



新编高等院校计算机科学与技术规划教材



程序设计

(C语言)实验指导

周春燕 张艳梅 编著

CHENGXU SHEJI
C YUYAN SHIYAN ZHIDAO



北京邮电大学出版社
www.buptpress.com

程序设计(C语言)实验指导

周春燕 张艳梅 编著



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书面向使用 C 语言进行程序设计的初学者,全书共分为 3 章。第 1 章是课堂同步实验指导,包括 12 个实验,共计 194 道题,以供在学习 C 程序设计语言具体知识点的时候同步采用;第 2 章是软件工程初步,简要介绍软件的概念和特点、软件开发过程以及结构化软件开发方法,并给出一个具体案例,使学生能在软件工程思想指导下来进行后续的课程设计;第 3 章是课程设计,给出了课程设计的 3 个综合型题目、课程设计实验步骤以及各类文档的模板,锻炼学生综合运用所学程序设计知识解决较复杂问题的能力。

本书可作为高等院校计算机专业或相关专业程序设计的实验指导书,也可作为学习参考书。

图书在版编目(CIP)数据

程序设计(C语言)实验指导/周春燕,张艳梅编著.--北京:北京邮电大学出版社,2011.9

ISBN 978-7-5635-2731-1

I. ①程… II. ①周…②张… III. ①C 语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 177661 号

书 名: 程序设计(C语言)实验指导

著作责任者: 周春燕 张艳梅 编著

责任编辑: 刘颖

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京联兴华印刷厂

开 本: 787 mm×1 092 mm 1/16

印 张: 16.75

字 数: 414 千字

印 数: 1—3 000 册

版 次: 2011 年 9 月第 1 版 2011 年 9 月第 1 次印刷

ISBN 978-7-5635-2731-1

定 价: 32.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

1. 本书面向的读者

国内很多高等院校都将 C 语言作为程序设计学习的入门语言。C 语言兼具高级语言和低级语言的特点,在嵌入式领域和系统级编程中有着广泛的应用。本书面向使用 C 语言进行程序设计的初学者,旨在提高大家的程序设计能力、问题分析和解决能力。

2. 本书编写背景

程序设计能力可以分成 5 个层次,从低到高依次是语言基础级(相当于认字、造句)、基本运用级(相当于写作文)、工具程序级(相当于写论文)、软件工程级(相当于写专著)以及大型系统级(相当于写系列丛书)。

通过 C 程序设计语言各语法点的学习,学生仅能达到语言基础级水平,而高年级的软件工程课程目标是让学生达到软件工程级水平,那么中间的基本运用级和工具程序级水平在何时、如何培养呢?基于多年的教学经验,笔者认为一个比较可行的方式是:在学完 C 语言各知识点之后,可以紧接着通过程序设计的课程设计来让学生达到基本运用级甚至是工具程序级的水平,培养学生综合应用所学知识分析问题、解决问题的能力。

因此,对于程序设计入门课程,除了需要安排传统的面向知识点的课堂同步实验之外,还应该有综合型课程设计。本着上述教学理念,本教材包含了课堂同步实验和课程设计两部分实验内容。

3. 本书编写思路

本书主体分成 3 章。

(1) 课堂同步实验

面向单独知识点的训练,如应用数组解题,这部分实验题目在介绍 C 程序设计语言具体知识点的时候同步采用。

本部分共设计了 12 个实验,每个实验都包括实验目的、实验覆盖的知识要点与难点、易出错点、实验题目难度划分以及实验中涉及的算法罗列,共计 194 道题。题目的设计思路是关注学生程序设计能力的训练,而非语法点的简单练习,具有一定的综合性。为了提高题目描述的准确性,对每一道题,都明确了输入输出要求,给出了测试用例。同时,为方便学生和老师选择题目,实验题目按照难度被划分成容易题、普通题、难题以及更难题 4 类。

(2) 课程设计

在完成课堂同步实验的练习后,就可以进入课程设计环节。课程设计的题目要求有一定的规模和难度,兼具一定的趣味性。本书 3 个题目(模拟单部电梯控制系统、模拟银行营业厅排队系统、模拟火车调度系统)源于现实生活,以小组为单位任选一个题目进行求解。通过对复杂问题进行分析和方案求解,旨在通过实践掌握基本的问题求解方法,即:将实际



