



高等学校计算机教材

C/C++程序设计 实验与习题指导

孙淑霞 李思明 刘焕君 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

高等学校计算机教材

C/C++程序设计 实验与习题指导

孙淑霞 李思明 刘焕君 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是 C/C++ 程序设计课程的配套教材。全书分为 5 部分。第 1 部分为实验指导, 由 11 个实验组成; 第 2 部分为习题, 由 11 章组成; 第 3 部分介绍 Visual C++ 6.0 的实验环境; 第 4 部分是习题解答。这 4 部分的内容与《C/C++ 程序设计教程》一书的每一章内容相对应, 以便进行实验教学和课后练习; 对于每一个实验, 都给出了提示和帮助性的指导; 习题给出了答案和解答。第 5 部分给出了 3 套 C 语言笔试和上机考试的模拟试题及答案。附录中提供了常见的编译错误信息、连接和运行中的错误信息, 以及常用的 Turbo C 的库函数、热键和编辑键。

本书可以作为大专院校非计算机专业本科生、研究生的相关课程的实验教学用书, 也可作为计算机专业相关课程的实验教材, 还可作为 C/C++ 程序设计自学者的参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

C/C++ 程序设计实验与习题指导/孙淑霞, 李思明, 刘焕君编著. —北京: 电子工业出版社, 2005.1
高等学校计算机教材
ISBN 7-121-00677-4

I. C… II. ①孙…②李…③刘… III. C 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 133423 号

责任编辑: 王昌铭

印 刷: 北京兴华印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 16 字数: 416 千字

印 次: 2005 年 1 月第 1 次印刷

印 数: 4000 册 定价: 22.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。
联系电话: (010) 68279077。质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

出版说明

信息社会对每一个人，尤其是科技人员、教育工作者和广大学生提出了知识的更新和重构的问题；在知识增值的情况下，对如何获取知识和信息产生了强烈的要求。虽然我们可以从网络、从媒体、从书本上获得信息和知识，但在教育领域，教材和技术参考书仍然是重要的知识来源，它们是加工（处理）后的信息。

计算机科学技术的发展，引起了一场革命，形成了一种文化，创新了一种观念，促进了社会的发展。以微电子为依托，计算机与通信的联姻，带来了全球经济文化的大变革，信息全球化已经初见端倪。为了适应知识的更新和信息的获取，经过高等院校、企业事业单位和应用领域用户的认真讨论，我们规划了这套电子信息类教材和技术参考书。

在现代社会中，电子信息类教材和技术参考书包含了计算机专业类和非计算机专业类、网络与通信专业类、电子技术类，读者层次也覆盖了各个应用领域，这些教材和技术书籍将对我们的信息技术的普及起到极大的推动和促进作用。

在现代教育体系中，教材的作用是不言而喻的，但教材的内容则各有千秋。由于学科发展、专业结构、培养对象的不同，全国统编教材已经不完全适应各个学校。多种风格各异、独具特色的教材如雨后春笋般层出不穷，并深受读者欢迎。一本教材的内容不可能包揽一切，也不可能适应各层次的读者对象。但是，教材一定要有自己的特色，这些特色是通过知识点及其之间的关联性体现出来的。这样，教材的选用者就可以了解该教材的适用范围和对象，就能够很好地规划自己的教学。

信息技术类参考书是对专业教材内容的一个重要补充，也是对专业技术实践环节（实验、实训、实作）的一个实施，这种参考书体系将紧跟国际新技术发展，把最实用的技术及时介绍给读者。

本系列教材及技术参考书的编写采用编委会形式（选题内容由需求引导，编委会讨论，专家、行家撰写，编委会审阅，最后定稿），以保证书的编写质量。

当我们跨进 21 世纪大门的时候，希望广大的读者能与我们携手并进，共同关心我们的教材和技术参考书的出版，将需求、建议和意见及时反馈给我们，与我们一起开辟一片新的天地。

