

基础教育系列



21世纪高校计算机应用技术系列规划教材
谭浩强 主编

C语言程序设计习题解答与 上机指导 (第二版)

林小茶 编著



8

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



21 世纪高校计算机应用技术系列规划教材

谭浩强 主编

C 语言程序设计习题解答与上机指导

(第二版)

林小茶 编著

内 容 简 介

本书是与铁道出版社出版的《C 语言程序设计（第二版）》（林小茶编著）配套的习题解答与上机指导，同时也适合单独作为学习 C 语言的辅导书。

书中内容分为两部分，第一部分是习题解答，这部分内容对教材中的大部分习题都给出了详细的解答，有些习题还给出了不同的解题方法，这些习题是作者多年以来在教学中积累、收集并经过验证的习题，全部通过了上机调试；第二部分上机指导中介绍了程序设计实验的一般步骤，据此给出了一份上机报告样例，并精心设计了 9 个上机实验，每个实验都给出了实验目的和要求、实验指导以及实验作业，帮助读者在实验中巩固所学知识；附录 A 总结了常用的 Visual C++ 6.0 动态调试命令，附录 B 介绍了 Turbo C++ 3.0 集成环境中比较常用的命令的使用方法。

本书适合作为大学本科应用型专业的学生以及高职高专学生学习“C 语言程序设计”课程的辅助教材，同时也可作为 C 语言自学者的辅助教材或参考书。

图书在版编目（CIP）数据

C 语言程序设计习题解答与上机指导/林小茶编著. —2 版. —北京：中国铁道出版社，2007.7

（21 世纪高校计算机应用技术系列规划教材. 基础教育系列）

ISBN 978-7-113-08216-1

I. C… II. 林… 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 122718 号

书 名：C 语言程序设计习题解答与上机指导（第二版）

作 者：林小茶

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：崔晓静 王雪飞

封面制作：白雪

印 刷：北京新魏印刷厂

开 本：787×1092 1/16 印张：15.75 字数：367 千

版 本：2007 年 9 月第 2 版 2007 年 9 月第 1 次印刷

印 数：1~5 000 册

书 号：ISBN 978-7-113-08216-1/TP·2539

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。

21 世纪是信息技术高度发展且得到广泛应用的时代, 信息技术从多方面改变着人类的生活、工作和思维方式。每一个人都应当学习信息技术、应用信息技术。人们平常所说的计算机教育其内涵实际上已经发展为信息技术教育, 内容主要包括计算机和网络的基本知识及应用。

对大多数人来说, 学习计算机的目的是为了利用这个现代化工具工作或处理面临的各种问题, 使自己能够跟上时代前进的步伐, 同时在学习的过程中努力培养自己的信息素养, 使自己具有信息时代所要求的科学素质, 站在信息技术发展和应用的前列, 推动我国信息技术的发展。

学习计算机课程有两种不同的方法: 一是从理论入手; 二是从实际应用入手。不同的人有不同的学习内容和学习方法。大学生中的多数人将来是各行各业中的计算机应用人才。对他们来说, 不仅需要“知道什么”, 更重要的是“会做什么”。因此, 在学习过程中要以应用为目的, 注重培养应用能力, 大力加强实践环节, 激励创新意识。

根据实际教学的需要, 我们组织编写了这套“21 世纪高校计算机应用技术系列规划教材”。顾名思义, 这套教材的特点是突出应用技术, 面向实际应用。在选材上, 根据实际应用的需要决定内容的取舍, 坚决舍弃那些现在用不到、将来也用不到的内容。在叙述方法上, 采取“提出问题-解决问题-归纳分析”的三部曲, 这种从实际到理论、从具体到抽象、从个别到一般的方法, 符合人们的认知规律, 且在实践过程中已取得了很好的效果。

本套教材采取模块化的结构, 根据需要确定一批书目, 提供了一个课程菜单供各校选用, 以后可根据信息技术的发展和教学的需要, 不断地补充和调整。我们的指导思想是面向实际、面向应用、面向对象。只有这样, 才能比较灵活地满足不同学校、不同专业的需要。在此, 希望各校的老师把你们的要求反映给我们, 我们将会尽最大努力满足大家的要求。