问题抽象为模型,通过模型分析找出求解问题的基本思路,然后按照求解思路,设计程序并解决实际问题。通过课程设计,也希望让读者能充分地体会“理论、抽象、设计”3个学科形态的有机结合,培养抽象和建模的能力。

在内容安排上,本部分包括实验题目、实验过程组织、实验验收要求及文档模板的完整详细描述。这些内容已经在过去5年的教学中进行了充分的实践,具有很强的可操作性。

(3) 软件工程初步

大部分高校在低年级都没有设置课程来普及软件工程基本思想,使得学生是直到上高年级的软件工程课时,才知道原来软件! = 程序,才知道如何规范地开发软件。而作者认为,软件工程思想是随着不断地体验而逐步建立起来的,而非通过一个学期授课就能建立的。而且,学会和他人协作是当今学生必须具备的素质,在低年级的其他一些课程中也存在需要多人合作来做一个题目的情况。因此,在课程设计中,增加了软件工程初步这一部分内容,面向低年级学生,简要介绍软件工程的基本思想,软件生命周期模型以及结构化软件开发方法(重点是结构化设计方法),让学生在软件工程思想指导下以小组为单位开展课程设计。为了避免这部分内容过于理论化、降低学习难度,本书结合交通灯控制系统这个案例,给出了该系统的需求分析、概要设计、详细设计以及部分源代码。

当然,本课程的特点决定了我们只是选取软件工程的一部分内容进行介绍,更系统地学习需要在软件工程课程中完成。

4. 本书内容安排

本书的第1章是课堂同步实验,包括下述12个实验:实验1 熟悉集成开发环境;实验2 选择与循环结构;实验3 算法设计(一);实验4 算法设计(二);实验5 函数;实验6 递归函数设计;实验7 一维数组;实验8 二维数组和回溯法;实验9 字符串;实验10 指针与动态内存分配;实验11 结构与链表;实验12 文件与应用有限状态自动机解题。其中设计应用有限状态自动机解题的实验题目,主要是为课程设计服务的。在课程设计中,需要对电梯、火车和银行窗口的行为进行分析,需要有一种建模工具。我们建议采用有限状态自动机模型,让读者体会实际问题如何抽象到模型、体会建模对于问题求解的帮助。

本书的第2章是软件工程初步,包括软件工程概述(包括软件的概念和特点、软件危机、软件质量要素、软件工程三要素)、软件开发过程、软件需求分析、软件设计(包括软件设计目标、软件设计过程、软件模块化设计、软件结构图、详细设计、人机界面设计)、软件实现、软件测试,最后是十字路口交通灯控制系统案例分析。

本书的第3章是课程设计,包括模拟电梯控制系统程序设计实验要求详细描述、模拟火车调度系统程序设计的实验要求详细描述、模拟银行营业厅排队系统程序设计的实验要求详细描述、课程设计的程序总体结构说明、课程设计的实验步骤说明、课程设计的实验报告模板以及课程设计的测试用例参考。

此外,本书还包括3个附录。

附录A:C语言编程注意事项。总结罗列了19个常见注意事项,主要在课堂同步实验中参考。

附录B:多线程使用说明。在课程设计中,如果要使指令的输入和运行控制、输出能宏



观上同时进行,需要使用多线程技术。本附录中简要介绍了线程的概念以及实验中需要用的线程创建和同步技术。

附录 C: Visual C++ 6.0 操作手册。介绍了如何在 Visual C++ 6.0 环境下编写、编译、链接、调试和运行 C 语言程序。

5. 本书使用建议

本书第 1 章在 C 程序设计语言教学过程中采用,第 2 章和第 3 章在课程设计中采用。

课程设计建议先介绍实验要求(见第 3 章的 3.1~3.3 节),然后介绍软件工程初步(结合交通灯案例)以及线程的使用。为了控制复杂性,本书要求学生先后提交两个版本,第一个版本中完成核心功能,第二个版本中再实现输入输出的图形化。每个版本中还区分基本要求和提高要求。读者可以结合自己的实际情况进行选择。如果只完成版本 1,建议包括上课在内用时 9 周左右。如果还要完成版本 2,则再追加 3 周左右。