编委会

高等学校电子信息类教材编委会名单

主 任：刘乃琦

委 员：丁志强 马 争 王世普

王昌铭 邓昌延 付 彦

刘乃琦 李志蜀 李玉柏

余 江 吴 跃 吴宗粹

张宽海 张 扬 张翠芳

钱寿宇 梁成华 黄书万

顾 问：刘锦德 俸远祯

前 言

学习语言如同学习写作，只有勤于思考，多练习才能收到好的效果。在多年的教学实践过程中，我们发现有不少学生在开始学习 C/C++ 语言程序设计时，感到入门难，掌握起来更难，对很多问题知其然不知其所以然，是是而非。对于初学 C/C++ 语言程序设计的读者，一定要勤于思考，多读程序、多编写程序、多上机调试程序，只有这样才能尽快地掌握和运用 C/C++ 语言去解决实际问题。基于此，我们根据多年的教学经验和体会，编写了这本教材。

本书作为计算机基础系列实验教材，与《C/C++ 程序设计教程》一书配套使用。全书分为 5 部分，其内容包括：实验指导、习题和习题解答，以及与其相关的 Visual C++ 6.0 的实验环境介绍，同时还给出了 3 套 C/C++ 语言笔试和上机考试的模拟试题及答案。书中安排的 11 个实验项目和 11 章习题与《C/C++ 程序设计教程》的每一章内容相对应，以便进行实验教学和课后练习。为了方便学习，对于每一个实验，都给出了提示和帮助性的指导，习题给出了参考答案和解答。

本书实验和习题的章节与《C/C++ 程序设计教程》一书相同，其主要内容包括：C 语言简单程序的编写和调试，C 语言程序设计基础（其中包括基本数据类型和基本输入/输出函数、运算符与表达式），控制结构，数组，指针，函数，文件，编译预处理，结构体与共用体，图形程序设计，C++ 程序设计基础。

本书在内容的安排上注重理论和实践的结合，使学生在学完每一章后，都可以通过完成相应的习题和上机实践巩固所学的理论知识。考虑到初学者的困难，实验指导根据题目的难易程度给予不同程度的提示和帮助。习题部分的分析也根据习题的难易程度给予详略程度不同的解释，使学生能够真正掌握所学的知识点。

为了帮助读者解决上机调试程序中的错误，附录中提供了常见的编译错误信息、连接和运行中的错误信息。

本书第 1 部分和第 3 部分由孙淑霞编写，第 2 部分和第 4 部分由李思明和刘焕君编写。丁照宇、彭舰也参加了本书的策划、出题、解题和实验等工作，孙淑霞统编了全书。由于水平有限，书中难免有错误之处，请读者批评指正。

最后要感谢高等学校电子信息类教材编委会在本书的出版过程中给予的指导和帮助，特别要感谢王昌铭老师在该书的出版过程中所做的大量工作，同时要感谢电子工业出版社给予的大力支持。

对于使用本教材的学校，如果需要书中的有关资料，可以直接与作者联系（作者邮件地址：ssx@cdut.edu.cn）。

编著者

2004 年 10 月

目 录

第 1 部分 实验指导	(1)
实验 1 C 语言简单程序的编写和调试	(1)
实验 2 C 语言程序设计基础	(3)
实验 3 控制结构	(8)
实验 4 数组	(13)
实验 5 指针	(19)
实验 6 函数	(24)
实验 7 编译预处理	(30)
实验 8 文件	(32)
实验 9 结构体与共用体	(35)
实验 10 图形程序设计	(40)
实验 11 C++程序设计基础	(43)
第 2 部分 习题	(49)
第 1 章 C 语言简单程序的编写和调试	(49)
第 2 章 C 语言程序设计基础	(49)
第 3 章 控制结构	(55)
第 4 章 数组	(62)
第 5 章 指针	(70)
第 6 章 函数	(79)
第 7 章 编译预处理与变量存储类型	(100)
第 8 章 文件	(105)
第 9 章 结构体与共用体	(111)
第 10 章 图形程序设计	(117)
第 11 章 C++程序设计基础	(117)
第 3 部分 Visual C++ 6.0 实验环境简介	(122)
3-1 Visual C++ 6.0 界面	(122)
3-2 Visual C++ 6.0 环境下文件的调试与运行	(123)
3-3 开发环境的菜单及工具栏	(130)
3-4 工程与工程工作区	(141)
第 4 部分 习题解答	(144)
第 1 章 C 语言简单程序的编写和调试	(144)
第 2 章 C 语言程序设计基础	(144)
第 3 章 控制结构	(152)

