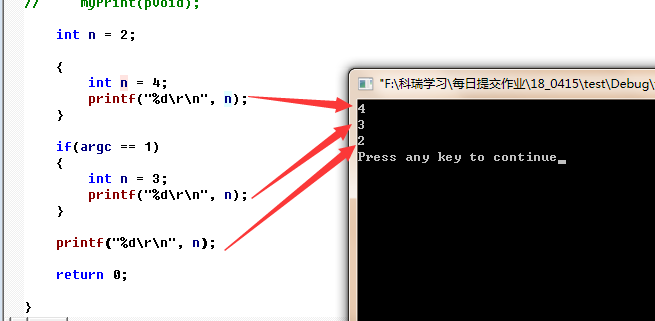
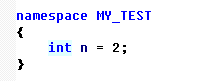
1. 名字空间
2. 内部作用域会屏蔽外部作用域，例：

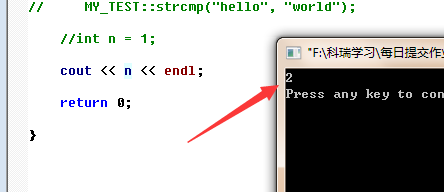


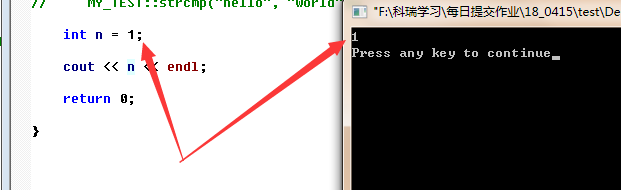


1. 名字空间作用域

当使用using namespace NAME时候，会提前将名字空间的变量释放出来，当变量名一样时会和自己定义的变量造成干扰。

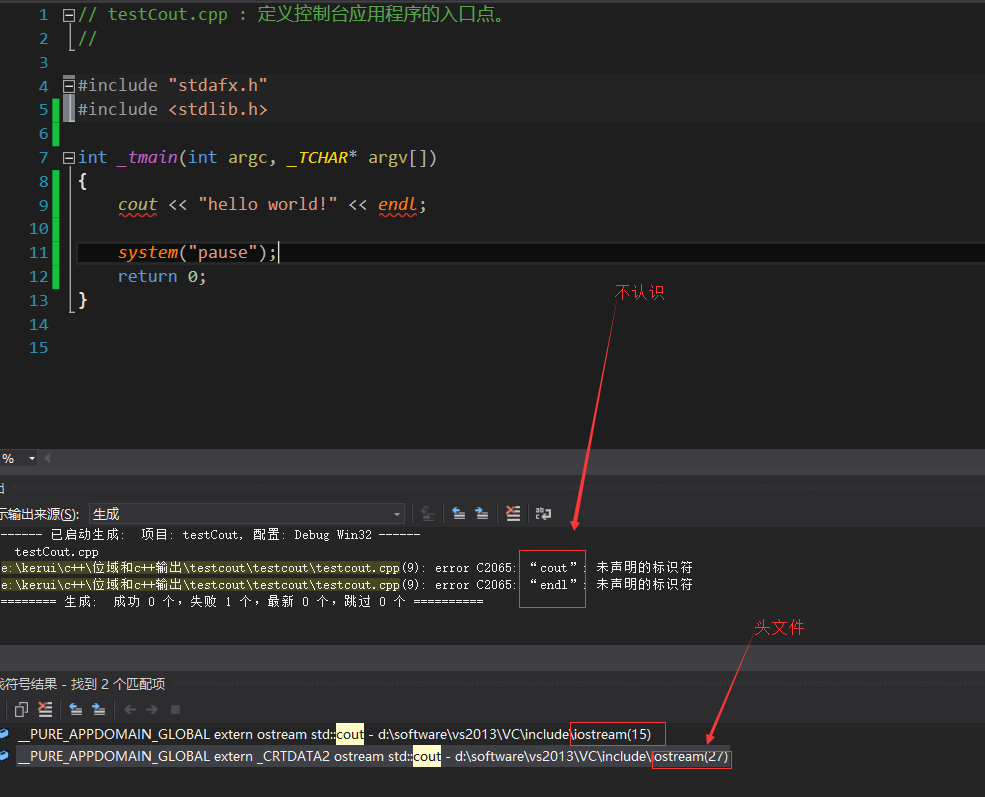