对于图形库函数,读者可以在互联网上免费获取。网络资源更新速度非常快,所以我们在此不作推荐。

小组规模建议控制在 3 人为宜,人数多则每人的任务量不饱满;人数太少,则不易暴露出多人协作相关的问题。

本书源自于北京邮电大学计算机学院多年的程序设计教学实践,作者具有丰富的教学经验。本书第 3 章由张艳梅编写,其他部分由周春燕编写。另外,特别感谢北京邮电大学计算机学院的石川老师、2008 级学生黑梦楠和张士菊同学对本书编写工作提供的帮助。

最后感谢读者选择使用本书。由于作者水平有限,书中难免有疏漏之处,欢迎各位同行和广大读者对本书提出修改意见和建议。

作者

目 录

第 1 章 课堂同步实验	1
实验 1 熟悉集成开发环境	1
实验 2 选择与循环结构	6
实验 3 算法设计(一)	15
实验 4 算法设计(二)	26
实验 5 函数	38
实验 6 递归函数设计	49
实验 7 一维数组	55
实验 8 二维数组和回溯法	68
实验 9 字符串	89
实验 10 指针与动态内存分配	106
实验 11 结构与链表	120
实验 12 文件与应用有限状态自动机解题	134
第 2 章 软件工程初步	149
2.1 软件工程概述	149
2.1.1 软件的定义和特点	149
2.1.2 软件危机与软件工程	150
2.1.3 软件质量要素	151
2.1.4 软件工程三要素	152
2.2 软件开发过程	153
2.3 软件需求分析	154
2.4 软件设计	155
2.4.1 软件设计目标	155
2.4.2 软件设计过程	156
2.4.3 软件模块化设计	156
2.4.4 软件结构图	161
2.4.5 详细设计	163
2.4.6 人机界面设计	163
2.5 软件实现	166
2.6 软件测试	166
2.6.1 软件测试基础	166



2.6.2 软件测试技术	167
2.6.3 软件测试过程	172
2.7 案例分析——十字路口交通灯控制系统	176
2.7.1 交通灯控制系统问题描述	176
2.7.2 需求分析	177
2.7.3 概要设计	180
2.7.4 源代码	190
第3章 课程设计	204
3.1 模拟电梯控制系统的实验要求	204
3.2 模拟火车调度系统的实验要求	208
3.3 模拟银行营业厅排队系统的实验要求	210
3.4 课程设计的程序总体结构	213
3.5 课程设计的实验步骤	214
3.6 课程设计的实验报告模板	216
3.7 课程设计的测试用例参考	221
附录A C语言编程注意事项	225
A.1 不定长字符序列的读取和处理	225
A.2 应该用 sizeof() 获取数据类型或变量占用的字节数	226
A.3 应避免使用依赖编译器求值顺序的语句	226
A.4 避免用“==”判断浮点数是否相等	227
A.5 要防范数值溢出	227
A.6 printf() 和 scanf() 中格式控制字符串与参数类型要匹配	228
A.7 执行 scanf() 时输入数据的方式与格式控制串要一致	229
A.8 使用 %c 格式符易出现的问题	230
A.9 数组下标不能越界	232
A.10 不可以利用数组名对数组内容进行复制和比较	232
A.11 字符串要以字符 '\0' 结尾	233
A.12 注意野指针	233
A.13 防止内存泄露	234
A.14 防止返回指向非静态局部变量的指针	234
A.15 不要混淆字符数组与字符指针变量	235
A.16 给头文件加宏定义锁	236
A.17 文件打开后要主动关闭	236
A.18 fopen() 中文件路径错误	237
A.19 文件操作易出现的其他问题	237



附录 B 多线程使用说明	238
B.1 进程和线程	238
B.2 创建线程	239
B.3 线程同步	242
B.4 利用互斥对象实现线程同步	245
附录 C Visual C++ 6.0 操作手册	249
C.1 新建 C 语言工程	249
C.2 编译源程序	250
C.3 运行程序	252
C.4 调试程序	252
C.5 创建、编译和运行工程	254
参考文献	257