本套教材可以作为大学计算机应用技术课程的教材以及高职高专、成人高校和面向社会的培训班的教材, 也可作为学习计算机的自学教材。

由于全国各地区、各高等院校的情况不同, 因此需要有不同特点的教材以满足不同学校、不同专业教学的需要, 尤其是高职高专教育发展迅速, 不能照搬普通高校的教材和教学方法, 必须要针对它们的特点组织教材和教学。因此, 我们在原有基础上, 对这套教材作了进一步的规划。

本套教材包括以下五个系列:

- 基础教育系列
- 高职高专系列
- 实训教程系列
- 案例汇编系列
- 试题汇编系列

其中基础教育系列是面向应用型高校的教材，对象是普通高校的应用性专业的本科生。高职高专系列是面向两年制或三年制的高职高专院校的学生的，突出实用技术和应用技能，不涉及过多的理论和概念，强调实践环节，学以致用。后面三个系列是辅助性的教材和参考书，可供应用型本科和高职学生选用。

本套教材自2003年出版以来，已出版了70多种，受到了许多高校师生的欢迎，其中有多种教材被国家教育部评为普通高等教育“十一五”国家级规划教材。《计算机应用基础》一书出版三年内发行了45万册。这表示了读者和社会对本系列教材的充分肯定，对我们是有力的鞭策。

本套教材由浩强创作室与中国铁道出版社共同策划，选择有丰富教学经验的普通高校老师和高职高专院校的老师编写。中国铁道出版社以很高的热情和效率组织了这套教材的出版工作。在组织编写及出版的过程中，得到全国高等院校计算机基础教育研究会和各高等院校老师的热情鼓励和支持，对此谨表衷心的感谢。

本套教材如有不足之处，请各位专家、老师和广大读者不吝指正。希望通过本套教材的不断完善和出版，为我国计算机教育事业的发展和人才培养做出更大贡献。

全国高等院校计算机基础教育研究会会长
“21世纪高校计算机应用技术系列规划教材”丛书主编

谭浩强

第一版前言

FOREWORD

学过程序设计的人都有一种体会：看别人编写的程序，好像挺明白的，但是一旦要自己编写一个程序，就感觉无从下手。这是因为程序设计是一门对实践环节要求很高的课程，初学者要想真正掌握 C 语言程序设计，最重要的是抓住两个关键环节：一个是多做程序设计的习题，多编程；另一个就是多上机，写在纸上的程序是否正确，最好的办法就是上机验证一下。为此，我们编写了这本习题解答与上机指导。本书在对教材中的习题进行解答的同时，也对一些基本的程序算法和规则进行了详细的分析，希望能够帮助读者尽快掌握 C 语言程序设计的基本规则与编程技巧，并能够熟练运用这些规则与技巧，编写出具有良好风格的应用程序，并最终能够顺利地通过上机调试。

本书的主要内容分为两部分：第一部分习题解答是对教材中大部分习题的详细解答；第二部分实验指导首先通过一个具体问题的解决过程介绍了程序设计实验的一般步骤，并据此给出了一份上机报告样例，同时在这一部分内容中还安排了九个精心设计的实验，每个实验都给出了详细的实验目的、要求和内容；附录 C 语言上机指南对 Turbo C 集成环境中比较常用命令的使用方法进行了详细的讲解，并举了一些实例来说明命令的使用方法。希望通过这些内容能够帮助读者，尤其是初学者掌握 C 语言的程序设计方法，并进一步加深对课程相关内容的理解与掌握。

提供习题解答的主要目的是为读者提供参考和借鉴，在这里要强调一点，程序设计是一个创作的过程，解决一个实际问题的程序肯定不是唯一的，因此，在阅读本书的参考答案之前，希望读者已经独立思考过主教材中的习题，不要急于看答案，这样才有助于程序设计水平的提高，并且不要把本书的参考源程序作为唯一的答案。同时，作者也对一些习题使用了不同的方法和思路，希望能对读者有更多的帮助。本书中所有程序都已经在 Turbo C 2.0 或 Turbo C++ 3.0 环境下调试通过。

本书由林小茶副教授编写。在本书的编写和出版过程中，全国计算机基础教育研究会会长谭浩强教授给予了指导和把关，同时还得到了陈维兴教授的帮助和支持，在此表示最衷心的感谢。