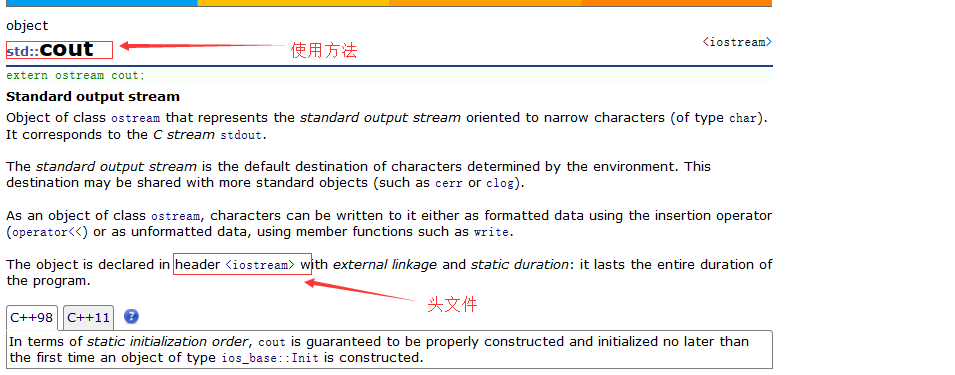


二.c++内容：cout

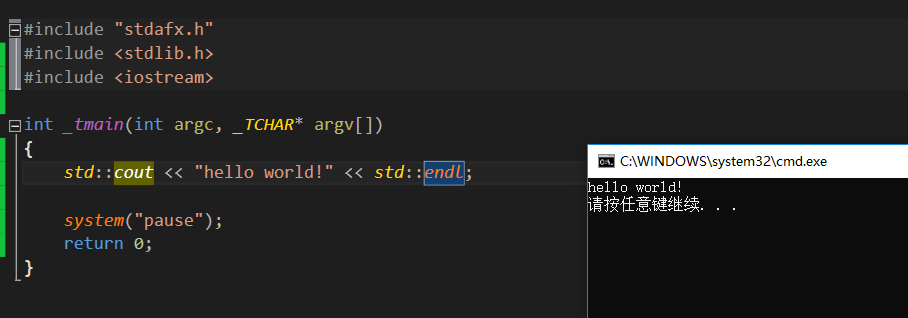
1.尝试输出"hello world!"



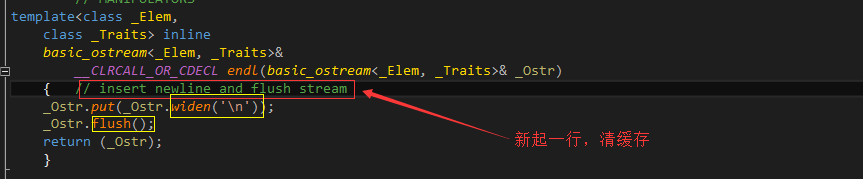
2.包含头文件



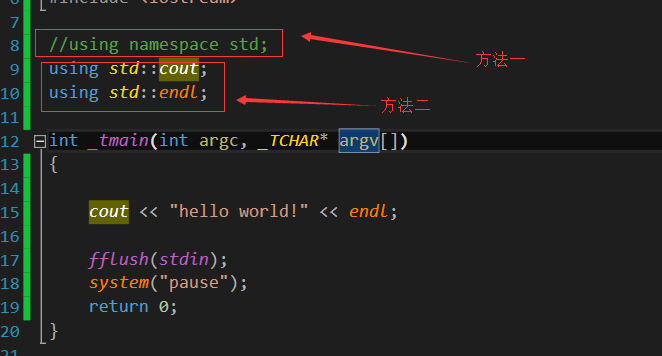
3.成功输出



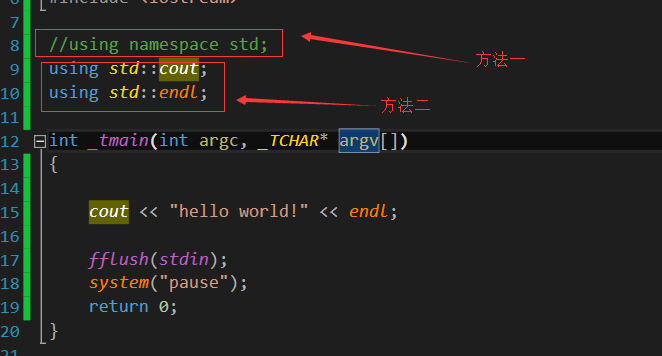
4.endl作用：换行(\n)



