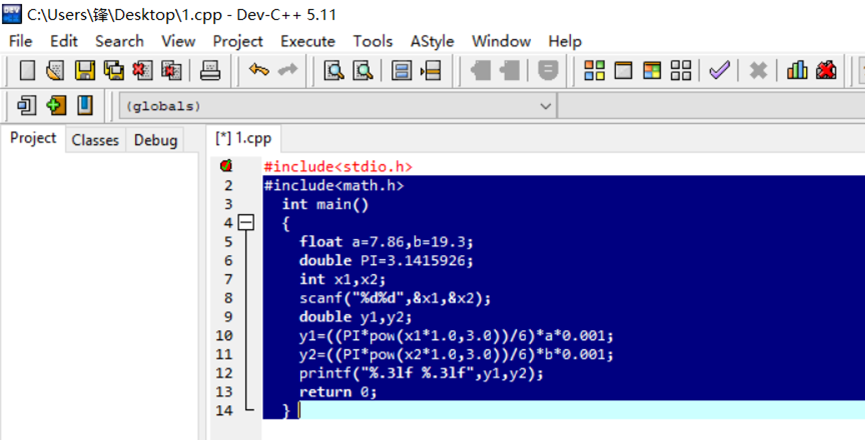
**计算机导论与程序设计报告**

通过一个多月的学习，我对C语言程序设计有了一定的理解，其实C语言就如同其它语言一样。对于初学C语言的电脑小白而言，要想去学好C语言，就要真正的把它当成一门语言来学。就像学英语一样，英语有自己的语法规定，词汇搭配……，C语言也是如此，要按照一定的格式要求才能编写出正确的程序。但是C语言有一点却比英语难得多——有时候你可以用几个简短的单词让人明白你的你的意思，但C语言不能，因为它是非对即错的。

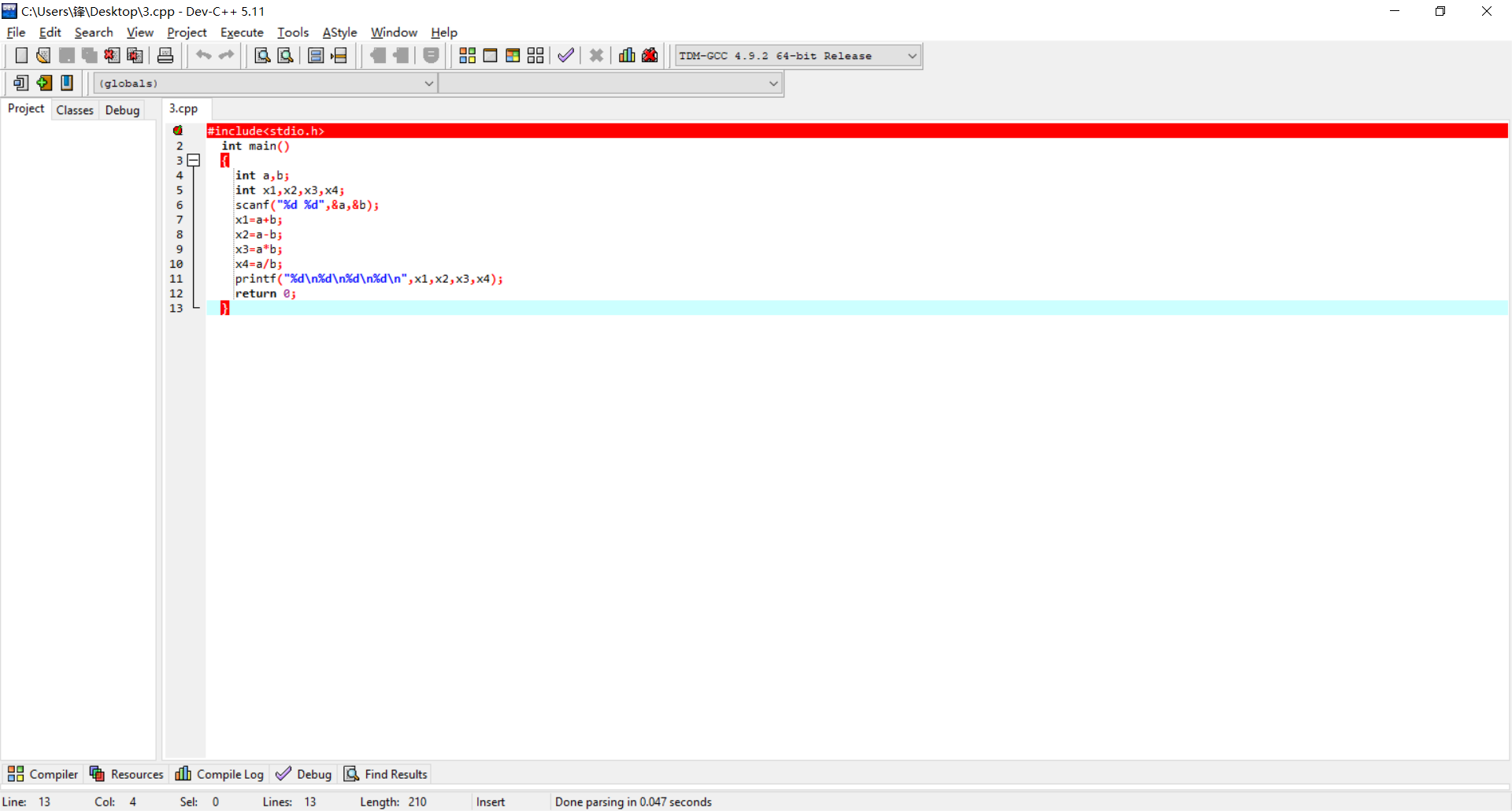
如果只是单纯的学习知识，像是背英语语法知识那样去学习一门编程语言、背下代码规则(当然熟识理论知识是必须的)，这样可能也可以初步懂得这门语言，但是在深入上却较为困难。我始终觉得学习编程,重点在于培养写代码的能力。就拿写文章来说，对于文字的认识，可能大家都懂，但是给定一个题目，写出来的文章质量就各有差异，有的人文章平淡无奇，而有的人的却是宏篇大论。编程也是如此，懂得了理论知识，就像是认识了文字，但是优秀的作家能将这些文字巧妙地组合在一起形成佳作，优秀的程序员也是如此，能将大家都懂得的理论知识以一种更为巧妙的方法通过代码展示出来并使之可执行