由于编者水平有限，书中难免还存在一些缺点和不足，殷切希望广大读者批评指正。

编者

2004 年 6 月

第二版前言

FOREWORD

本书是与铁道出版社出版的《C 语言程序设计（第二版）》（林小茶编著）配套的习题解答与上机指导。

由于《C 语言程序设计（第二版）》采用了 Visual C++ 6.0 作为调试环境，因此，本书在第一版的基础上作了一些修订。

考虑到初学者要掌握调试工具有一定的难度，因此本书上机指导部分写得非常详细，对每个实验按步骤进行描述，由易而难，由浅入深，并尽量使用图解来说明。学生只要按照书中描述的步骤去做，就可以逐步掌握调试程序的方法，给指导老师带来极大的方便，这在其他教材中是不多见的，也是本书对第一版的主要改进之处。

同时，还从以下几个方面对第一版的内容作了修订：

(1) 对比较复杂的程序设计题目，采取了题目分析、N-S 结构图和程序代码的顺序给出题目的参考答案。

(2) 尽量对习题答案中的程序给出注释，以方便学生自学。

(3) 对所有的程序在 Visual C++ 6.0 环境下重新进行了调试。

(4) 对标有“*”符号的题目没有给出参考答案，方便老师留作业。

(5) 本书分为两大部分，即习题解答和上机指导。增加了附录 A 和附录 B。附录 A 以表格的方式对 Visual C++ 6.0 环境下使用的动态调试命令进行了总结，方便读者查阅。附录 B 是 Turbo C++ 3.0 上机指南，考虑到与第一版教材的衔接和部分师生的需求，保留了 Turbo C++ 3.0 的内容。

感谢章子辰和赵阳阳帮助调试了全部的程序。

最后，借此次本书再版的机会，向使用本书作为教材和学习参考书的教师和其他读者表示衷心的感谢，并殷切希望广大读者对本书的内容和编写方法提出宝贵的意见和建议。

由于编者水平有限，疏漏之处在所难免，请广大读者批评指正。

编者

2007 年 6 月

目 录

CONTENTS

第一部分 习题解答

| | |
|------------------------|-----|
| 第 1 章 C 语言概述..... | 1 |
| 第 2 章 C 语言基本知识..... | 3 |
| 第 3 章 运算符和表达式..... | 10 |
| 第 4 章 顺序和选择结构程序设计..... | 17 |
| 第 5 章 循环结构程序设计..... | 31 |
| 第 6 章 函数..... | 48 |
| 第 7 章 数组..... | 59 |
| 第 8 章 指针..... | 89 |
| 第 9 章 结构体等构造数据类型..... | 114 |
| 第 10 章 文件..... | 142 |

第二部分 上机指导

| | |
|---------------------------|-----|
| 第 11 章 实验的一般步骤..... | 156 |
| 11.1 分析问题..... | 156 |
| 11.2 制定测试计划..... | 157 |
| 11.3 设计程序方案..... | 157 |
| 11.4 编写程序..... | 158 |
| 11.5 上机调试..... | 158 |
| 11.6 实验总结与实验报告内容..... | 159 |
| 11.7 实验报告样例..... | 159 |
| 第 12 章 上机指导..... | 162 |
| 12.1 实验 1 基础练习..... | 162 |
| 12.1.1 实验目的和要求..... | 162 |
| 12.1.2 实验指导..... | 162 |
| 12.1.3 实验作业..... | 166 |
| 12.2 实验 2 基本数据类型与运算符..... | 166 |
| 12.2.1 实验目的和要求..... | 166 |
| 12.2.2 实验指导..... | 166 |
| 12.2.3 实验作业..... | 171 |
| 12.3 实验 3 顺序与选择结构..... | 172 |
| 12.3.1 实验目的和要求..... | 172 |
| 12.3.2 实验指导..... | 172 |
| 12.3.3 实验作业..... | 178 |

