作业四

1. **翻译**

6.2.3 当事情变得棘手的时候

肯尼迪总统的父亲约瑟夫·帕特里克据说曾这样教育过他：“当事情变得棘手的时候，艰难得走下去。”(36)但是他从来没有进行过软件调错。而这部分又恰恰包含了调试代码进行得艰难时我们所要做事情的小提示。

看看一些常见的错误。例如有：

函数的参数传递顺序不对，

拼错名称等等，把大写字母打字打成了小写字母，

没有初始化变量，

检测两个浮点数相等而不是近似相等（记住浮点数算法和以前在学校里学习的算法不一样），

测试数值相等（例如通过写下L1==L2来比较两个列表）当你打算测试对象相等时（例如id(L1)==id(L2））,

忘了一些内置函数有副作用,

36:据说他还告诉肯尼迪：“不要去买不是很必要的单独一票。如果我为了压倒性胜利去出钱，我觉得我就该死。”

调用另一个类型函数的引用对象时忘记（），

创建了一个无意的别名，或者

其他典型的对你来说容易犯的错误。

不要问你自己为什么程序没有按照你原来预想的那样运行。相反地，问你自己它为什么会这样运行。那应该是一个容易回答的问题，也有可能是你怎样调试这个程序的关键性的第一步。

请记住，错误可能不会出现在你想要出现的地方。如果你那样的话，就可能在很久之前就已经发现它了。一个可行的办法是，去看看那些你觉得错误不可能出现的地方。正如夏洛克·福尔摩斯说的那样，排除所有的其它影响因素，那么最终保留的那个一定就是真相。

试图向别人说明你的问题。我们每个人都有自己成长的盲点。仅仅尝试向别人说明你的问题也会引导你看见一些你原来错过的东西，这种事很常见。试图去解释一些错误为什么不出现在确定的地方是一件好事。

不要相信你所阅读到的所有东西。特别是，不要相信文档。代码可能不会像评论建议的那样运行。

停止调试，开始编写文档。这会帮助你从一个不同的观点去接近问题。

走开，明天再试一次。这可能意味着错误会比你一直忍受着它迟些解决，但是你很有可能花更少的时间去寻找这个错误。这就是，用延迟时间来换取效率（同学们，这就是你们要早点做习题的最好的原因）。

6.2.4 当你发现错误后

当你认为自己已经在代码中发现了一个错误时，你会抵挡不住想要马上去修改代码调试错误的诱惑。然而，更好的是不要那么着急。你要记住，我们的目标不是修改程序中的一个错误，而是让它快速有效作为一个没有错误的程序运行。

问问你自己这个错误能不能解释所有观察到的症状，或者只是冰山一角。如果是后者，先想想这个错误是不是和其它变化相一致，这样会更好。假设，例如，你发现的错误是不经意修改了一个列表。你可以本地避免这个错误（通过复制这个列表），或者你可以考虑用一个元组来代替这个列表（因为元组是不可变的），可能会消除掉代码中其他类似的错误。

在做任何修改之前，努力想想这样修改之后会产生的后果。会不会破坏其他代码？这样会不会更加复杂？会不会给整理其他代码提供机会？

确保你能回到初始地方。没有什么比你修改了一系列代码后比没修改之前离目标更远，而且回不到初始位置时更痛苦。磁盘空间通常是富足的。用它来保存你程序的旧版本。

最后，如果还有许多不能解决的错误，你可能就需要考虑是否像这样一个一个错误的慢慢修改了，即使是正确的方法。也许你应该思考一下是否有更好的其他方法来组织你的程序，或者其他更简单的算法让你程序成功运行。

1. **小结及感想**

说实话，我的英语水平不是很好，所以翻译得应该也不是很好，但是我觉得我翻译的东西都能看懂就行了，当然不排除里面肯定会有一些错误，也许意思完全变了，但没办法，我只能做到这个程度了。其实翻译上面文字时，想到了大一学c++的时候，有些时候上机写完程序，觉得自己写对了，结果一运行，发现有错，而且还完全不知道在哪，有时候越改错越多，如果是在大一的时候看见这段话，我觉得我的编程水平应该会有所提高。