我还记得刚上第一节课的时候, 我是满怀信心的去听的,到最后却听的一头雾水。随着课程的深入,我发现自己越来越跟不上老师的进度, 这让我感到很心慌。我觉得自己应该静下心来好好学学, 然后我就从第一章开始慢慢的看, 通过反复的看,我慢慢的跟上了大家的进度。

以下是我关于在上机过程中遇到的一些问题以及心得。

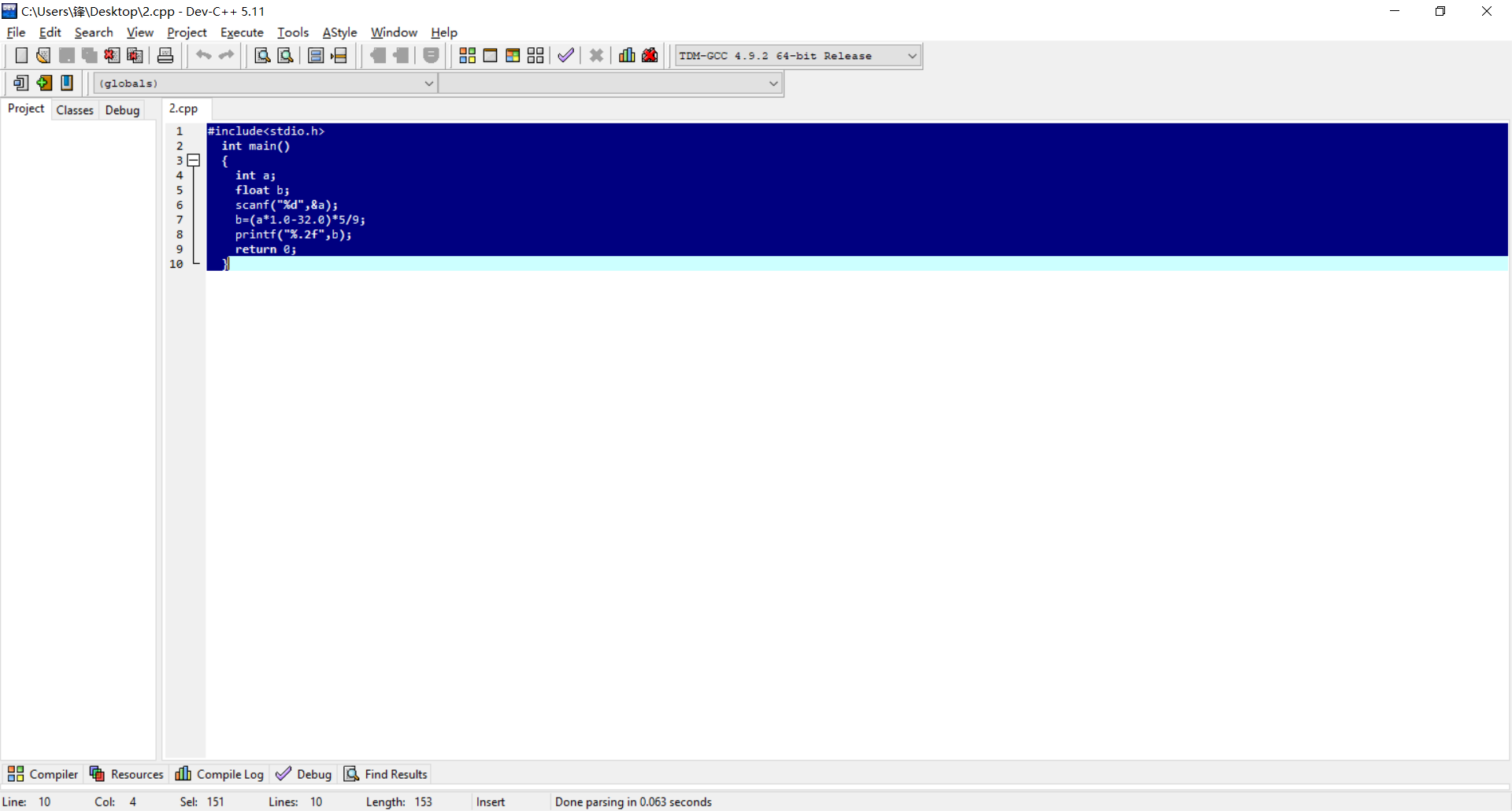
由

1. 于前几节课没有认真听老师的话，不知道类似Π这样的常量时，可以在程序的开头用define来定义
2. 在使用pow、fabs等数学函数时，需要添加#include<math.h>
3. 运用程序来计算时一般不要化简公式，因为这不仅失去了编写程序的本意更会造成计算结果精度的损失



这道题让我想起了老师上课时直接在printf内进行操作的示范，同样这道题也适用，我们同样可以直接舍弃给那么多结果赋值，并且这样还能减少定义的数使得程序设计得更加简洁明了

还有让我印象深刻的问题就是精度问题和分号以及算法的选择了

第一次上机就遇上了精度这个比较常见的问题，例如

第一次的时候写的是(a-32)\*5/9,后面在旁边同学的指导下才想起两个整数类型相除结果也是整数。还有印象较深的精度问题就是老师上课所讲的a\*b\*c和a\*c\*b虽然在32位编译器结果相同，但在64位编译器中结果却不相同。所以说，在运算浮点数时不能随意改变运算顺序，否则会出现精度问题使得程序不符合机试的要求。因此无论是上机还是平时在寝室自己练习都要注意精度问题并且养成良好的习惯。

其次就是分号了。

在第一次上机以及到目前为止我经常会在（）的后面不由自主地加上分号。虽然说删一个分号并不是一件麻烦事，但偶尔忘记删分号，却在运行的时候才发现多了个分号着实令人恼火。这里我也根据我的一些经验总结了一些分号问题。

不要加的主要有int main()、if()、for()、while()……这些后面不加分号。

