

面向21世纪大学计算机基础教育系列规划教材

C 语言程序设计 习题解析与实验指导

范慧琳 主编 叶双 冯姝婷 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

面向 21 世纪大学计算机基础教育系列规划教材

C 语言程序设计习题解析与实验指导

范慧琳 主 编

叶 双 冯姝婷 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是《C 语言程序设计》的配套辅导用书。全书分为两篇，第一篇是与主教材各章对应的习题解析，内容包括习题的解题思路、主要知识点、难点与易错点分析以及题解。为便于期末复习和检验学习效果，另附有 5 套等级考试模拟题和题解，题型包括选择题、改错题、填空题和编程题。第二篇是上机实验指导，分别给出了 Turbo C 2.0 上机指南，Turbo C++ 3.0 上机指南，以及如何在 Turbo C 集成开发环境中调试程序。针对主教材各章的主要内容，精心设计了 12 个具有代表性的实验及相关的实验操作提示，以帮助读者通过实验更好地理解和把握 C 语言程序设计的特性和方法，并为每个实验题目给出了参考答案。

本书可作为高校各专业 C 语言辅导教材，也可作为自学者的参考资料和各类计算机培训的辅助教材。

图书在版编目 (CIP) 数据

C 语言程序设计习题解析与实验指导 / 范慧琳主编.

北京: 中国铁道出版社, 2007.1

(面向 21 世纪大学计算机基础教育系列规划教材)

ISBN 978-7-113-07747-1

I. C… II. 范… 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 (2007) 第 019633 号

书 名: C 语言程序设计习题解析与实验指导

作 者: 范慧琳 叶 双 冯姝婷

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 秦绪好

责任编辑: 苏 茜 杨 勇 王春霞

封面设计: 付 巍

封面制作: 白 雪

责任校对: 刘彦会

印 刷: 北京新魏印刷厂

开 本: 787×1092 1/16 **印张:** 16 **字数:** 373 千

版 本: 2007 年 2 月第 1 版 2007 年 2 月第 1 次印刷

印 数: 1~5 000 册

书 号: ISBN 978-7-113-07747-1/TP·2112

定 价: 21.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签, 无标签者不得销售

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。



面向 21 世纪大学计算机基础教育系列规划教材

编
委
会

主 任： 宁正元

副主任： 俞建家 鄂大伟

委 员： (按姓氏字母先后为序)

陈志明 方春春 黄华国 林逢春

林宏康 刘张榕 翁黎朗 吴忠斌

谢建国 谢金达 曾光清 张秀玉

张银明 赵忠孝

进入 21 世纪以来, 社会信息化进程不断加速, 信息化范围和各行各业信息化的程度都是空前的且不断地快速推进。高等学校非计算机专业的计算机基础教育也由计算机应用基础教育向着信息技术基础教育迈进, 逐步步入更加科学合理、更加符合 21 世纪人才素质需求的新阶段。

为了推动计算机基础教育的深入发展, 上一届“教育部高等学校非计算机专业计算机基础课程教学指导分委员会”曾在系统调研的基础上提出了三个文件, 即《关于进一步加强高等学校计算机基础教学的意见》、《计算机基础教学内容的知识结构与课程设置》和《高等学校非计算机专业计算机基础课程教学基本要求》, 提出了一门“大学计算机基础”加上几门核心课程的“1+x”的课程设置方案, 直接影响并推动着新一轮计算机基础课程的教学改革向着纵深的方向发展。本届“教育部高等学校非计算机专业计算机基础课程教学指导分委员会”秉承上了一届“教育部高等学校计算机专业计算机基础课程教学指导分委员会”的思路, 将陆续推出针对不同专业类别、不同层次院校的计算机基础课程教学的指导性文件; 许多地区、部门和高校也都积极地推行更深入的计算机基础课程教学改革尝试。

教材是体现教学改革、实现教学要求的重要保证。为了更好地促进计算机基础教育的改革, 中国铁道出版社在全国范围内邀请具有丰富经验的计算机基础教学一线教师, 在深入探讨和研究“教育部高等学校非计算机专业计算机基础课程教学指导分委员会”一系列文件的基础上, 组织编写“1+x”方案的系列教材, 争取在近几年内陆续出版。该套教材将力争具备以下特色:

- (1) 体系完整、内容先进, 注重非计算机专业学生的特点, 适应新世纪对人才素质培养的要求。
- (2) 体现课程内容的基础性和系统性, 基本内容准确明晰。
- (3) 注重应用, 技术性、应用性内容的讲解采用任务驱动和案例教学模式, 适应教师指导下的学生自主学习的教学模式, 保证教学效果。
- (4) 强调实践, 重视实验教材和辅导教材建设。
- (5) 适应现代教学手段的采用, 除纸质教材外, 还配有多媒体电子教案, 重视教学网站和教学资源库的建设。
- (6) 追踪课程改革进展, 适时更新版本, 及时反映计算机信息技术的新发展。

我们希望本套教材的陆续出版对深化计算机基础教学改革能有所助益; 并希望得到来自一线师生在使用过程中的宝贵意见和建议, 以便于教材不断改进和完善。

宁正元

2007 年 1 月

计算机是一门理论和实践相结合的学科，实践训练是学习和掌握计算机必不可少的环节。为了配合《C 语言程序设计》的学习，我们同步编写了这本习题解析与实验指导书，章节安排上与主教材保持一致。全书分为两篇：第一篇是与主教材各章对应的习题解析，内容包括习题的解题思路、主要知识点、难点与易错点分析以及题解。为便于期末复习和检验学习效果，另附有 5 套等级考试模拟题和题解，题型包括选择题、改错题、填空题和编程题，在本书附录中配有参考答案。第二篇是上机实验指导，分别给出了 Turbo C 2.0 的上机指南、Turbo C++ 3.0 的上机指南，以及如何在 Turbo C 集成开发环境中调试程序。针对主教材各章的主要内容，精心设计了 12 个具有代表性的实验及相关的实验操作提示，建议实验为 20 学时。实验的编排由浅入深，既有基础性实验，也有综合应用的提高性实验。以帮助读者通过实验更好地理解和掌握 C 语言程序设计的特点和方法，最后为每个实验题目给出了参考答案。

本书全部习题解答和实验程序均由作者在 Turbo C 2.0 集成环境下调试通过。

《C 语言程序设计》多媒体教学课件将于教材正式出版以后制作，届时连同全部例题与习题的源程序文件一起免费提供给使用本教材的教学单位或个人。有需要者请与中国铁道出版社计算机图书中心联系。

教材的编写出版获得华侨大学教材建设基金资助。在本书编写过程中，福建农林大学、泉州师范学院、漳州师范学院和福建师范大学等单位的老师们给出了许多宝贵的建议和意见，中国铁道出版社编辑出版人员为本书的编辑出版付出了辛勤的劳动，在此向他们表示衷心的感谢！

由于时间仓促、水平有限，书中不妥之处在所难免，恳请读者批评指正。

编者

2007 年 1 月

第一篇 习题解析

第 1 章 C 语言概述.....	1
第 2 章 数据类型、运算符与表达式.....	3
第 3 章 数据的输入和输出.....	7
第 4 章 选择型程序设计.....	21
第 5 章 循环控制.....	29
第 6 章 数组.....	42
第 7 章 函数.....	56
第 8 章 编译预处理.....	61
第 9 章 指针.....	62
第 10 章 结构体、共用体与枚举型.....	70
第 11 章 位运算.....	76
第 12 章 文件.....	80
第 13 章 计算机等级考试模拟试卷.....	86
试卷一.....	86
试卷二.....	93
试卷三.....	100
试卷四.....	107
试卷五.....	115

第二篇 上机实验指导

第 14 章 Turbo C 2.0 上机指南.....	124
14.1 Turbo C 开发环境的建立与使用.....	124
14.1.1 Turbo C 的安装.....	124
14.1.2 Turbo C 的启动.....	124
14.1.3 Turbo C 的工作窗口.....	125
14.2 编辑 C 源程序文件.....	127
14.2.1 编辑一个新文件.....	127
14.2.2 编辑一个已存在的文件.....	128
14.3 改变 Turbo C 工作环境.....	129
14.3.1 改变用户工作目录.....	129
14.3.2 重建 Turbo C 工作环境.....	130
14.4 编译、连接和运行.....	132
14.4.1 对单个文件程序的编译和连接.....	132

