**翻译作业**

1. **翻译部分**
   1. **当事情变得艰难的时候**

据说肯尼迪总统的父亲在教育他的孩子的时候曾经说过，“当事情变得艰难的时候，意志坚定的人将继续前行。”可是他却从来没有调试过程序。下面的段落中包含了一些对调试过程变得艰难时我们应该如何做具有实用意义的线索。

寻找常见的可能犯的错误，例如：你有没有

1. 在将参数传递给函数时弄错了顺序；
2. 拼错其中的名词，例如：在你应该输入小写字母的时候你却输成了大写；
3. 重新初始化变量失败；
4. 测试两个浮点值是否完全相等而不是约等于（记住，浮点运算和你在学校里所学的运算是不一样的）；
5. 在你想要两个对象相等时（例如：id（L1）==id（L2）），检查它们的数值是否相等（例如：通过写出两列表达式L1==L2来比较）；
6. 忘记一些内置函数所具有的副作用；
7. 忘记“（）”会将一个对函数类型对象的引用转变成函数的调用；
8. 创建一个没有定义过的对象；
9. 犯一些你自己常犯的错误。

不要再一直问自己为什么程序没有像你所想象的那样运行，相反的你应该问自己为什么程序会像现在这样运行。那可能会是一个更加容易回答的问题，并且在找出修复这个问题的办法中这可能会是最好的第一步。

记住错误也可能并没有出现在你认为它应该出现的地方。如果它真的出现在了那儿，那么你应该早就找到了它。一个决定从哪里开始寻找错误的最有实用性的方法是问自己错误最不可能出现在哪里。就像福尔摩斯所说，“消除所有其他的可能，剩下的那一个就是真相。”

尝试着向别人阐述你所遇到的问题。我们每个人都有盲点，很多时候在向别人阐述自己的问题的时候可能会帮助你发现自己所遗漏的东西。一个最好的方法是尝试着去解释为什么错误没有发生在你所认为应该出现的地方。

不要相信你所读过的任何东西，尤其是那些文献资料。你的程序也许并没有按照意见中所说的那样运行。

停止调试并开始写下其中的一些问题。这会帮助你从另一个角度寻找到解决问题的方法。

停止调试并离开，明天再继续尝试。这可能意味着解决这个错误的时间会比你一直坚持下去要来的晚，但是你可能会花更少的时间来寻找错误。这也就是说，我们也许可以用等待来换取效率。（同学们，这是一个很好的早一些而不是更晚去做编程错误集的理由！）

* 1. **当你找到错误的时候**

当你认为你找到了程序中的一个错误时，那种想要马上开始编程并调试修改的冲动是几乎不可抗拒的。可是，实际上最好是能放慢一点。记住最终的目标并不是去修补好一个错误，而是要更快更有效率的向没有错误的程序前进。

问问自己这个错误是否能解释所有可观察到的症状，或者它是否只是冰山一角。如果是后者，那么你最好在考虑修改这个错误的时候要保持它和其它的变化相一致。举个例子来支持这个观点，你已经发现了错误的原因是因为一不小心改变了一个列表，你可以局部地避开这个问题（也许是通过将列表备份），或者可以考虑用元组来替代列表（因为元组是保持不变的），这样也许能够消除在代码的中的其他部分出现的相同错误。

在做任何改变之前，尝试着去思考这个修改可能衍生出的其他问题。这样的修改会破坏其他地方吗？它会使程序变得过于复杂吗？它是不是能为整理好程序中的其他部分提供机会呢？

你要永远确保自己能够回到修改之前。没有什么比在意识到自己做的一长串的修改反而使你离目标越来越远并且没有办法能再次回到你最开始的地方更加让人沮丧。磁盘的空间通常都很充足，所以记得用它来储存你老版本的程序。

最后，如果出现了许多难以解释的错误，你也许应该思考一下逐个找到错误并修改是不是正确的方法，思考一下是否有更好的解决问题的方法或者一些能更加轻松正确使用的算法，这样也许会使你的情况变得更好一些。

1. **个人体会与总结**

“当事情变得艰难的时候，意志坚定的人将继续前行。”我记得这句话是老师第一次上课的时候就跟我们多次强调的，换句话来说也就是越挫越勇吧。其实在刚开始学c++的时候个人就比较害怕，常常在编程的时候觉得自己的逻辑不够清晰，头脑有的时候转不过弯来，不过一年的学习下来感觉自己越来越不害怕，也慢慢能自己把程序编起来。现在对于软件工程这门课，我想也是一样的吧，熟能生巧，自己多去琢磨、多去尝试，总能不断总结经验越做越好。

这两篇文章中有三点也是我自己比较有感触的地方：

1. 对于许多在编程中常犯的小错误，这些小错误往往比大方向上的错误来的更加可怕。通常来说，逻辑上的错误总是很容易被人一眼看出，而细节上的小错误，一旦在开始就被忽视的话，后面检查的时候就很难再被发现，就比如字母的大小写、“==”的使用等等。这也就提醒了我们，在编程的时候一定要注意细节，细节决定成败，只有确保每一个小点都正确才能写出一个完整并且正确的程序。
2. 在编程的时候一定要确保自己能回到最初的地方。这一点在大一编程的时候就有较深的体会，因为以前常常就是因为不注意，直接在程序上修改也没有把以前做的保存起来，导致程序越改越乱，越改报错越多，自己也越来越痛苦。所以对于自己每次修改运行的程序一定要保存起来以备不时之需。其次，在编程的时候，我觉得最好能在每条语句后面加上相应的注释，我认为这是一个比较好的习惯，一旦报错，也可以较为容易地帮助自己找到逻辑上的错误
3. 关于在调试程序时长时间没能成功就应该先将这个工作放一放，隔一定时间之后再去尝试。这个方面其实应该是大学以前就有所体会了，以前遇到比较难的数学题，花很长时间都没能解出来的时候，我就会选择换另一件事来做，毕竟长时间思考一个同样的问题，思维容易禁锢，这个时候做点别的事放松或者看看别的资料后再回来看，有的时候会灵光一现有不同的体会，对于编程的调试我想也是如此。适当的放松，有的时候也许会带来新的思路与突破。