| | | |
|--------|--------------------------|-----|
| 12.4 | 实验 4 循环结构 | 179 |
| 12.4.1 | 实验目的和要求 | 179 |
| 12.4.2 | 实验指导 | 179 |
| 12.4.3 | 实验作业 | 185 |
| 12.5 | 实验 5 函数 | 187 |
| 12.5.1 | 实验目的和要求 | 187 |
| 12.5.2 | 实验指导 | 187 |
| 12.5.3 | 实验作业 | 194 |
| 12.6 | 实验 6 数组 | 195 |
| 12.6.1 | 实验目的和要求 | 195 |
| 12.6.2 | 实验指导 | 195 |
| 12.6.3 | 实验作业 | 202 |
| 12.7 | 实验 7 指针 | 203 |
| 12.7.1 | 实验目的和要求 | 203 |
| 12.7.2 | 实验指导 | 203 |
| 12.7.3 | 实验作业 | 209 |
| 12.8 | 实验 8 结构体、联合体和枚举 | 211 |
| 12.8.1 | 实验目的和要求 | 211 |
| 12.8.2 | 实验指导 | 211 |
| 12.8.3 | 实验作业 | 217 |
| 12.9 | 实验 9 文件 | 218 |
| 12.9.1 | 实验目的和要求 | 218 |
| 12.9.2 | 实验指导 | 218 |
| 12.9.3 | 实验作业 | 223 |
| 附录 A | Visual C++ 6.0 的动态调试命令 | 224 |
| 附录 B | Turbo C++ 3.0 上机指南 | 226 |
| B1 | 使用 Turbo C++ 3.0 调试程序的步骤 | 226 |
| B2 | Turbo C 的工作窗口 | 227 |
| B3 | Turbo C 命令的功能 | 229 |
| B3.1 | 【File】菜单 | 229 |
| B3.2 | 【Edit】菜单 | 230 |
| B3.3 | 【Search】菜单 | 231 |
| B3.4 | 【Compile】菜单 | 233 |
| B3.5 | 【Project】菜单 | 234 |
| B3.6 | 【Option】菜单 | 236 |
| B3.7 | 【Window】菜单 | 236 |
| B4 | 程序的运行和动态调试 | 237 |
| B4.1 | 程序的运行 | 237 |
| B4.2 | 程序动态调试方法 | 239 |

第一部分 习题解答

第 1 章 C 语言概述

【1-1】观察生活，写一个日常生活中的程序。

解：略。

【1-2】有三个同样大小的瓶子，一瓶装醋，一瓶装黄酒，剩下一瓶是空的，请用语言描述如何将装醋的瓶子装酒，而将装酒的瓶子改为装醋。

解：

第一步，将醋倒入空瓶中；第二步，将酒倒入原来装醋的瓶中；第三步，将已经装入原空瓶中的醋倒入原来装酒的瓶中。

【1-3】低级语言和高级语言的特点分别是什么？

解：

低级语言的优点是执行速度快，并且可以直接对硬件进行操作，如主板上的 BIOS 及一些设备的驱动程序等。低级语言的缺点也是显而易见的：可读性差，可维护性差，可移植性差，而且用低级语言编写程序的效率低下，并且不能保证程序的质量。

高级语言的出现大大提高了程序员的工作效率，降低了程序设计的难度，并改善了程序的质量。用高级语言编写的程序看起来更像是英语，很容易读懂，使程序具备了良好的可读性和可维护性。用高级语言编写的程序还具有很好的可移植性。

【1-4】简述 C 语言的特点。

解：

C 语言的优点包括：① C 语言同时具备了高级语言和低级语言的特征；② C 语言是结构化程序设计语言，具有结构化程序设计所需的控制语句；③ C 语言支持模块化程序设计，其程序是由函数构成的，对每个函数可以单独编写和调试；④ C 语言具有丰富的数据类型；⑤ C 语言的运算符种类多，功能强大；⑥ C 语言的基本组成部分紧凑、简洁，关键字少；⑦ C 语言有大量标准化的库函数；⑧ 生成的代码质量高。

C 语言的缺点包括：① C 语言比较灵活，在语法上不如一些著名的高级语言（如 PASCAL）严格，错误检查系统不够可靠；② 如果不加以特别注意，C 程序的安全性将会降低。例如对指针的使用没有适当的限制，指针设置错误，可能导致内存中的信息被破坏，如果经常出现这种错误，极有可能导致系统的崩溃。

【1-5】根据主教材第 1 章例 1.1 和例 1.2, 自己编写一个程序, 尽量使用实例中介绍过的规则。程序功能自定。

解:

```
/*-----*/
/*-----calculate ax2+bx+c -----*/
#include"stdio.h"
void main()
{
    int a,b,c,x,sum;
    a=4;
    b=2;
    c=1;
    x=3;
    sum=a*x*x+b*x+c;
    printf("%d*d*d+%d*d+%d=%d\n",a,x,x,b,x,c,sum);
}
```