第4章 数组	(156)
第5章 指针	(160)
第6章 函数	(168)
第7章 编译预处理与变量存储类型	(177)
第8章 文件	(180)
第9章 结构体与共用体	(183)
第10章 图形程序设计	(187)
第11章 C++程序设计基础	(189)
第5部分 模拟试题及答案	(191)
笔试模拟试题	(191)
试题一	(191)
试题二	(199)
试题三	(208)
笔试模拟试题参考答案	(217)
试题一	(217)
试题二	(217)
试题三	(218)
上机模拟试题	(218)
试题一	(218)
试题二	(220)
试题三	(220)
上机模拟试题参考答案	(221)
试题一	(221)
试题二	(223)
试题三	(224)
附录	(226)
附录A 常用的 Turbo C 库函数	(226)
附录B Turbo C 的常用热键和编辑键	(231)
附录C Turbo C 编译错误信息	(234)
致命错误	(234)
一般错误	(234)
警告信息	(243)
连接和运行中的出错信息	(246)
参考文献	(247)

第1部分 实验指导

实验1 C语言简单程序的编写和调试

一、实验目的

熟悉 Turbo C 集成环境，掌握在该环境下进行程序调试的一般方法。

二、实验要求

1. 熟悉 Turbo C 集成环境的使用，掌握在 Turbo C 集成环境下输入、编译、调试和运行 C 程序的基本过程和方法。
2. 通过编写简单程序，掌握 C 程序的基本组成和结构，以及用 C 程序解决实际问题的步骤。
3. 编写实验报告（一）。

三、实验内容

- 1.1 在 Turbo C 集成环境中练习程序的输入、编译、连接和运行的基本方法。

〔指导〕

- (1) 启动 Turbo C，进入 Turbo C 集成环境。
- (2) 输入程序 S1-1.C。

源程序 S1-1.C

```
#include <stdio.h>
main()
{
    printf("Hello, C! \n");
}
```

- (3) 执行“Compile”菜单中的“Compile to OBJ”命令，编译程序 S1-1.C，产生目标程序 S1-1.OBJ。
- (4) 执行“Compile”菜单中的“Link EXE file”命令，连接目标程序 S1-1.OBJ 和函数库 stdio.h，生成可执行程序 S1-1.EXE。
- (5) 执行“Run”菜单中的“Run”命令，按组合键 Alt-F5，观察屏幕上输出的程序运行结果：

Hello, C!

1.2 练习程序的编辑（包括块定义、块移动、块复制等块操作）和“另存为”命令。

【指导】

(1) 按照 S1-2.C 的内容编辑程序 S1-1.C。

源程序 S1-2.C

```
#include <stdio.h>
main()
{ int a, b, sum;          /* 增加变量的定义 */
  a=40; b=50;             /* 增加变量的赋值 */
  sum=a+b;                /* 增加变量的求和 */
  printf("Hello, C! \n");
  printf("Sum is %d \n", sum); /* 用块操作增加该行 */
}
```

在程序的编辑过程中，熟练使用块操作可以提高编辑程序的速度。请按如下方法增加语句行：printf("Sum is %d \n", sum);

① 定义块首。将光标定位在 printf("Hello, C! \n"); 语句行的开始，按组合键 Ctrl-K B（按住 Ctrl 键，然后依次按下字母键 K 和 B）；

② 定义块尾。将光标定位在该行的末尾，按组合键 Ctrl-K K，这时该行被着色；

③ 复制块。将光标定位在下一行的行首，按组合键 Ctrl-K C，将定义的块复制到该行；

④ 取消块定义。按组合键 Ctrl-K H；

⑤ 将 printf("Hello, C! \n"); 中的 Hello, C! 修改为：Sum is %d。

(2) 执行 File 菜单中的“Write to”命令，在“New Name”框中输入文件名 S1-2.C，按回车键，将当前文件保存为 S1-2.C。

1.3 练习程序的简单调试。

【指导】

对程序 S1-2.C 做如下操作：

(1) 去掉 printf("Hello, C! \n"); 中的分号“;”，重新进行编译，观察编译错误。

这时 Turbo C 集成环境的 Message 窗口中出现如下信息：

Compiling E:\C\S1-2.C:

