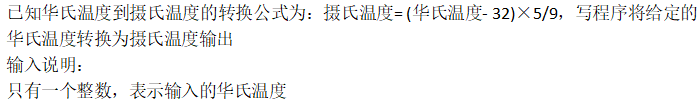
上机报告

2020级 计算机导论 自动化大类 张赫

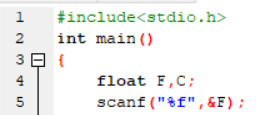
学习了几周计算机导论后，我们终于于2020年10月11日进行了第一次上机练习。第一次上机练习主要以实践操作为主，理论学习为辅。主要练习内容为简单程序的编写、调试、编译与运行。

在练习过程中，我深刻体会到了C语言的强大泛用性与灵活性。虽然我只学习了一些简单的C语句和简单函数，但这些看似简单的知识却足以解决许多复杂的问题，甚至大部分时候效率能够远超人工计算。但同时我也发现了C语言实际使用时的一些不便，例如浮点数精度不足、格式错误导致的无法运行等。以下是一些我的实际体会：

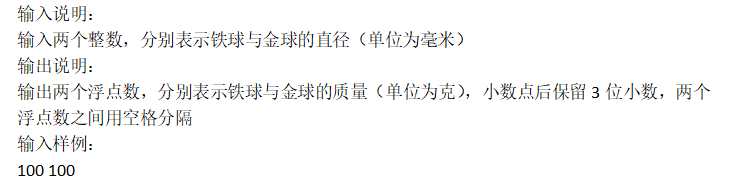
1.如图，在第二题《温度转换》中，我此时并未了解到精度不足带来的问题。



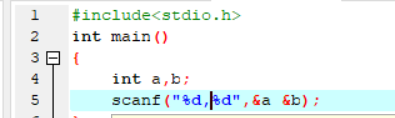
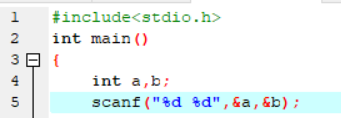
题目所给条件为输入一个“整数”，而我当时在学习了不同数据类型之后认为浮点数作为可以精确到小数点的数据类型一定会比整数类型更加精确。因而我多此一举地写作：



如图所示，将输入的F以浮点数记录。在经历无数次运行无误却提交错误之后我也终于意识到数据类型的重要性。浮点数作为可以精确到小数点后几位的数据类型与整数相比免不了会产生四舍五入之前或之后数据产生差异的问题，应时刻注意输入、输出值的数据类型，而不是一味追求盲目的精度。

1. 如图，在第一题《计算球体重量》中，

题目要求输入两个整数。而在之前的自行训练中我所使用的将两个输入值分隔开的方法均是使用逗号，在遇到使用空格时我反而束手无策。在scanf语句中进行了多次尝试：



这本是一个小小的错误，起因仅仅是我在学习输入语句时不够仔细而忘记正确的输入方式。但就是这个错误使得程序根本无法编译运行。后来在同学的提醒下改正。

1. 同样在第一题中，输入语句中scanf(“%d%d”,&a,&b);中a与b需要加入取地址符&数据方可正确输入。以下是查阅资料得知的取地址符作用：

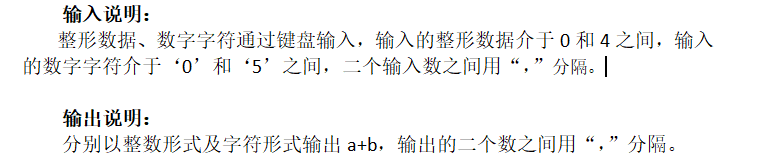
scanf(“格式控制字符串”, 地址表列);

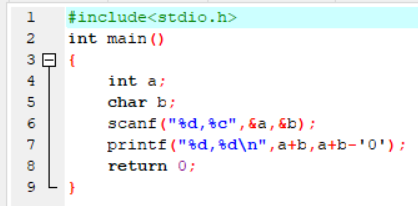
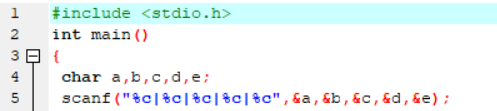
其中，格式控制字符串的作用与printf函数相同，但不能显示非格式字符串，也就是不能显示提示字符串。地址表列中给出各变量的地址。地址是由地址运算符“&”后跟变量名组成的。