另外关于算法的选择，比如说if语句和switch语句,这两种条件语句都是用来执行判断功能的，那什么时侯用if什么时侯用switch,如果很好地了解它们的结构和作用的话那么就能灵活运用：若它的条件分支是多个而且条件的值是一个确定的值时就可以选用switch。反之，在大多数情况下就用if。但掌握了语言法规则，不一定就能编出程序来。其实学习编程就是要学习一种思想，一种思维方法，首先要确定一种算法,如从1加到100,判断润年等,我认为这些算法都是一种思维方式，我们要尽量在日常的练习中逐渐拥有自己的思路，多想想一个程序还可不可以有另外一种方法来实现它并且如果想到了那就去做，写程序切忌半途而费，想到一个思路，就要按自己思路把它写完。如果实在不行就去问问老师，就算错了，也可以让自己了解为什么这样做会错，错在哪里，会出什么结果，以后自己也很难在这个地方犯同样的错误。

其它的问题例如scanf内要用取地址符意识老生常谈我也不多加赘述了。

然后是我关于怎样去学好C语言的一些想法

1、一定要多练习，多“敲”代码

编程是一门实践性的学科，绝对不是理论。如果不动手“敲”代码的话，永远都学不会编程。很多问题只有在“敲代码”的时候才能发现，才会有更加深刻的体会、领悟和理解。而不是靠死记硬背书中的注意点，那样真的很痛苦。我在学习编程的时候从来都不会刻意记忆什么注意点，这些知识点都是在不停“敲代码”的过程中，自然而然深入我的脑海。

2、一定要多记笔记

编程需要不断地积累。有时候我们在编写一些复杂的的程序时经常会烦恼要写这么长的代码而别人或老师却能用简短的语句就能起相同的效果，这时候我们就应该记点笔记。我们一定要学会模仿别人优秀的代码、优秀的算法，然后将它记下来。站在巨人的肩膀上学习我们才能学的更高效。但是我们的记忆能力是有限的，时间长了难免会遗忘，所以一定要学会记笔记。一有心得、体会、感悟也可以写下来，这些都是很珍贵的。我们在记笔记的时候，如果眼前没有计算机则可以先写在纸上，但事后一定要将它整理成电子版。整理成电子版看起来会很方便、舒适，还可以随意地增添和删改，保存时间也长。