Error E:\C\S1-2.C 7: Statement missing ; in function main

Warning E:\C\S1-2.C 8: 'sum' is assigned a value while is never used in function main

其中：

第 1 条信息表示当前正在编译的程序是 E: 盘文件夹 C 中的 S1-2.C；

第 2 条信息指出在 S1-2.C 的第 7 行有语法错误，错误是：在 main 函数的第 7 行少了分号“;”，这种错误通常是因为上一行的末尾少了分号“;”；

第 3 条信息指出在 S1-2.C 的第 8 行有警告错误，错误是：在 main 函数中，sum 被赋予了一个未使用过的值。这个错误是由于上一个错误造成的，如果在上一行的末尾加上分号，该错误就不会再出现。

(2) 把 main 改为 mian，重新进行编译和连接，观察编译和连接错误。

这时编译没有错误，连接时出现如下错误信息：

Linking E:\CS1-2.EXE:

Linking Error: Undefined symbol '_main' in module COS

其中:

第 1 条信息表示当前正在连接的程序是 E: 盘文件夹 C 中的 S1-2.EXE;

第 2 条信息指出连接错误是: 没有定义 main 函数, 造成该错误的原因是因为函数名 main 写错了。

提示:

(1) 在调试程序的过程中, 如果出现编译错误, 要由上至下一个一个地去修改, 每改一处, 就要重新编译一次, 不要想一次把所有错误都修改后再编译。因为, 有时一个错误会引起下面程序段中与之有关的行也出现错误, 改正了这一个错误, 其他错误也就随之消失了。

(2) 有些错误会出现在连接阶段, 例如, 在实验 1-3 中, 把 main 写成了 mian, 编译程序把 mian 当成是用户自定义函数进行编译, 没有语法错误, 也就没有报错。但由于 C 程序必须要有一个且只能有一个 main() 函数, 连接程序没有发现 main() 函数, 因此在连接阶段报错。

如果将 printf 误写为 print, 连接程序也会因为找不到相应的库, 在连接阶段给出连接错误的提示信息:

Undefined symbol '_print'

说明 print 是未定义符号。

(3) 当调试程序中出现了编译、连接或运行错误, 可以查看附录 C 中提供的常见错误信息。要注意培养自己独立分析问题和解决问题的能力, 积累查错的经验, 逐渐提高调试程序的能力; 千万不要被错误所吓倒, 相信自己一定会在调试程序的过程中成长起来。

实验 2 C 语言程序设计基础

一、实验目的

掌握 C 语言的运算符和表达式的正确使用及 C 语言的几种基本数据类型的定义和初始化。
掌握基本输入/输出函数的使用方法。

二、实验要求

1. 通过编程进一步理解和掌握运算符的确切含义和功能。
2. 理解和掌握运算符与运算对象的关系。例如单目运算符只对一个运算对象进行操作, 双目运算符需要两个运算对象。
3. 理解和掌握运算符的优先级和结合方向。
4. 通过编写程序, 掌握 C 语言的几种基本数据类型, 如整型 int、字符型 char、实型 float、双精度型 double, 以及由这些基本类型构成的常量和变量的使用方法。
5. 掌握基本输入/输出函数的使用方法。其中包括 printf() 函数、scanf() 函数、getchar() 函数和 putchar() 函数。

6. 编写实验报告 (二)。

三、实验内容

2.1 编程 (文件名为 S2-1.C)。定义 3 个 int 变量 x, y, z 和一个实型变量 $average$, 计算并输出 3 个整数的平均值 $average$, 即 $average=(x+y+z)/3$, 其中 x, y, z 的值可以通过变量的初始化赋值 $x=1, y=1, z=0$, 也可以通过赋值语句赋值, 观察运行结果。将 x, y, z 和 $average$ 改为实型变量, 再观察运行结果。要求理解整除和不同类型数据进行混合运算的转化规律, 并对结果进行合理的解释。

【指导】

(1) 当把 x, y, z 定义为整型变量时, 所做运算: $(x+y+z)/3$ 是整除运算。根据 C 语言的语法规则: 整除结果应舍去小数部分, 保留整数部分, 而不做四舍五入。因此, $average=(x+y+z)/3$ 是将取整的结果赋给 $average$ 。