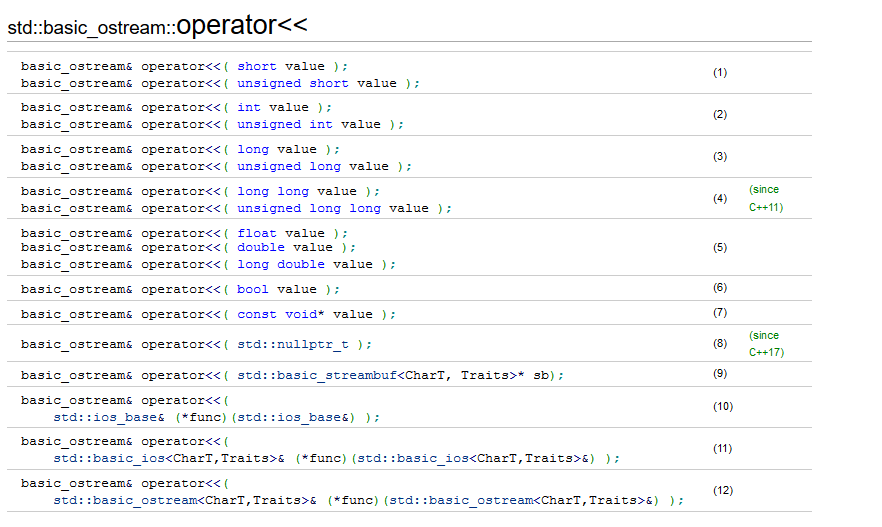
5.采用以下方法，也可以成功调用cout



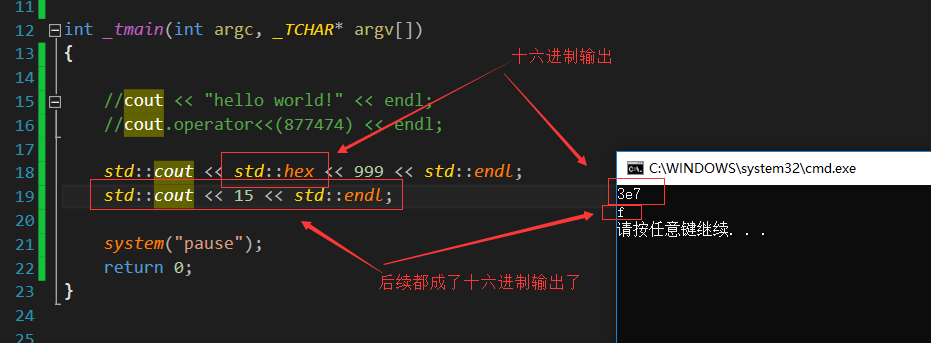
6.输出函数



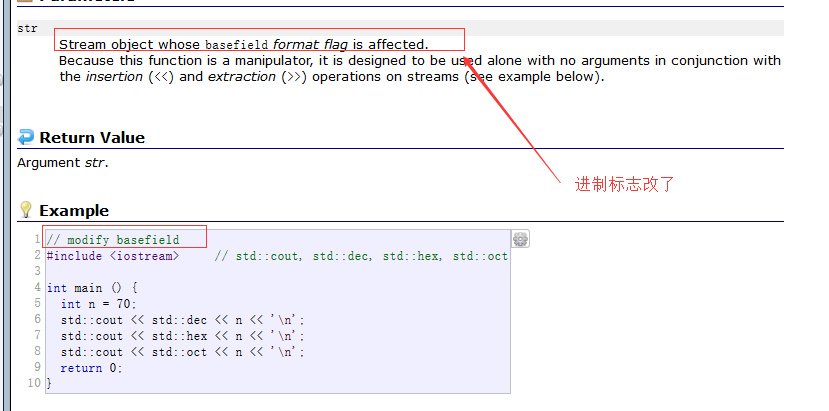
查看官方定义：重载，同意函数有不同类型的参数



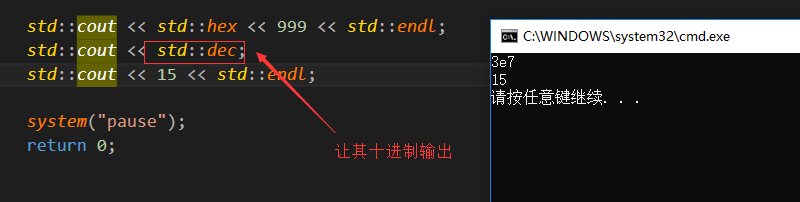