例如：&a、&b分别表示变量a和变量b的地址。

这个地址就是编译系统在内存中给a、b变量分配的地址。在C语言中，使用了地址这个概念，这是与其它语言不同的。 应该把变量的值和变量的地址这两个不同的概念区别开来。变量的地址是C编译系统分配的，用户不必关心具体的地址是多少。

输入语句作为基本语句之一，其格式有诸多要求。对于C语言初学者而言所编写程序无法正常运行的罪魁祸首有一半是格式问题。诸如“；”、“&”等，虽简单却必不可少。

1. 在第四题与最后一题中均用到了%c格式将字符转化为其对应的ASCII编码。

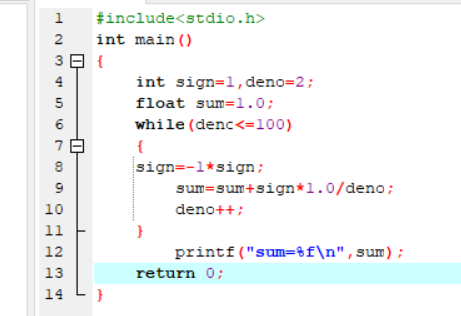


%c可以把输入的数字按照ASCII码相应转换为对应的字符。在进行字母的大小写转换中，我们尚未学到相关语句可以直接将小写大写相互转换，于是只能另想通路。因此容易联想到将字母用数字表示的ASCII编码。大小写字母所对应的ASCII编码差值恒为32，借助这一点便可以轻松解决该问题。而在最后一题中，需同时计算整数形式与字符形式的a+b。于是便需定义a为整数，b为字符形式。若输入3,5则输出结果中a+b项结果为3+53=58，a+b-’0’则会使输出强制转换为字符形式，结果为8。此处既可以使用%c来输出，亦可在输出表达式中加入一个-’0’来进行强制转换（相似用法还有在乘法除法中使用1.0之类将整数转化为浮点数）。在C语言中，用ASCII编码可以方便的处理字母与数字及其运算关系。

1. 在某一次的练习题中，已知m是一个三位整数，从左到右用a,b,c表示各位数字，现要求求出bac的值并计算m最后一个字节。

本题需要用到取余。在给出一个十进制整数时，要取其各位只需要c=m%10即为m除以10取余，得数即为各位。B=m/10%10所求即为十位，以此类推，利用取余即可求出各位数字。而计算最后一个字节需要用到进制的知识，即一个字节有八位，计算最后一个字节即计算最后八位，此时需注意应使用十六进制表示m。而一个十六进制数占四位，取最后一个字节即为取最后两个十六进制数。即bac=b\*100+a\*10+c n=m/16%16\*16+m%16,最后printf(“%X”,n)即可。须注意%x输出十六进制中若有字母则字母大小写取决于x大小写。C语言实际操作中进制的转换极其重要。不同进制有不同的作用场合，而当人工计算进制转换不够方便时，也可以将编译软件作为进制转换器。

1. 在例题2.4中，计算1-1/2+1/3-1/4+1/5-.........+1/99+1/100中



使用了while循环语句。需要注意的是while语句条件后不应出现“；”，同时while之下的复合语句需要使用一对{}包含在内。（deno++含义为先使用后加一）。

7.关于定义常量：在#include<stdio.h>预处理处加一句#define PI 3.1415926（中间使用空格隔断）。

选择结构程序知识点：

1. if语句用于实现两个分支选择结构，switch语句用于实现多个分支选择结构。
2. 关系运算符的优先次序为：算术运算符>关系运算符>赋值运算符。其中，关系运算符包括<,<=,>,>=,==（判断是否等于，一个=为赋值）,!=（不等于），逻辑运算符包括！非（a非零则!a=0），&&与（ab均非零，(a && b)值为一），||或（a与b其一是非零就是真）

循环结构知识点：

1. 循环体若有一个以上的语句，应该使用{}将其包含，作为复合语句出现。
2. 循环体中应具备使循环趋向于结束的语句，例如i++等。
3. do-while型会先无条件执行循环体，之后再判断条件是否成立。while型则会先判断条件是否成立再执行
4. for语句中表达式三也应包含使循环趋向于结束的语句，甚至可将循环体放到表达式3中。
5. 三种循环均可以使用break跳出循环，用continue结束本次循环。

以上，即上机实机操作得出的注意事项与知识点。

2020.10.16