(2) 当定义 x, y, z 为整型, $x=1, y=1, z=0$ 时, $average$ 的值就为 0。要想使 $average$ 的值为非 0, 应将 $x+y+z$ 的值或 3 转换为实型。可用如下方法之一进行实型转换:

① $average=(x+y+z)/3.0;$ (把 3 改为 3.0)

② $average=(float)(x+y+z)/3;$ (把 $x+y+z$ 的结果强制转换为实型)

(3) 如果把 x, y, z 和 $average$ 都定义为 float 型, 结果就不再为 0, 而是等于 0.67。

通过该程序的编写, 目的是进一步认识数据的类型, 以及不同类型数据的混合运算。

2.2 改写程序 S2-1.C (文件名为 S2-2.C), 通过键盘用 scanf() 函数输入 x, y, z 的值。要求输出结果的形式为 $average=\dots$ 。

【指导】

在 S2-1.C 中, 变量 x, y, z 的值是通过变量的初始化或赋值语句赋的值。用这种方法编的程序不灵活, 如果要改变 x, y, z 的值, 就要修改程序。如果用 scanf() 函数输入 x, y, z 的值, 就不需要对程序做任何修改, 只需在运行程序的过程中, 给 x, y, z 输入所需的值。

比较下面 3 个程序, 模仿比较好的一种方法编写 S2-2.C。

源程序 S2-2-1.C

```
#include <stdio.h>
main()
{ int x=1,y=1,sum;          /* 定义变量 */
  sum=x+y;                  /* 求和 */
  printf("sum=%d\n",sum);    /* 输出 */
}
```

该程序只能求 $x=1$ 与 $y=1$ 的和。如果要求其他数据的和, 则必须修改程序中的初始化, 然后重新编译、连接并运行。

源程序 S2-2-2.C

```
#include <stdio.h>
main()
{ int x,y,sum;              /* 定义变量 */
```

```

scanf("%d%d",&x,&y);    /* 输入 */
sum=x+y;                /* 求和 */
printf("sum=%d\n",sum); /* 输出 */
}

```

该程序可以求任意两个数的和，而不需要修改程序。当程序运行时，输入要求和的数，程序就可输出结果。

源程序 S2-2-3.C

```

#include <stdio.h>
main()
{ int x,y,sum;          /* 定义变量 */
  printf("\nEnter x y: "); /* 提示输入 */
  scanf("%d%d",&x,&y);    /* 输入 */
  sum=x+y;              /* 求和 */
  printf("sum=%d\n",sum); /* 输出 */
}

```

该程序与 S2-2-2.C 的区别是：在输入之前增加了屏幕提示，在运行程序时，可按屏幕提示进行数据的输入。

运行上面每一个程序，真实感受一下它们的不同。

通过该程序的编写，目的是从一开始就采用较好的编程风格。

提示：

通过 S2-2-1.C、S2-2-2.C、S2-2-3.C 程序的编写，应该学会用良好的编程风格编写程序。

(1) 编写程序时要考虑程序的通用性，需要变化的量尽量不要通过赋值的方式给定（例如，S2-2-1.C 中的 x 和 y ），而是通过输入的方式使变量得到当前所需的值（例如，S2-2-2.C 中对 x 和 y 的输入）。

(2) 从键盘输入数据时，最好先给出提示信息，提示要输入的数据（例如，S2-2-3.C 中在输入 x 和 y 前用 `printf()` 函数输出的屏幕提示信息）。

2.3 运行程序 S2-3.C，观察输出的结果，并对输出结果作出合理的解释。

源程序 S2-3.C

```

main()
{
  float a1,a2;
  double b1,b2;
  a1=3141.59; a2=0.000001;
  b1=3141.59; b2=0.000001;
  printf("%f, %lf\n", a1+a2, b1+b2);
}

```

【指导】

程序中的 $a1$ 和 $b1$ 、 $a2$ 和 $b2$ 的值分别相同，但程序中 $a1+a2$ 和 $b1+b2$ 的输出结果却不会相同。这是因为 `float` 型数据和 `double` 型数据的精度不同，实型数在计算机中是非精确表示