14.4.2	对多个文件程序进行编译和连接.....	133
14.4.3	查看编译和连接过程中的显示信息.....	134
14.4.4	运行.....	137
14.5	程序动态调试方法.....	138
14.5.1	按步执行方法.....	138
14.5.2	设置断点方法.....	141
第 15 章	Turbo C++ 3.0 上机指南.....	143
15.1	Turbo C++系统的建立.....	143
15.1.1	系统的安装与启动.....	143
15.1.2	Turbo C++工作窗口简介.....	144
15.2	Turbo C 命令的功能与使用.....	145
15.2.1	File 主菜单项.....	145
15.2.2	Edit 主菜单项.....	147
15.2.3	Search 主菜单项.....	149
15.2.4	Compile 主菜单项.....	151
15.2.5	Project 主菜单项.....	153
15.2.6	Options 主菜单项.....	155
15.2.7	Window 主菜单项.....	156
15.3	程序的运行和动态调试.....	157
15.3.1	程序的运行.....	157
15.3.2	程序动态调试方法.....	158
第 16 章	C 程序常见错误与调试要领.....	162
16.1	Turbo C 程序的常见错误.....	162
16.1.1	变量、表达式和语句错误.....	162
16.1.2	数组使用中的错误.....	166
16.1.3	库函数调用错误.....	167
16.1.4	自定义函数中出现的错误.....	168
16.1.5	使用指针变量容易出现的错误.....	170
16.1.6	其他常见错误.....	171
16.2	Turbo C 集成环境调试程序基本要领.....	172
16.2.1	静态检查与动态调试.....	172
16.2.2	编译错误和连接错误的排除.....	174
16.2.3	逻辑错误与运行错误的排除.....	175
16.2.4	程序调试实例.....	177
16.2.5	程序调试命令和热键.....	181
第 17 章	上机实验安排.....	183
17.1	上机实验要求.....	183
17.2	上机实验.....	184

实验一 Turbo C 的使用及 C 程序的结构.....	184
实验二 基本数据类型、运算符与表达式.....	185
实验三 顺序结构程序设计.....	188
实验四 选择结构程序设计.....	191
实验五 循环结构程序设计.....	193
实验六 数组及其应用.....	195
实验七 函数及其应用.....	197
实验八 编译预处理.....	199
实验九 指针.....	200
实验十 结构体、共用体和枚举类型.....	201
实验十一 位运算.....	202
实验十二 文件.....	203
附录 A Turbo C 2.0 常见错误信息.....	204
附录 B 模拟试卷参考答案.....	212
附录 C 上机实验参考答案.....	218
参考文献.....	245

第 1 章 C 语言概述

1. 怎样理解和认识 C 程序的基本结构？

C 程序一定要有且仅有一个主函数（即 main 函数），而函数由语句构成，当执行程序时，从 main 函数的第一条语句开始执行到最后一条语句，程序也随之结束。程序的其他组成部分，如用户自定义函数、预编译命令、注释等是根据需要加入到 C 程序中的。

