**课程实践作业四**

1. **翻译**
   1. **当你逐渐觉得困难**

据说肯尼迪总统的父亲在教育他的孩子时曾说：“当你逐渐觉得困难，困难也正远离你而去。”但他从未调试过程序。下面的段落包含一些当你在调试程序遇到问题时的一些实用的注意事项。

寻找常见的错误，比如，你有没有

1. 将对象传递给函数时弄错了顺序；
2. 拼写错误，比如将大写拼成了小写；
3. 变量初始化失败；
4. 检查两个浮点型变量是否完全相等而非近似相等（记住浮点型的运算不同于你在学校里学到的运算）；
5. 想要两个对象相等时确保他们数值相等（例如，想要id(L1) == id(L2)，列出它们的表达式并比较L1 == L2)）；
6. 忽略有些内联函数具有的副作用；
7. 忽略括号可以将对函数对象的引用变为调用；
8. 无意的创造了一个别名；
9. 或者犯了一些你常犯的错误。

不要再质问自己为什么程序没有按照你预想的那样运行，相反，多想想它为什么会这样运行，这样会更容易回答一些，而且还会很好的启发你如何去修复这个程序。

记住错误很有可能出现在你没想到的地方。如果它真的在那里，那你很早之前就会发现它了。一个比较实用的方法是想想错误不可能出现在哪，就去那查查看，正如福尔摩斯所言：“排除所有的不可能，剩下的无论多么的不可思议，都是真相。”

尝试将问题讲给其他人听。每个人都有盲点，将问题阐述给其他人听很可能会让你发现自己忽视的地方。将你认为错误不可能出现在某些地方的原因讲给其他人听是个不错的做法。

别迷恋书本，特别是各种文档资料，毕竟代码并不一定按照注释里设想的那样运行。

暂停调试，做些记录。这会让你从另一个角度看待问题。

出去走走，明天再说。相比于潜心钻研，也许这样会拖延你的进度，但却节省了大量的时间，提高了效率。（同学们，这就是为什么要越早开始编程练习越好）

**2.2当你找出问题所在**

当你觉得自己找到了代码里的错误，那你就会难以抑制的想去继续编写并测试它。但最好还是先等等，记住你的目标不仅仅是修复一个错误，而是尽可能快而有效地完成一个没有错误的程序。

问问自己这个错误解释了所有的问题，还仅仅展示了冰山一角。如果是后者，最好多想想这个错误与其他变化之间的联系。比如说你发现错误的原因是不小心改变了一个列表，你可以局部的避开这个问题（比如备份列表），或者用元组来代替列表（因为元组是不变的），也许这样你可以消除代码中类似的错误。

在做出任何改变之前，先了解这样的改变可能引发的其他问题。它会破坏程序的其他地方吗？还是会让程序复杂化？有没有可能让程序的某些地方更加简洁明了？

一定要确保你可以回到修改之前的状态，再没有什么比发现越改越糟还不能回到初始状态更让人泄气的事了。磁盘空间很充足，别忘了修改前备份程序。

最后，当你发现有很多难以解释的错误，你就该想想逐一排错是不是正确的方法了。思考一下有没有更好的解决办法或者更加简介的算法，这样或许情况会好很多。

1. **体会**

好像又回到了最初接触C++语言的时候，不过这次并没有完全不知所措。编写程序很困难，但只要有了思路多挣扎一会还是会有办法的。最纠结的还是调试程序，看着那几个改来改去也不知道在哪里的bug，真纠结。看了这篇文章，豁然开朗：

1. 拓展思路，摆脱思维定势的魔咒。
2. 平时注重小细节，免得它在程序里“大展身手”。
3. 学会交流，听取别人的意见。
4. 休息一会，放松一下。
5. 一定要在修改前做好备份！否则，那很有可能会演变成一个悲伤的故事。