的, 用 float 型变量做两个位数很大的数的加法运算或除法运算时, 将得不到预期的结果; 而用 double 型变量进行运算则可以得到较准确的结果。

该程序的目的是进一步认识不同类型的数据, 其精度是不相同的, 编程时应该根据要求定义变量的类型。

2.4 下面程序 (S2-4.C) 的输出结果是什么? 对输出结果给予合理的解释。

源程序 S2-4.C

```
main()
{char ch;
 int k;
 ch='a'; k=10;
 printf("%d, %x, %o, %c", ch, ch, ch, ch, k);
 printf("k=%d\n", k);
}
```

【指导】

在 C 语言中, 字符数据既可以用字符形式输出, 也可以用整数形式输出。字符 a 的 ASCII 十进制代码为 97, 按十六进制形式输出为 61, 按八进制形式输出则为 141。本题中第 1 个 printf 函数调用语句中输出项 k 是多余的, 因此不予输出; 第 2 个 printf 函数调用语句中的格式说明中包含了两个连续的 % 字符, 根据 C 语言的规定, %% 不再作为格式描述字符使用, 而是处理成字符 “%” 的原样输出, 因此该语句中的输出项 k 任然没有对应的格式描述符, 也将不予输出。

该程序的目的是为了掌握变量按不同格式的输出生, 进一步认识格式描述符, printf 函数中参数的正确使用。

2.5 写出程序 (S2-5.C) 的运行结果, 并对输出结果给予合理的解释。

源程序 S2-5.C

```
#include <stdio.h>
main()
{float x;
 double y;
 x=213.82631;
 y=213.82631;
 printf("%-4.2f, %-6.2e\n", x, y);
}
```

【指导】

使用 f 格式描述符输出浮点数时的一般形式是 %m.nf 或 %-m.nf。其中 m 指定输出数据所占的总列数, n 指定小数点后的位数, “-” 是使输出数据左对齐; 当输出数据宽度大于 m 时, 数据的整数部分将按照实际位数输出。在本例中由于输出数据宽度大于 m, 因此 % 后面的 “-” 对输出格式无影响。使用 e 格式描述符输出浮点数时的一般形式是 %m.ne 或 %-m.ne。其中 m、n、“-” 的含义与 f 格式相同。在不同的计算机中, 对指数部分应占的宽度规定不同, 而数值部分均按标准化指数形式输出 (即小数点前必须有而且仅有一位非

零数字)。该题中 y 的实际宽度为 9，而格式说明中所给定的域宽为 6，所以只能按标准化指数形式输出 y 的整数部分，截去小数部分并四舍五入。

该程序的目的是为了进一步认识格式描述符。

2.6 下面程序 (S2-6.C) 的功能是从键盘上输入 $x=25$, $y=36.7$, $c=C$, 然后将输入的内容输出到屏幕上。调试程序 S2-6.C, 修改有错误的语句行, 并输出正确的结果。

源程序 S2-6.C

```
#include <stdio.h>
main()
{ int x;
  float y;
  char c;
  scanf("x=%d,y=%d,c=%c",x,y,c);
  printf("\nx=%d,y=%d,c=%c",x,y,c);
  getch();
}
```

【指导】

调试程序 S2-6.C 时要注意的问题有:

(1) 函数 `scanf()` 的输入表列要求是变量地址的表列, 否则在编译时会给出警告错误:

Possible use of xxx before definition

(2) 函数 `scanf()` 的格式描述符要与输入表列中变量的类型一致。如果类型不一致, 将得不到正确的输入值。如果 `float` 型变量用 `%d` 格式输入, 得到的是 0; `int` 型变量用 `%f` 格式输入, 运行时会给出错误信息:

scanf: floating point formats not linked

Abnormal program termination

(3) 如果函数 `scanf()` 的格式描述符中有按原样输入的字符, 输入数据时一定要按格式输入这些字符。

(4) 函数 `printf()` 的格式描述符要与输出表列中变量的类型一致, 否则得不到正确的输入结果。如果 `float` 型变量用 `%d` 格式输出, 得到的是 0; `int` 型变量用 `%f` 格式输出, 运行时会给出错误信息:

printf: floating point formats not linked

Abnormal program termination

2.7 编程 (S2-7.C)。执行下列语句后, a, b, c, d, e 的值分别等于多少? 为什么?

```
int a,b,c,w=1,x=2,y=3,z=4,d=5,e=6;
```

```
a=b=c=1;
```

```
++a++b&&++c;
```

```
(d=w>x)&&(e=y>z);
```