第 1 章 课堂同步实验

实验 1 熟悉集成开发环境

一、实验目的

1. 掌握在一种集成开发环境下(如 Visual C++6.0、Dev-C++等)新建一个 C 程序,并会编辑、编译、链接、运行程序,查看程序运行结果。
2. 初步了解、掌握单步调试技术。
3. 通过简单编程掌握整型(int)、字符型(char)、浮点型(float、double)变量的定义和使用,常量的定义和使用,简单算术表达式的书写,输入 printf 输出 scanf 语句的基本使用。

二、知识要点与难点

1. 变量的含义和定义,变量的输入输出。(Problem 1、Problem 2)
2. 基本类型变量的定义和使用。(Problem 5)
 - char、int、float、double 这 4 种为常用类型。
 - short、long、long long 这 4 种为不常用类型。
3. 常量的含义和定义。需掌握两种定义方法:例如 #define N 10 与 const int N=10; 并理解这两种方法的不同之处。(Problem 4)
4. 输入输出语句 printf、scanf 的使用。掌握对各种基本数据类型数据的输入与输出。掌握浮点数保留 n 位小数的输出方法。(Problem 4)
5. 简单的算术运算与算术表达式的书写。(Problem 3)

三、易出错点

1. 书写标识符时,忽略了大小写字母的区别。注意:C 语言区分大小写字母,Age 和 age 是两个不同的标识符。
2. 变量未定义就使用。所有的变量必须先定义后使用,否则编译不通过。
3. 使用未初始化和未赋值的变量。注意:非全局和静态变量在定义时系统不会自动进行初始化,其初始值是一个与该程序运行环境有关的随机数。对这样的变量,使用前不显示初始化或者赋值是错误的。
4. 语句末尾忘记加分号。分号是 C 语句中不可缺少的一部分,语句末尾必须有分号。本次实验中出现的主要是输入输出语句、赋值语句,以后会学习到更多类型的语句。
5. 程序运行时,结果显示窗口一闪而过,可能是忘记在程序结束处加上 system("pause"); 语句。



6. scanf 读入变量时,容易忘记 & 符号,这样会导致内存泄露以致“发送错误报告”。
7. 运行 scanf() 时输入数据的方式与格式控制字符串要求不符。详见附录 A 第 A.7 条。
8. printf() 和 scanf() 中格式控制字符串与参数类型不匹配。详见附录 A 第 A.6 条。
9. 使用 printf 打印浮点数时,输出小数点后 x 位的方式写错。例如, m 为 float 类型的变量,正确保留 3 位小数的方法: printf("%.3f\n", m); 注意 ‘%’、‘.’、‘3’、‘f’ 这几个字符的顺序。

四、实验题目难度划分

本实验共 5 题。

- (1) 简单题: Problem 1、Problem 2;
- (2) 中等难度题: Problem 3、Problem 4;
- (3) 难题: Problem 5。

五、实验中涉及的算法

本次实验基本没有涉及算法的内容。

Problem 1: 求两数的和与差

问题描述: 输入两个整数,求这两个整数的和与差并输出。和为两数相加,差为第 1 个输入的数减去第 2 个输入的数。请使用整型(int 型)变量存储这两个整数。

输入与输出要求: 输入的两个整数之间以空格分隔,并且这两个整数以及它们的和、差均不会超过 int 变量的存储范围。程序运行效果中的 ‘↵’ 代表回车。输出这两个整数的和与差,假设输入了 1 与 2 这两个整数,那么输出是 “The sum is 3 and the difference is -1.”, 这句话要占一行。

程序运行效果:

Sample 1:

```
Please input two integers: 1 2 ↵  
The sum is 3 and the difference is -1.
```

Sample 2:

```
Please input two integers: 100 -45 ↵  
The sum is 55 and the difference is 145.
```