运行结果:

4*3*3+2*3+1=43

【1-6】编写程序, 显示如图 1-1 所示的图案。

解:

```
/*-----*/
/*-----show rectangle-----*/
#include"stdio.h"
void main()
{
    printf(" *\n");
    printf(" ***\n");
    printf(" *****\n");
    printf("*****\n");
}
```

```

*
***
*****
*****
```

图 1-1 习题 1-6 输出图案

【1-7】编写程序, 显示如图 1-2 所示的界面, 并上机调试该程序。

解:

```
/*-----*/
/*-----login interface-----*/
#include"stdio.h"
void main()
{
    printf("-----\n");
    printf("your name: \n");
    printf("-----\n");
    printf("your password: \n");
    printf("-----\n");
}
```

```

-----
your name:
-----
your password:
-----
```

图 1-2 习题 1-7 输出界面

【1-8】调试 C 程序需要经过几个步骤?

解:

要完成一个 C 程序的调试, 必须经过编辑源程序、编译源程序、连接目标程序和运行可执行程序四个步骤。

第 2 章 | C 语言基本知识

【2-1】 在 C 语言中，标识符的构成规则是什么？

解：

在 C 语言中，标识符的构成规则是：① 以字母或下划线开头；② 在第一个字符后，可以是任意的数字、字母、下划线组成的序列。长度不超过八个字符。

【2-2】 关键字、特定字和用户定义字分别是什么？

解：

关键字是 C 编译程序预先设定的标识符，它们代表固定的意义，用户不能随便使用。若随便使用，可能出现意想不到的错误，虽然能通过编译，但运行结果不对，且不容易检查出错误所在。

特定字是具有特殊含义的标识符。虽然它们不是关键字，但是人们在习惯上把它们看成关键字。所以一般用户定义的标识符也不要使用它们。

用户定义字是用户在程序中按照语法规则定义的标识符。用户定义字可以用来标识用户自己使用的变量、符号常量、数据类型以及函数等。通俗地说，用户定义字就是程序员在程序设计时为变量、常量以及函数起的名字。

【2-3】 为什么要对变量进行“先定义后使用”？

解：

在 C 语言中，所有的变量都必须先定义后使用。这样做的原因是：① 编译系统会根据定义为变量分配内存空间，分配空间的大小与数据类型有关，例如，字符型变量占一个字节，整型变量占两个字节等；② 未经过定义的标识符，系统不允许将其作为变量名使用，这样会给程序员调试程序带来方便；③ 编译系统可以根据变量的类型检查对该变量的运算是否合法。

【2-4】 字符常量和字符串常量有什么区别？

解：

字符常量用单引号括起，字符串常量用双引号括起；字符常量存储时占一个字节，字符串常量存储时除了字符以外，还必须包含字符串终止符。

【2-5】 下列符号哪些可作为 C 语言的用户定义字，哪些不可以，为什么？

- | | | | | |
|---------------|----------|----------|----------|-----------|
| (1) 2m | (2) mean | (3) c | (4) ked | (5) var |
| (6) _n25 | (7) exp | (8) i*j | (9) mn | (10) data |
| (11) finished | (12) a+c | (13) int | (14) n_m | (15) -n |
| (16) n-m | (17) ok? | | | |

解：

可以作为 C 语言用户定义字的是：mean, c, ked, var, _n25, exp, mn, data, finished, n_m。其余不能作为 C 语言的用户定义字，因为它们与用户定义字构成规则不符。2m 由数字

打头; i*j 和 a+c 中间分别含有 “*” 和 “+”; int 是关键字; -n、n-m、ok? 分别含有特殊符号“-” 和 “?”。

【2-6*】 下列哪些常量是不合法的字符型常量? 为什么?

- (1) 65 (2) 'b' (3) 'ab' (4) '\n' (5) '\075'
 (6) '\078' (7) '\\ (8) '\xaf'

解: 略。

【2-7】 下列哪些常量是不合法的整数? 为什么?

- (1) 1234 (2) -11 (3) 01234 (4) 077 (5) 0x12ed
 (6) 0x4001 (7) 09 (8) 2f

解:

不合法的整数包括: 09 和 2f, 因为 0 开头的是八进制常量, 八进制数不包括 9; 而 2f 没有加前缀 0x, 所以是十进制数, 而十进制数不包括 2f。

【2-8*】 下列哪些常量是不合法的浮点数常量? 为什么?