【指导】

在逻辑表达式的计算中要注意:

(1) 对于 `++a||++b&&++c` 运算, 若 `++a` 为真, 后面的运算就不再进行。

(2) 对于 `++a&&++b||++c` 运算, 若 `++a` 为真, 继续计算 `++b`; 若 `++b` 为真, 就不再计算 `++c`; 若 `++b` 为假, 则继续计算 `++c`。

(3) 逻辑运算或关系运算的结果是一个逻辑值, 即 0 或 1。

2.8 假设有 `unsigned int a=3,b=10;`

`char c=9, d=020;`

计算 `a<<2|b>>1` 和 `~c&d<<1` 的结果, 并编程 (S2-8.C) 验证。

【指导】

(1) 进行位运算时, 先将要运算的数据转换为二进制。

(2) 对于 `a<<2|b>>1`, 移位运算优先于按位或运算。

(3) 对于 `~c&d<<1`, 左移运算优先于按位与运算, 求反运算符优先级最高。

实验3 控制结构

一、实验目的

通过实验掌握结构化程序设计的基本思想和方法, 以及 C 语言的基本控制结构和控制转移语句。

二、实验要求

1. 掌握形成控制结构语句的使用方法, 熟练运用。

① 选择结构语句: `if` 语句, `switch` 语句。

② 循环结构语句: `for` 语句, `while` 语句, `do-while` 语句。

2. 掌握控制转移语句的正确使用方法, 并在编程中能灵活使用, 通过编程掌握在什么情况下使用下列语句。

① 限定转向语句: `break` 语句, `continue` 语句, `return` 语句。

② 无条件转向语句: `goto` 语句。

3. 编写实验报告 (三)。

三、实验内容

3.1 阅读下面程序 (S3-1.C), 说明该程序的功能, 写出执行过程及结果。

源程序 S3-1.C

```
#include "stdio.h"
main()
{
    int day;
    printf("请输入要查询的星期: \n");
```



```

scanf ("%d",&day);
if (day==1)
    printf("上午: 英语, 数学, 政治, 体育  下午: 法律\n");
else if (day==2)
    printf("上午: 物理, 计算机, 美术  下午: 音乐\n");
else if (day==3)
    printf("上午: 英语, 数学, 语文  下午: 锻炼\n");
else if (day==4)
    printf("上午: 计算机, 物理, 形势  下午: 班会\n");
else if (day==5)
    printf("上午: 写作, 实习  下午: 听力\n");
else if (day==6 || day==7)
    printf("休息\n");
else
    printf("非法输入\n");
}

```

3.2 将程序 S3-1.C 用 switch 语句改写并实现。

【指导】

(1) 在 switch 语句中, case 后面只能是一个常量。不能将 switch 语句表示为:

```

switch(day)
{ case day==1: printf("上午: 英语, 数学, 政治, 体育  下午: 法律\n");
  case day==2: printf("上午: 物理, 计算机, 美术  下午: 音乐\n");
  case day==3: printf("上午: 英语, 数学, 语文  下午: 锻炼\n");
  .....
}

```

(2) 下面程序段中 switch 语句中的 case 后面的表示都是错误的。

```

int x;
scanf ("%d",&x);
switch(x)
{ case x:
  case x+1:
  case x+2:
}

```

3.3 求水仙花数。如果一个 3 位数的个位数、十位数和百位数的立方和等于该数自身, 则称该数为水仙花数。编程 (S3-3.C) 求出所有水仙花数, 并写出输出结果。

【指导】

该算法的描述如下:

- ① 假设 a, b, c 分别为一个 3 位数 x 的个位、十位和百位上的数, 那么, 当 $a^3+b^3+c^3=x$ 时, 称 x 为水仙花数。
- ② 3 位数从 100~999, 要判断其中的每一个数是否是水仙花数, 可以用循环实现。