2. 什么是源程序、编译程序和解释程序？

用高级语言或汇编语言编写的程序称为源程序，必须将其翻译成二进制的机器语言程序后才能执行。编译程序和解释程序就是起到翻译作用的两种程序。

“编译程序”一次翻译产生目标程序，然后执行。再次运行时只要运行其目标程序即可，显然运行速度比较快。但缺点是修改源程序后需要重新编译，否则所作的修改无效。

而“解释程序”按翻译一句执行一句的方式运行。运行速度较慢，但对源程序的每次修改都会反应在运行结果中。

3. 用户自定义的标识符可以和库函数同名吗？

用户自定义的标识符不能与关键字同名，同时最好不要与库函数名和预编译命令同名。

4. 简述从编辑 C 源程序到生成可执行文件并运行的全过程，如图 1-1 所示。

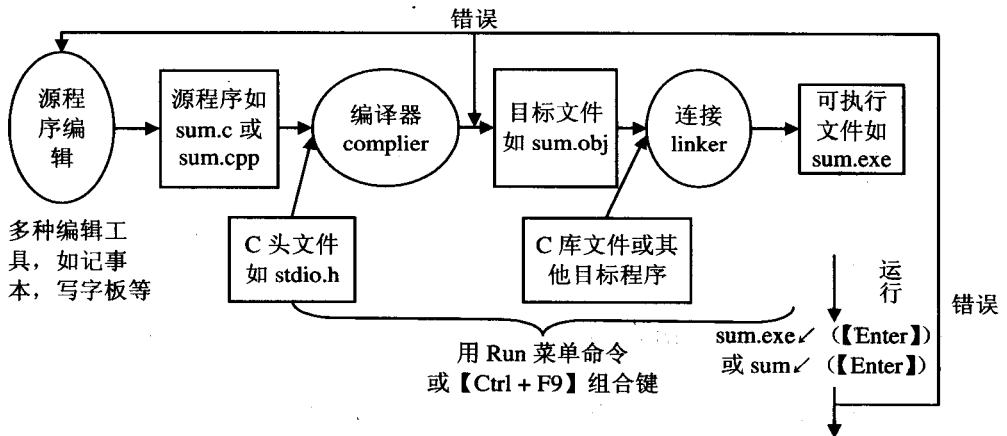


图 1-1 从编辑 C 源程序到生成可执行文件并运行的全过程

5. Turbo C 的 Message 窗口中给出的错误信息行号一定是错误的发生处吗？

不一定。例如，某语句末尾缺少分号，窗口给出的错误信息行号提示在该语句的下一条语句。

6. 程序调试可采用哪几种方法？

可采用以下几种方法。

- 通过打印中间结果，分段排除错误。
- 在程序中设置断点，粗略判断错误位置。
- 对于“死机”或“死循环”等现象，可使用【Ctrl+Break】组合键强制结束，并找到问题的原发位置。
- 综合使用 Run 菜单的单步执行和定点运行功能。

7. 程序的错误类型大概有几种?

程序的错误类型大概有以下几种:

(1) 编译错误: 编译阶段出现的问题, 通常为语法错误(如语句没有以“;”结尾)或警告(如定义的变量没有使用, 但不影响运行结果)。

(2) 逻辑错误: 因程序设计上存在问题而得不到正确结果。

(3) 运行错误: 在运行时才出现的错误, 如除数为 0 等。

(4) 连接错误: 连接阶段出现的错误, 如找不到库文件, 或库文件路径错误等。

8. 请参照本章示例, 编写一个显示“Very good!”的 C 程序。

参考答案:

```
#include <stdio.h>
void main()
{
    printf("Very good!\n");
}
```

解析: 请注意 printf 函数中双引号内\n 的使用, 它起到回车(相当于键盘上的【Enter】键)的作用, 即在显示屏上显示 Very good 后光标移至下行行首。如果没有使用\n, 若再运行其他程序, 其输出结果会接在 Very good 后面, 使得显示不够清晰。

9. 请参照本章示例, 编写一个 C 程序, 显示以下信息:

```
^ ^
@ @
-
```

参考答案:

```
#include <stdio.h>
void main()
{ printf("^ ^\n");
  printf("@ @\n");
  printf("-\n");
}
```

解析: 这里要注意 printf 函数中双引号内空格和回车(\n)的正确使用, 这是初学者比较容易忽略的。

10. 请参照本章示例, 编写一个 C 程序, 计算并显示表达式“(a*b)/c+(x+y)”的值(其中 a=2、b=3、c=2、x=7、y=13)。

参考答案:

```
#include <stdio.h>
void main()
{ int a=2,b=3,c=2,x=7,y=13,result; /*变量 result 用于存放结果*/
  result=(a*b)/c+(x+y);          /*列式计算, 并将结果赋给 result*/
  printf("%d\n",result);          /*输出计算结果 result 的值*/
}
```

运行结果:

```
23
```

解析: 在给变量命名时, 除了要符合用户自定义标识的命名规则之外, 最好有一定的意义(如将存放计算结果的变量命名为 result), 这样可增加程序的可读性。

第2章 数据类型、运算符与表达式

1. C语言的数据类型有哪些？

C语言数据类型包括：基本类型、指针类型、构造类型和空类型4大类。其中基本类型又包含整型、实型、字符类型和枚举类型。构造类型包括数组类型、结构体类型和共用体类型。

2. 什么是变量？什么是变量名、变量的地址、变量的值？试举例说明。

在程序运行时，数据被存放在一定的存储空间中。数据连同其存储空间被抽象为变量；人们是通过变量名来引用变量的；变量的地址指的是变量对应存储空间的首地址；而变量的值即变量对应存储空间中存储的数据的值。例如，程序中通过语句 `int a;` 定义变量 `a`，它表示了一个内存的存储区域及存储的数据。要使用变量 `a` 的值，可直接通过变量名来引用，如 `b=a;` 或 `printf("%d\n",a);` 等。而变量的地址表示为 `&a`，如 `scanf("%d",&a);` 通过 `scanf` 函数读取的数据，将存放在地址 `&a` 所指向的内存的存储区域中。