7.输出十六进制



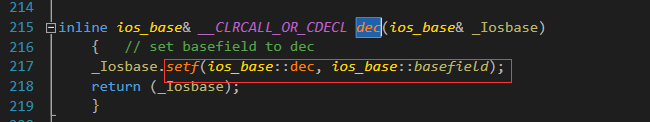
因为我们改变了进位标志：



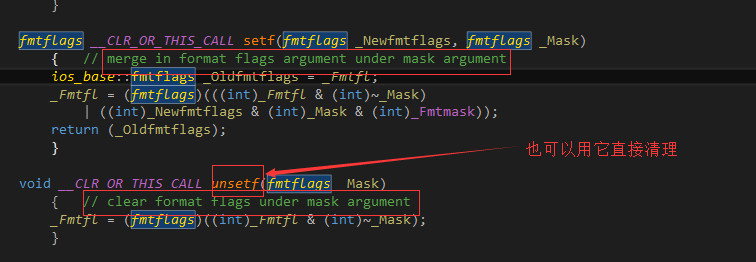
如何修改进位标志：



我们跟进去看下就明白了：



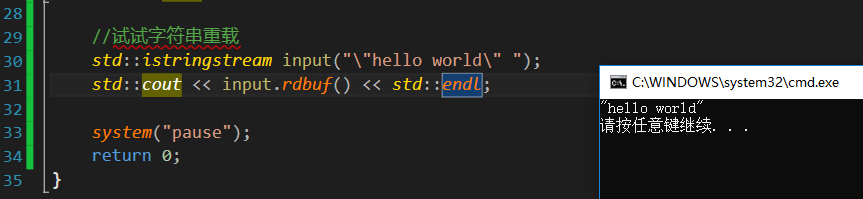
我们可以看下unsetf(）函数，是直接清理标志掩码的