- (1) 12E7 (2) 12E7.2 (3) 0.0 (4) .1E-10 (5) 105
 (6) 0E0 (7) 6.6 (8) 3E0xed

解: 略。

【2-9*】 确定下列常量的十进制值:

- (1) 02 (2) 034 (3) 0x17 (4) '\007' (5) '\011'

解: 略。

【2-10】 编写程序, 使用 scanf 函数和 printf 函数接收和显示各种基本数据类型的数据。

解:

```
#include"stdio.h"
void main()
{
    char ch;                               /*定义字符型变量*/
    int a,b,c,d;                           /*定义整型变量*/
    long j;                                /*定义长整型变量*/
    float e;                               /*定义单精度浮点型变量*/
    double f;                              /*定义双精度浮点型变量*/
    printf("Enter a char:");              /*提示输入字符型数据*/
    scanf("%c",&ch);                      /*接收字符型数据*/
    printf("%c\n",ch);                    /*输出字符型变量的值*/
    printf("Enter a int:");                /*提示输入整型数据*/
    scanf("%d",&a);                        /*接收整型变量*/
    printf("%d\n",a);                     /*输出整型变量的值*/
    printf("Enter a octal int:");          /*提示输入八进制整型数据*/
    scanf("%o",&b);                        /*按八进制接收整型数据*/
    printf("0%o\t",b);                    /*按八进制输出整型变量的值*/
    printf("%d\n",b);                     /*按十进制输出整型变量的值*/
    printf("Enter a hex int:");            /*提示输入十六进制整型数据*/
    scanf("%x",&c);                        /*按十六进制接收整型数据*/
    printf("0x%x\t",c);                   /*按十六进制输出整型变量的值*/
    printf("%d\n",c);                     /*按十进制输出整型变量的值*/
    printf("Enter a unsigned int:");       /*提示输入无符号整型数据*/
    scanf("%u",&d);                        /*按无符号数接收整型数据*/
}
```

```

printf("%u\n", d);          /*输出无符号整型数据*/
printf("Enter a long int:"); /*提示输入长整型数据*/
scanf("%ld", &j);         /*接收长整型数据*/
printf("%ld\n", j);       /*输出长整型变量*/
printf("Enter a float:");  /*提示输入单精度浮点型数据*/
scanf("%f", &e);         /*接收单精度浮点型数据*/
printf("%f\t", e);       /*输出单精度浮点型数据*/
printf("%e\n", e);       /*输出单精度浮点型数据*/
printf("Enter a double:"); /*提示输入双精度浮点型数据*/
scanf("%lf", &f);       /*接收双精度浮点型数据*/
printf("%lf\t", f);     /*输出双精度浮点型数据*/
printf("%le\n", f);     /*输出双精度浮点型数据*/
}

```

运行情况如图 2-1 所示。

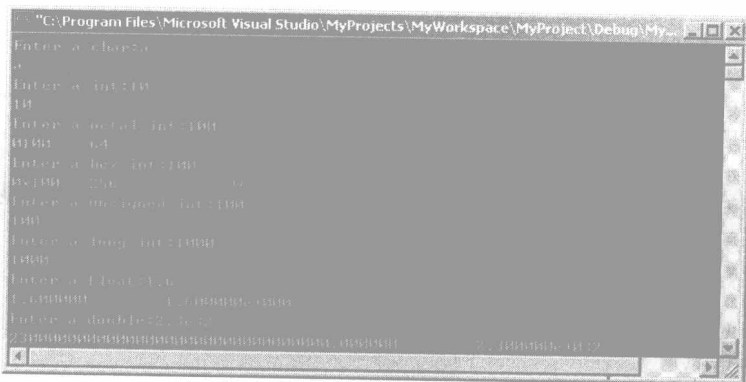


图 2-1 程序执行情况

【2-11】 编写程序，使用 `getchar` 函数接收一个字符，用 `printf` 函数显示；使用 `scanf` 函数接收一个字符，用 `putchar` 函数显示。

解：

```

#include "stdio.h"
void main()
{
    char c;
    printf("Enter a char:");          /*提示输入字符*/
    c=getchar();                    /*接收第一个字符*/
    printf("%c\n", c);              /*输出第一个字符*/
    getchar();                      /*跳过回车键*/
    printf("Enter a char:");        /*提示输入字符*/
    scanf("%c", &c);                /*接收第二个字符*/
    putchar(c);                    /*输出第二个字符*/
    printf("\n");
}