Problem 2: 基本类型数据的输入和输出

问题描述: 编写一程序,按照顺序分别输入某学生的性别、年龄和身高(以米为单位),然后将三者按照规定的格式输出。性别要用一个字符型(char 型)变量存储,年龄要用一个整型(int 型)变量存储,身高要用一个单精度浮点型(float 型)变量存储。输入时三者各占一行。

输入与输出要求: 首先输入的是一个字符 ‘M’ 或 ‘F’, 代表性别; 然后输入的是一个整数, 代表年龄; 最后输入的是一个浮点数, 代表身高。要输出的语句形如: “The sex is M, the age is 18, and the height is 1.660000.”, 这句话要占一行。

**程序运行效果:****Sample 1:**

Please input sex:M ✓

Please input age:18 ✓

Please input height:1.66 ✓

The sex is M,the age is 18,and the height is 1.660000.

Sample 2:

Please input sex:F ✓

Please input age:19 ✓

Please input height:1.71 ✓

The sex is F,the age is 19,and the height is 1.710000.

Problem 3: 四则运算

问题描述:已知四则算式 $(a+b) * (a-b) + c/d$,要求从键盘输入4个浮点型变量,按照输入顺序分别存入4个双精度浮点型(double型)变量中,假设这4个变量名即为a、b、c、d。然后将a、b、c、d代入四则算式中,将计算结果保存到另外一个双精度浮点型(double型)变量中。最后将该结果变量的值输出。

提示:双精度浮点型(double型)变量的定义方法为:double a;

双精度浮点型变量的输入方法为:scanf("%lf",&a);

双精度浮点型变量的输出方法为:printf("%lf",a)。

输入与输出要求:输入4个浮点数(以空格分隔),依次为四则算式中的a、b、c、d。这里d的值不会为0。输出四则算式的结果,假如输入的4个浮点数为1.5、2.0、10.00、5.0,那么要输出语句“The result is 0.250000.”,这句话要占一行。

程序运行效果:**Sample 1:**

Please input four float numbers:1.5 2.0 10.00 5.0 ✓

The result is 0.250000.

Sample 2:

Please input four float numbers:1.1 1.2 1.3 1.4 ✓

The result is 0.698571.

Problem 4: 求圆的周长与面积

问题描述:已知圆的半径(浮点数),求该圆的周长和面积并输出。圆的半径、周长、面积均要用双精度浮点型(double型)变量来存储。在程序中,还要定义一个名为PI的符号常量,值为3.14159265,即 π 值。圆的周长公式为 $2 * \pi * R$,圆的面积公式为 $\pi * R * R$,其中R为圆的半径。

输入与输出要求:输入一个正的浮点数,即圆的半径。输出圆的周长与面积。假如输入的半径为5.5,那么要输出语句“The perimeter is 34.5575,the area is 95.0332.”,这句话要



占一行。注意,结果保留4位小数。双精度浮点数打印4位小数的方法为:printf("%.4lf", area);“%.4lf”中“.4”即表示输出小数点后4位。

程序运行效果:

Sample 1:

Please input a float number:5.5 ✓

The perimeter is 34.5575,the area is 95.0332.

Sample 2:

Please input a float number:100.78 ✓

The perimeter is 633.2194,the area is 31907.9263

Problem 5:数据类型

问题描述:C语言为程序员提供了丰富的数据类型,常用到的数据类型有字符型(char 型)、短整型(short 型)、整型(int 型)、长整型(long 型)、拓展长整型(long long 型)、单精度浮点型(float 型)、双精度浮点型(double 型)。要求在程序中分别定义上述所有类型的变量,变量名按照上述顺序依次为a、b、c、d、e、f、g。按照上述顺序读入这些不同类型的数据,并按照原顺序输出。最后再输出这些变量在所用的操作系统中所占的字节数,这里要用到计算变量占用内存字节数的函数sizeof()。

提示:短整型变量的定义:short b;

短整型变量的输入:scanf("%hd",&b);

短整型变量的输出:printf("%hd",b);

长整型变量的定义:long d;

长整型变量的输入:scanf("%ld",&d);

长整型变量的输出:printf("%ld",d);

long long 型变量的定义:long long e;

long long 型变量的输入:scanf("%lld",&e);

long long 型变量的输出:printf("%lld",e)。

sizeof()的用法:

假如想知道在操作系统中,一个int型变量所占的内存空间为几个字节,则用以下代码:

int a=0;printf("%d\n",sizeof(a));或者 printf("%d\n",sizeof(int))。

注意:在某些非ANSI C标准的编译器下,很可能不支持long long数据类型,如VC 6.0,但VS 2003以上版本均支持。

输入与输出要求:输入包括7个部分,每个部分对应一种类型数据的输入。输出包括7个部分,每个部分占一行,对应一种类型的数据及其在内存占用的字节数。输出如“The 'int' variable is 2000000000,it takes 4 byte.”。

说明:受限于内存字节数,任何数据类型只能表示一定范围内的数,当试图往变量中存储在范围之外的数,数据将无法正确存储。这一点在第2个样例中可以明显看到。在该例中,第2~6个输入的数超出了数据类型的表示范围,因此无法正确存储;第7个数和输入有细微差别是因为浮点类型的数在内存中是近似存储的,目前大家先记住这个结论,原因后面将会介绍。



程序运行效果:

Sample 1:

Please input seven parts:

X ✓

1234 ✓

2000000000 ✓

2000000000 ✓

5000000000000000 ✓

1.2345 ✓

9.87654321 ✓

The result is:

The 'char' variable is X, it takes 1 byte.

The 'short' variable is 1234, it takes 2 bytes.

The 'int' variable is 2000000000, it takes 4 bytes.

The 'long' variable is 2000000000, it takes 4 bytes.

The 'long long' variable is 5000000000000000, it takes 8 bytes.

The 'float' variable is 1.234500, it takes 4 bytes.

The 'double' variable is 9.876543, it takes 8 bytes.

Sample 2:

Please input seven parts:

X ✓

2000000000 ✓

5000000000000000 ✓

5000000000000000 ✓

50000000000000005000000000000000 ✓

123456789123456789 ✓

123456789123456789 ✓

The result is:

The 'char' variable is X, it takes 1 byte.

The 'short' variable is -27648, it takes 2 bytes.

The 'int' variable is 1382236160, it takes 4 bytes.

The 'long' variable is 1382236160, it takes 4 bytes.

The 'long long' variable is 2538972135152631808, it takes 8 bytes.

The 'float' variable is 123456790519087100.000000, it takes 4 bytes.

The 'double' variable is 123456789123456780.000000, it takes 8 bytes.



实验2 选择与循环结构

一、实验目的

1. 掌握整型数据、字符型数据的输入、输出处理,算术表达式和逻辑表达式的正确书写。
2. 练习使用 C 语言中的选择结构和循环结构解决简单问题。

二、知识要点与难点

1. 字符型数据的读取有两种方式:scanf("%c",&m);,或者 m=getchar();。
2. 选择结构:if...else 语句的使用,在使用该语句时,要尽量加上大括号,哪怕大括号里只有一句代码;嵌套的 if...else 语句的使用(即 if 或 else 部分的语句又是 if...else 语句);switch 语句的使用。
3. 循环结构:while 语句的使用、for 语句的使用,要理解它们的本质是一样的;while 语句与 do...while 语句的区别;break 语句与 continue 语句对循环的控制。
4. 利用 printf 打印制表符 tab 的方法、深入了解转义字符'\t'的作用。(Problem 4)
5. 难点:循环语句的嵌套,将两个 for 循环嵌套使用,for 循环与 while 循环嵌套使用等;选择结构的嵌套使用。

三、易出错点

1. 在 if...else 选择结构的条件表达式中,很可能将“==”错写为“=”,使得表达式运算结果出错。在许多高级语言中,用“=”符号作为关系运算符“等于”。但在 C 语言中,“=”是赋值运算符,“==”是关系运算符。C 语言中赋值表达式的值为赋值号右边表达式的值,值非零表示真,否则表示假。例如 if(a=0)中的“a=0”值为假,if(a=20)中的“a=20”值为真。
2. 嵌套的 if...else 语句中,if 和 else 的配对不正确。
3. if...else 选择结构中,由于漏写大括号导致程序出现编译错误。例如:

```
if(a>b){  
    c = d;  
    f = e;  
}  
else{  
    c --;  
    d --;  
}
```