例：



8.试下字符串重载

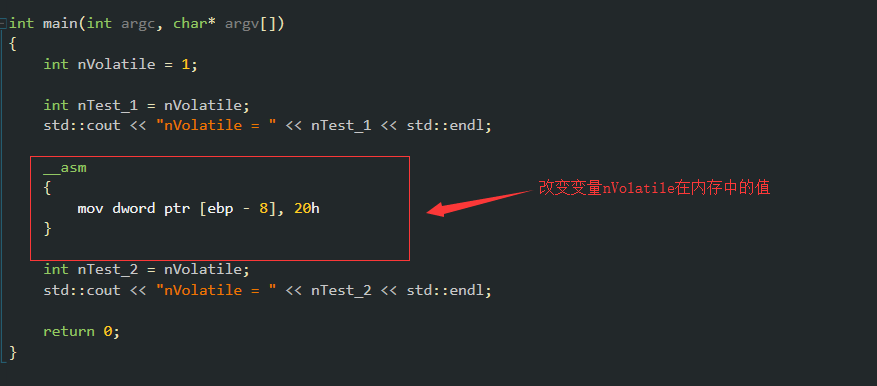


三．volatile

1.作用：禁止编译器优化

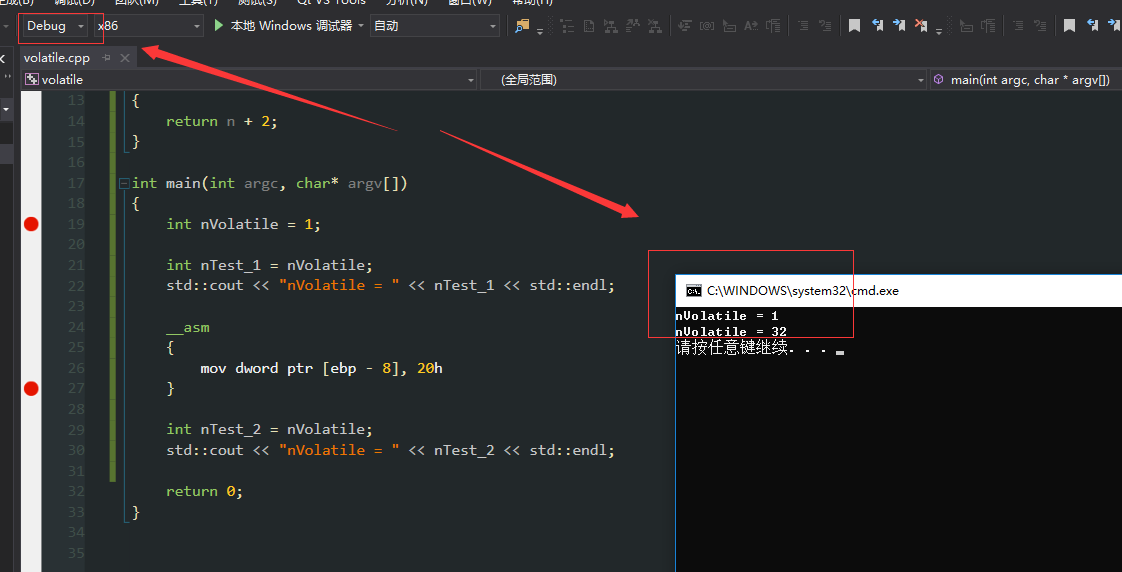
2.分析：

2.1 源码：

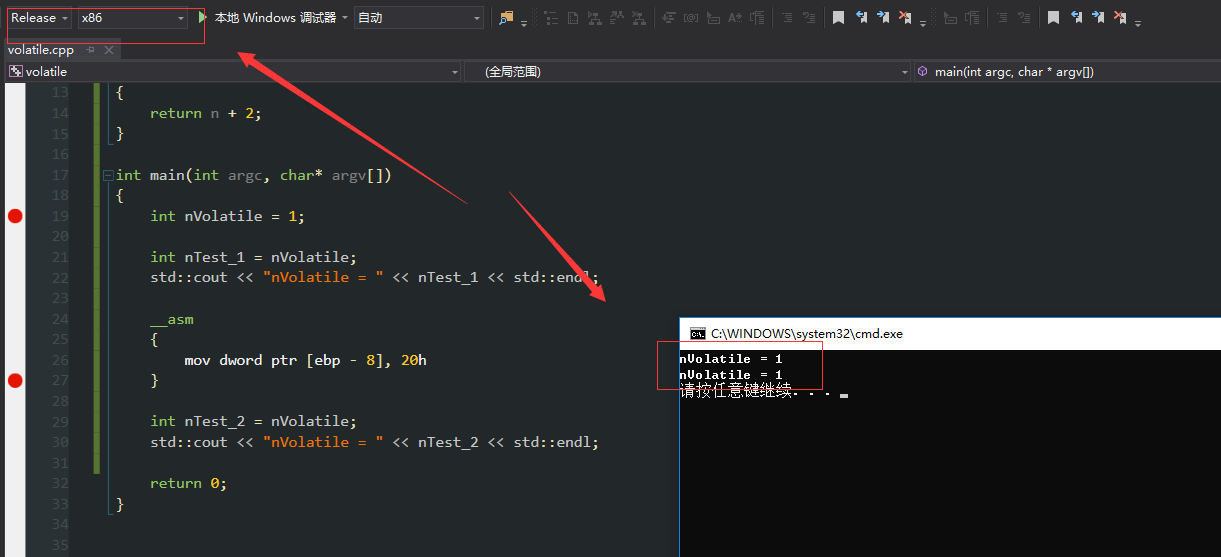


2.2当变量没有volatile修饰时