3. 为什么在引用变量之前必须先定义变量指定其类型？

因为在执行程序时，变量要占据一个内存的存储区域。若不事先定义变量的类型，就无法根据类型分配存储空间，同时也无法正确地执行对该变量的操作，如使用合适的运算符等。

4. 简述整型常量的表示形式。

整型常量可以用十进制、八进制和十六进制表示。其中，以0开头的整数是八进制；以0x开头的整数是十六进制，整型常量没有二进制的表示形式。

5. 简述字符型常量的表示形式。

字符型常量可以使用以下几种形式表示：

(1) 一个字符加单引号。

(2) 字符的ASCII码。

(3) 转义字符，即反斜杠 `\` 加规定的单个字符或字符的ASCII码的八进制和十六进制，并用单引号括起来。

6. 什么是字符常量？在C语言中'a'、'A'和"A"是同一字符常量吗？

不是。由于C语言区分大小写，所以'a'和'A'是不同的。而"A"是字符串常量，除了字符A外，还有一个字符串结束符，因此需要占用两个字节空间。

7. 字符串常量"\\22a,0\n"有效字符的长度是多少？占用的存储空间长度是多少？

有效字符的长度是6，由于还有一个字符串结束符，所以整个字符串占用7个字节的存储空间。

8. C语言中，在ASCII码存储范围内的整数与字符是通用的吗？

是。在ASCII码存储范围内的整数的值即是该字符的ASCII码。

9. 什么是单目、双目和三目运算符？

这里的“目”表示参与运算的操作数的个数，也称做“元”。单目运算符即只有一个操作数的运算符，如`++`、`-`、`!`等。所有的单目运算符的优先级都高于双目和三目运算符，且结合性都是从右向左。双目运算符即有两个操作数的运算符，如`*`、`/`、`>`等，除了赋值运算符以外的双目运算符都是自左向右结合的。而三目运算符即需要三个操作数的运算符。C语言

中只有一个三目运算符，即条件运算符，它的结合性也是自右向左的。

10. 如何理解关系运算和逻辑运算中的“真”和“假”？

在关系运算和逻辑运算中，若关系和逻辑性成立的话，将其视为“真”，否则视为“假”。但这只是人们习惯上的概念。在 C 语言中，最好直接用“非零”和“零”来界定“真”和“假”，才不会引起混乱。

11. 使用符号常量 BANJI 代表学校总的班级数，RENSHUO 代表平均每班学生数，编写程序计算并显示全校的学生数。

参考答案：

```
#include <stdio.h>
#define BANJI 4
#define RENSHUO 45
void main()
{ printf("全校的学生数为%d人\n",BANJI*RENSHUO);
}
```

运行结果：

全校的学生数为 180 人

解析：

(1) #define 是预编译处理命令，所以使用它定义符号常量时，不能写在 main 函数的函数体内。

(2) “预编译”的意思即“编译之前”，也就是在程序被编译（翻译成机器代码）之前，会先执行预编译命令。对于本题来说，就是先将程序中所有的 BANJI 和 RENSHUO 分别用 4 和 45 替代，然后再进行编译，如图 2-1 所示。