错写成:

```
if(a>b)  
    c = d;  
    f = e;  
else  
    c --;  
    d --;
```

在右侧代码中,因为漏写大括号,使得代码实际为:

```
if(a>b)  
    c = d;
```



```
f = e;
else
    c--;
d--;
```

因为 else 没有匹配的 if, 从而编译报错。

4. 循环语句的条件判断出现错误导致程序进入死循环; 或由于每次循环结束后循环变量不发生变化导致程序进入死循环。在循环条件的判断中, 要注意 $>$ 、 $>=$ 、 $<$ 、 $<=$ 等运算符号的正确使用, 否则可能出现多循环一次或者少循环一次的情况。

5. 循环结构的循环体漏写大括号导致程序出现逻辑错误。例如:

```
t = 100;
while(t > 0)
{
    c * = 10;
    t / = 10;
}
```

错写成:

```
t = 100;
while(t > 0)
    c * = 10;
    t / = 10;
```

在右侧代码中, 因为漏写大括号, 导致循环体中只有语句 $c * = 10;$, 从而使程序进入死循环。

6. while 语句与 do...while 语句的混淆, 要理解这两类语句哪个先判断再循环, 哪个先循环再判断。

7. switch 语句中的 case 语句后容易遗忘 break 语句。要清楚 switch 语句中使用和不使用 break 语句的区别。

四、实验题目难度划分

本实验共 11 题。

- (1) 简单题: Problem 1~Problem 4;
- (2) 中等难度题: Problem 5~Problem 7;
- (3) 难题: Problem 8~Problem 10;
- (4) 更难题: Problem A。

五、实验中涉及的算法

1. 简单的选择结构。(Problem 1、Problem 2)
2. 字符的 ASCII 码值, 字符的大小比较。(Problem 2)
3. 简单的循环结构。(Problem 3)
4. 打印制表符 tab, 使输出更加工整。(Problem 4)
5. 稍复杂的选择结构, sqrt() 函数的使用, 表达式计算。(Problem 5)
6. 有关整数拆分的算法。(Problem 6~Problem 8)



7. 简单的循环嵌套。(Problem 9)
8. 稍复杂的模拟题目,switch 语句的使用。(Problem 10)
9. long long 型变量的应用,数学问题、简单循环的应用。(Problem A)

Problem 1:判断两数的大小

问题描述:已知有两个整数,使用 if...else 选择结构将它们中的较大数选择出来,存到 max 变量中;将较小数选择出来,存到 min 变量中,并将选择结果输出。

输入与输出要求:输入包括两个整数,占一行,用空格分隔。输入的整数可以相同,此时输出“The two numbers are equal.”,占一行。当输入不相同,输出如“The larger number is 100,the smaller number is 25.”,占一行。

程序运行效果:

Sample 1:

Please input two integers:100 25 ✓

The larger number is 100,the smaller number is 25.

Sample 2:

Please input two integers:100 100 ✓

The two numbers are equal.

Problem 2:判断字符相邻

问题描述:已知任意 3 个字母,判断它们是否相邻。所谓 3 个字母相邻指的是第 1 个字母与第 2 个字母相邻,第 2 个字母与第 3 个字母相邻,如“abc”,“bcd”,“XYZ”等。提示:可以利用字符相减操作判断两字母是否相邻。两个字符之差如果是-1,则相邻。例如‘a’-‘b’=-1,则‘a’与‘b’相邻。

注意:如果 3 个字符中有非字母的输入,判断结果应为不相邻,例如输入“123”,“12a”即为不相邻。

输入与输出要求:输入包括 3 个字符,以回车结束,占一行,这 3 个字符之间没有空格。当输入的 3 个字符相邻时,输出“The letters are adjacent with each other.”;否则输出“The letters are not adjacent with each other.”。

程序运行效果:

Sample 1:

Please input three char:abc ✓

The letters are adjacent with each other.

Sample 2:

Please input three char:AbC ✓

The letters are not adjacent with each other.