确的结果，哈哈，原来如此!!! 不禁又想，C中引入了新的iostream流库，它又是怎样处理这个问题的呢？于是做了下面的实验：以下是引用片段：`#include using namespace std. int main() ...{ char a, b. cout cin >> a. cout cin >> b. cout return 0. }`当输入a回车，输入b回车时，运行结果如下：当连续输入多个字符时，运行结果如下：可见，C中的标准输入流已经对于换行符做了处理，看来在C中利用标准输入处理特殊字符要自己做一些工作了 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