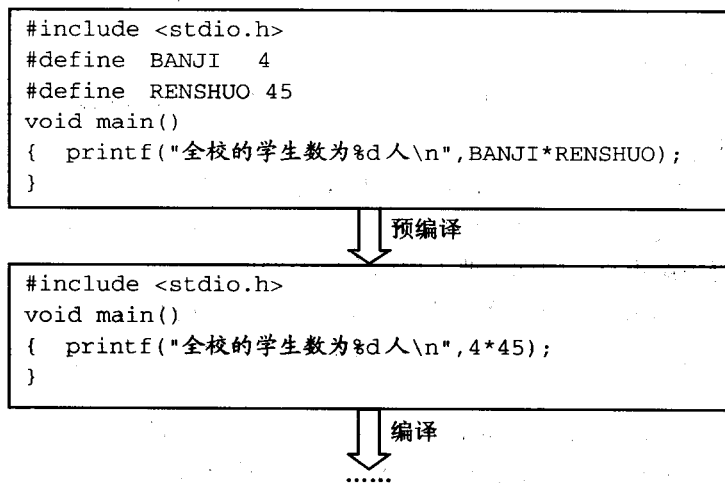


图 2-1 预编译和编译

12. 定义整数变量 x (x=5) 和 y (y=15)，计算并在屏幕上显示表达式“(x+y)*x/y”的值。

参考答案：

```
#include <stdio.h>
```

```

void main()
{ int x=5,y=15;
  float z;
  z=1.0*(x+y)*x/y;
  printf("z=%0.2f\n",z); /*输出结果,保留两位小数*/
}

```

运行结果:

z=6.67

解析:

(1) 当除运算符“/”两侧的运算量都是整数时,其运算结果也是整数(仅取商);当除运算符“/”两侧的运算量只要有一个是实数时,其运算结果也是实数。

(2) 整型数据(int)和实型数据(float或double)相乘结果为双精度类型(double)。

(3) 本题中,在算式 $(x+y)*x$ 前乘以1.0可使“/”运算符左侧数据的类型变为double型,从而使结果为商本身。另外,存放结果的变量z须定义为实数类型。

13. 编写程序把大写字母转换为相应的小写字母。

参考答案:

```

#include <stdio.h>
void main()
{ char ch;
  printf("请输入大写字母: ");
  scanf("%c",&ch);
  ch=ch+32;
  printf("对应的小写字母: %c\n",ch);
}

```

运行结果:

请输入大写字母: G

对应的小写字母: g

解析: 大写字母的ASCII码比相应的小写字母的ASCII码小32。

14. 编写程序分别以字符和整数的形式输出两个不同字符变量的值。

参考答案:

```

#include <stdio.h>
void main()
{ char ch1,ch2;
  scanf("%c%c",&ch1,&ch2);
  printf("%c,%d\n",ch1,ch2);
}

```

运行结果:

ab

a,98

解析: 字符变量可以按字符形式输出,也可以按整数形式(即该字符的ASCII码)输出。同样地,若一个整型变量在0~255(对应unsigned char类型)或-128~127(对应char类型)之间,也可以按字符形式输出。例如:

```

#include <stdio.h>
void main()
{ int i=97,j=-159;

```

```
printf("%c, %c\n",i,j);
}
```

运行结果:

a,a

15. 试用条件运算符编写程序: 输入两个字符, 输出其中较小字符的 ASCII 码值。

参考答案:

```
#include <stdio.h>
void main()
{ char ch1,ch2;
  scanf("%c%c",&ch1,&ch2);
  printf("%d\n",ch1<=ch2?ch1:ch2);
}
```

运行结果:

ba
97

解析: 条件表达式的格式是: 表达式 1? 表达式 2: 表达式 3。执行时, 若表达式 1 的值为非零, 则整个条件表达式的值为表达式 2 的值; 否则, 整个条件表达式的值为表达式 3 的值。

16. 设变量 x=10.2, y=20.5, 编程实现两个变量的值互换。

参考答案:

```
#include <stdio.h>
void main()
{ float x=10.2,y=20.5,z;
  z=x;
  x=y;
  y=z;
  printf("x=%.1f,y=%.1f\n",x,y);/*输出 x 和 y 的值, 保留 1 位小数*/
}
```

运行结果:

x=20.5,y=10.2

解析: 互换两个变量值的时候需要一个中间变量 (如本题的 z) 来辅助。赋值时要注意顺序, 过程如图 2-2 所示。

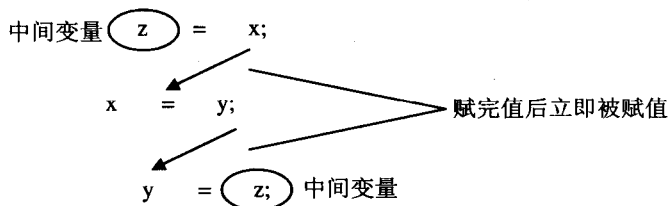


图 2-2 互换两个变量值

第3章 数据的输入和输出

1. 在使用 `printf` 函数和 `scanf` 函数时是否可以省略头文件 `stdio.h` 的说明?

