**实验环境**

**C语言实验环境：Code::Blocks 13.12、GNU GCC Compiler**

**Java语言实验环境：**

**java version "1.8.0\_60"**

**Java(TM) SE Runtime Environment (build 1.8.0\_60-b27)**

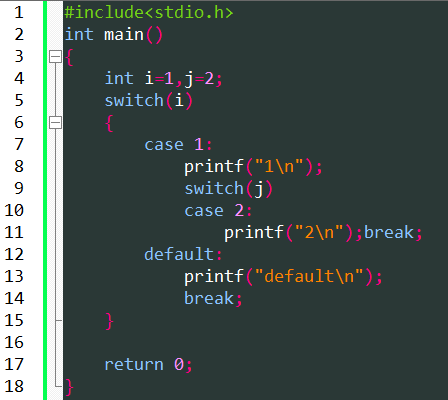
**Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode)**

**文本语言编写：Notepad++**

7.18 C/C++中，switch语句中的default分支相当于if语句中的else部分。在C/C++中是否存在与“悬垂else”歧义性类似的“悬垂default”二义性？请予以解释。

解答：

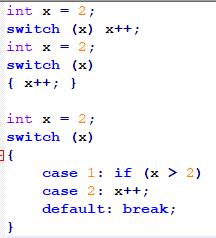
代码编写如下：



代码分析：

从代码结构分析，上述代码中default可能属于内层switch(j)也可以视为属于外层switch(i)，这样就有两棵不同的分析树，导致具有二义性。

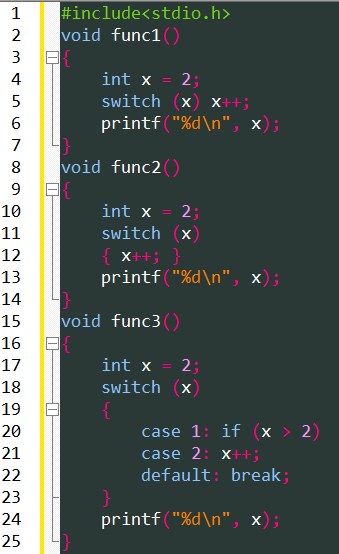
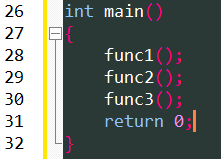
7.30 根据C/C++/Java的语法，有下述三个可能的switch语句：



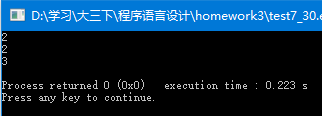
（a）在C/C++中哪一个是合法的？指出执行完合法的switch语句后x的值？

解答：

代码编写如下：

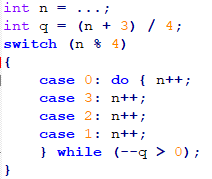
程序运行如下：



实验结论：

实验依次输出2，2，3，三种语句都是合法的。

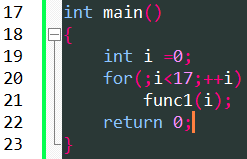
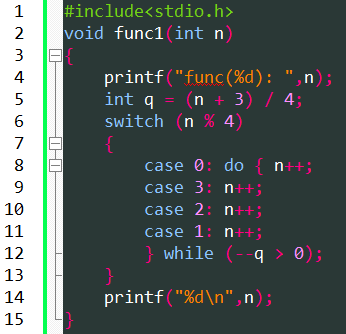
7.31考虑C/C++中如下合法的switch语句：



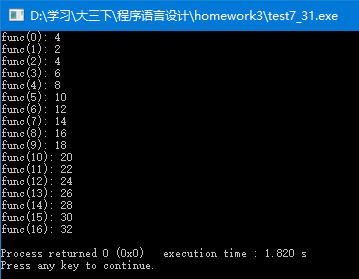
假设n取值(a)0；(b)1；(c)5；(d)8。在上面的switch语句中，求各个case执行后n的值？当这段代码执行后，你能总结出n取值的一般规律吗？如果可以，说明该规律：如果不可以，请解释原因。

解答：

代码编写如下：



程序运行如下：

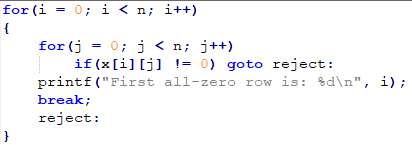


实验结果：

当n == 0时，执行后n = 4。

当n > 0 时，执行后n = 2 \* n。

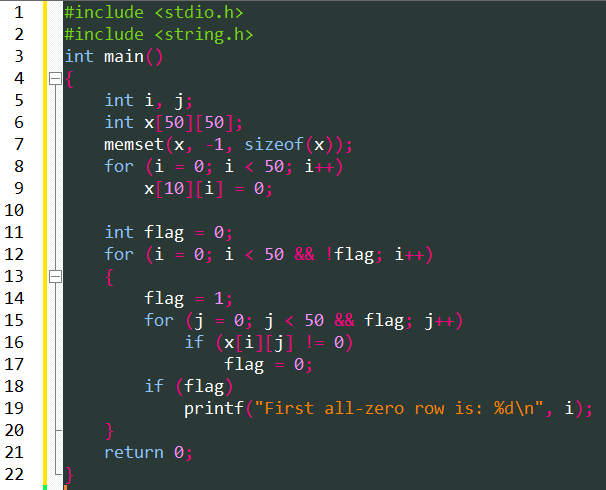
7.34 Rubin[1987]使用下面的例子“证明”了GOTO基本上能使代码清晰简明:



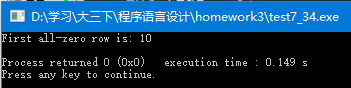
假设这段程序要在矩阵x[n][n]中求第一个零行，其中n >= 0。

解答：

1. 不用GOTO和break语句，重写这段程序，代码如下：

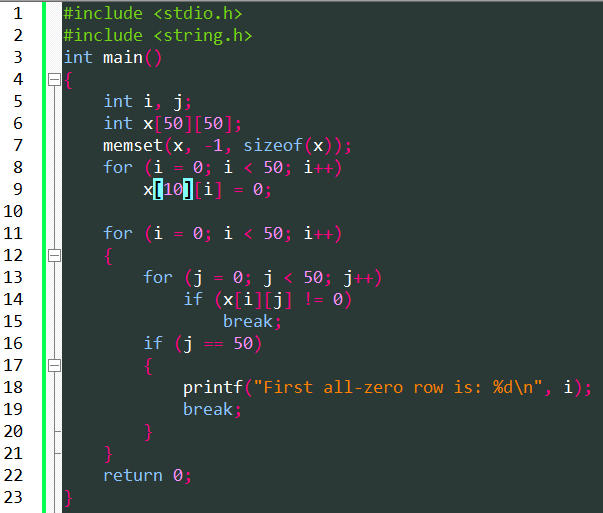


程序运行结果如下：

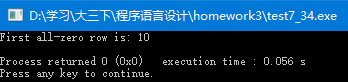


第一个0行为10，结果正确

1. 不用GOTO语句，但可用break或continue语句，重写这段程序。用break或continue语句，重写这段程序，代码如下：



程序运行结果如下：



第一个0行为10，结果正确

1. 你同意Rubin的观点吗？为什么？

答：不同意，goto语句可以任意地破坏程序的顺序结构。在阅读，维护和重用时，使编写代码变得困难。

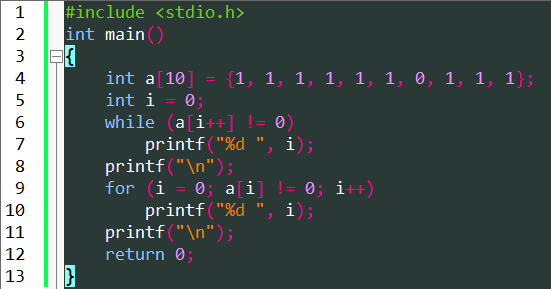
7.37 C++和Java的循环经常使用空循环体而将所有副作用放进测试之中，如下例:



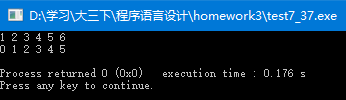
（a）这些循环等价吗？（b）使用这种形式的代码有什么优点或缺点？解释其理由。

解答：

代码编写如下：



编译运行结果：



实验结论：

从输出看，这两个循环是不等价的，区别在于，这里while循环结构主体代码运行前，i自身就已经+1；而这里的for循环结构主体代码运行后，i才进行+1操作。

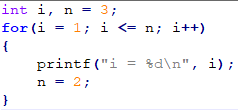


优点：编写代码灵活，简单。

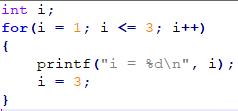
缺点：代码可读性较差，维护，修改，重用更麻烦。

7.38 找一个翻译器测试下面的C/C++代码段（或者用其他语言重写，并测试），解释其功能：

(a)

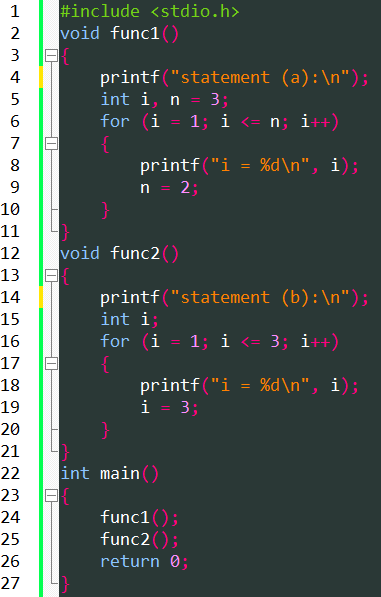


(b)

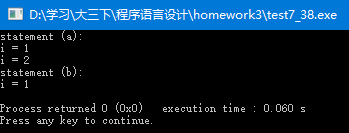


解答：

重写代码如下：



编译运行结果：



测试结果：

前者在循环体中运行了两次，而后者在循环体中只运行了一次。