debug版本显示如下：

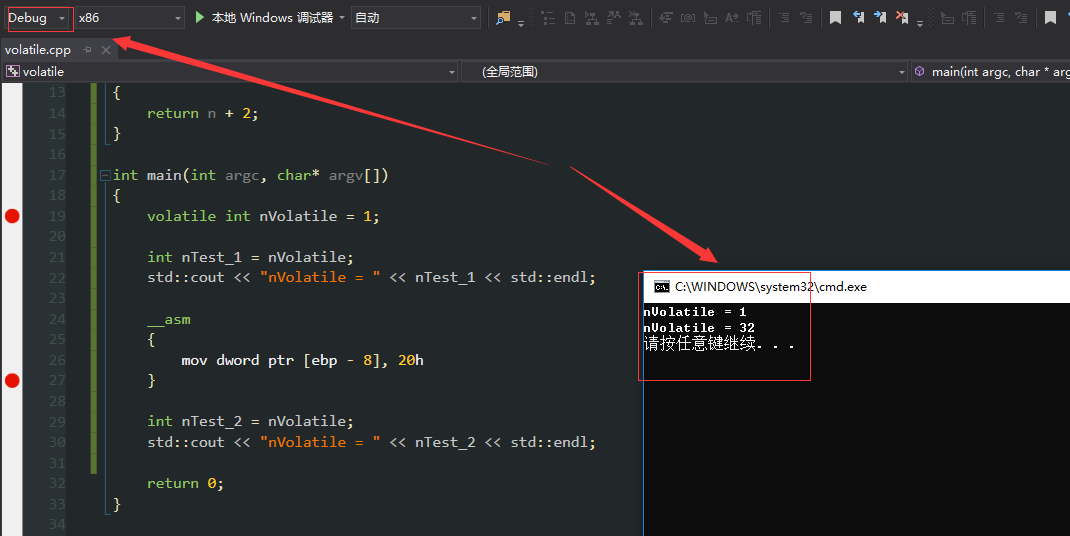


Release版本：

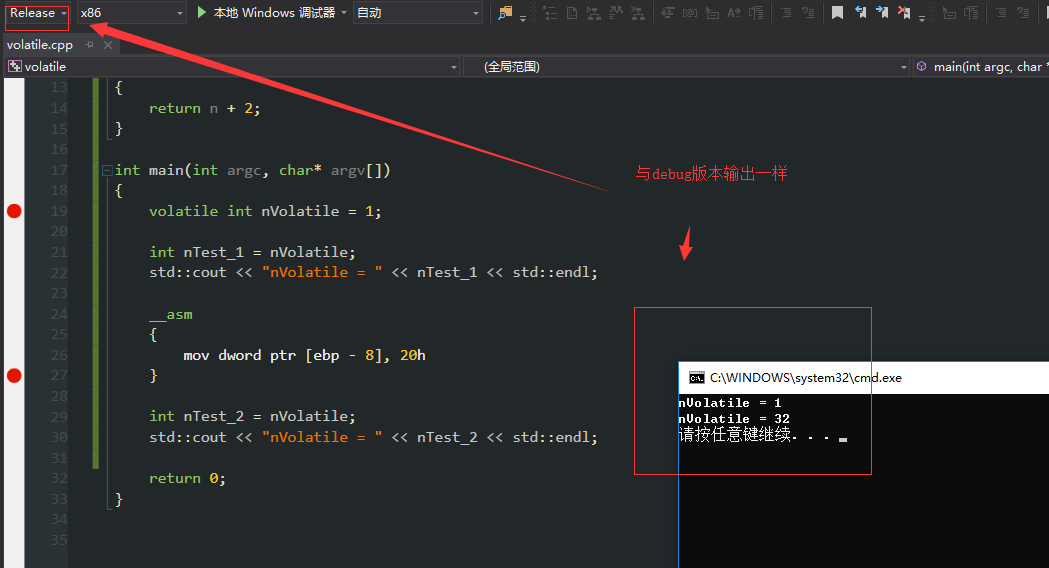


2.3 当变量有volatile修饰时：

Debug版本：



Release版本：



1. 结论：当变量用volatile修饰时，就是告诉编译器，该变量的读取必须到内存中去，不能从寄存器中取值