在 Turbo C 2.0 中允许省略头文件的说明。但在其他版本中,如 Turbo C 3.0 等都需要加以说明。

2. 简述 `printf` 函数的函数格式。在使用时,应注意什么问题?

`printf` 函数的参数包括格式控制字符串和输出列表。

其中,格式控制字符串规定了输出的格式,包括普通字符和格式说明。普通字符可以直接输出,而格式说明则规定了输出列表元素的输出格式。

输出列表的元素可以是变量、常量或表达式。格式说明与输出列表元素在数量上应该相等,在类型上应匹配。

若格式说明与输出列表元素的数量不一致,例如:

```
printf("%d,%d\n",2,4,6);
```

输出结果为:2,4。本例再一次强调了“按格式控制字符串规定的格式输出”这一点。

3. 增加域宽与输入精度能否提高输出数据的实际精度?为什么?

不能。因为增加域宽与提高输入精度都只是改变输出数据的输出形式,而不是改变输出数据本身。

4. 使用 `scanf` 函数,当在终端上输入一串字符时,系统如何知道哪几个字符算作一个数据项?

`scanf` 函数有 3 种方法界定一个数据项。

- 根据格式说明中指定的域宽界定。
- 非字符数据项可根据回车、制表符 (Tab) 和空格来界定。
- 根据非法字符界定。

5. 试写出以下程序各 `printf` 语句的运行结果 (空格用 “^” 表示):

```
#include <stdio.h>
void main()
{ short x=5432;
  int i=3456,c='x';
  char z=65;
  printf("\n%d,%o,%x",x,x,x);
  printf("\n%d,%o,%x",-x,-x,-x);
  printf("\n%c,%d,%c,%d",z,z,c,c);
}
```

参考答案:

5432,12470,1538

-5432,165310,eac8

A,65,x,120

解析: 本题涉及的知识点是整型数据的输出,具体包括:

(1) 整型数据以十进制、八进制和十六进制形式的输出。

整型数据以不同进制输出时,要进行进制转换。如十进制 5432 的八进制为 12470,十六进制为 1538。

(2) 负整数的输出 (尤其以八或十六进制的形式输出)。

十进制负数在以十进制输出时不变, 以其他进制输出时可以先将负数转为二进制, 然后再转为八进制和十六进制。例如:

```
printf("\n%d,%o,%x",-x,-x,-x);
```

首先将 $-x$ (即 -5432) 转为二进制 (即 5432 的二进制取反加一) $11101010\ 11001000$, 然后转为八进制 165310 , 十六进制 $eac8$ 。输出为:

```
-5432, 165310, eac8
```

(3) 字符型数据除了以字符形式输出外, 还可以按整数形式输出; 整型数据 (在 $-128\sim 127$ 或 $0\sim 255$ 范围内) 也可以按字符形式输出。换句话说, 在 $-128\sim 127$ (对应 `char` 类型) 或 $0\sim 255$ (对应 `unsigned char` 类型) 范围内的整型数据类型与字符类型可以互换。例如:

```
printf("\n%c,%d,%c,%d",z,z,c,c);
```

字符型变量 $z=65$, 以字符型输出时输出的是 ASCII 码为 65 的字母 'A'; 而整型变量 $c='x'$, 以整型输出时输出的是 'x' 的 ASCII 码 120 。

6. 试写出以下程序各 `printf` 语句的运行结果 (空格用 “^” 表示):

```
#include <stdio.h>
void main()
{ double x=1234.6657867,y=3.45;
  printf("\n%f,%f",x,y);
  printf("\n%4f,%9f",x,y);
  printf("\n%9.4f",y);
  printf("\n%9.2f",x);
  printf("\n%9.0f",x);
  printf("\n%9.4f",-y);
  printf("\n%9.2f",-x);
  printf("\n%9.0f",-x);
}
```

参考答案:

```
1234.665787,3.450000
1234.665787,^3.450000
^^^3.4500
^^1234.67
^^^^1235
^^-3.4500
^-1234.67
^^^-1235
```

解析:

本题涉及的知识点是实数的输出, 具体包括以下几点。

(1) 实数的输出格式 `%f`

在该默认输出格式下, 实数的整数部分全部输出, 小数部分输出 6 位数。若小数部分的位数大于 6 位, 则四舍五入; 否则, 补零。例如:

```
printf("\n%f,%f",x,y);
```

输出为:

```
1234.665787,3.450000
```