```

【2-12】 编写程序，采取使用 `%s` 输出字符串常量的方法输出下面的字符。

your name:

your password:

解:

方法一:

```
#include"stdio.h"
void main()
{
    printf("%s\n", "-----");
    printf("%s\n", " your name");
    printf("%s\n", "-----");
    printf("%s\n", " your password:");
}

```

方法二:

```
#include"stdio.h"
void main()
{
    printf("%s", "-----\n ");
    printf("%s", " your name:\n ");
    printf("%s", "-----\n ");
    printf("%s", " your password:\n ");
}

```

【2-13】 编写程序, 使用 `getchar` 函数接收一个字符, 用 `putchar` 函数显示; 使用 `scanf` 函数接收一个字符, 用 `printf` 函数显示。

解:

```
#include"stdio.h"
void main()
{
    char c;
    printf("Enter a char:          /*提示输入字符*/
    putchar(getchar());          /*输出接收的字符*/
    getchar();                    /*跳过回车键*/
    printf("\nEnter a char:");    /*提示输入字符*/
    scanf("%c", &c);              /*接收字符*/
    printf("%c\n", c);            /*输出字符*/
}

```

【2-14】 编写程序, 采取使用 `%c` 输出字符变量的方法输出如图 2-2 所示图案。

解:

```
#include"stdio.h"
void main()
{
    char a='*', s=' ', n='\n';
    printf("%c%c%c%c%c%c%c", s, s, s, a, s, s, s, n);
    printf("%c%c%c%c%c%c%c", s, s, a, a, a, s, s, n);
    printf("%c%c%c%c%c%c%c", s, a, a, a, a, a, s, n);
    printf("%c%c%c%c%c%c%c", a, a, a, a, a, a, a, n);
}

```

```

*
***
*****
*****

```

图 2-2 习题 2-14 输出图案

注意: 本题的要求是用 `%c` 输出字符变量, 所以采用以下语句:

```
printf("%c%c%c%c%c%c%c", s, s, s, a, s, s, s, n);
```

【2-15*】 编写程序, 接收两个双精度浮点数, 输出它们的和。

解: 略。

【2-16】 编制一个关于程序中内存变量的列表，根据下列程序填写内存变量表变化的情况。

```
#include"stdio.h"
void main()
{
    double base;
    double height;
    double area;
    base=3.5;
    height=6.7;
    area=base*height/2.0;
    printf("The area of the triangle is %f\n",area);
}
```

解:

内存变量的变化如表 2-1 所示。

表 2-1 内存变量的变化

| 变 量 名 | 值 的 变 化 |
|--------|-----------|
| base | 3.5 |
| height | 6.7 |
| area | 11.725000 |

【2-17】 请修改下面的程序，使其能够通过编译。

```
Void Main
{
    int a=6;b=8;
    int c;
    c=a*b;
    printf(' %d',c);
}
```

解:

该程序有以下几点错误：`void main` 是关键字，应该使用小写字母；`b` 在定义之前就被引用了；“`printf(' %d',c);`”的控制字符串使用单引号是错误的。

正确的程序是：

```
#include"stdio.h"
void main()
{
    int a=6,b=8;
    int c;
    c=a*b;
    printf("%d",c);
}
```

【2-18】 请修改下面的程序，使其能够正确运行。

```
void main();
{
    char c
    getchar(c);
    printf("%s",c);
}
```

解:

该程序有以下几点错误：`void main()`后面不能有分号；`char c`后面应该有分号；`getchar()`函数没有参数，正确的调用方式是“`c=getchar()`”；`c`是单个字符，不能用`%s`输出。

正确的程序是:

```
#include"stdio.h"
void main()
{
    char c;
    c=getchar();
    printf("%c",c);
}
```

【2-19】请修改下面的程序,使其能够将输入的整数在屏幕上正确显示。

```
void main()
{
    int i;
    scanf("%d",i);
    printf("%d",i);
}
```

解:

该程序中有一个错误是初学者非常容易犯的,就是“scanf("%d",i);”中的变量名前面没有加取地址符。

正确的程序是:

```
#include"stdio.h"
void main()
{
    int i;
    scanf("%d",&i);
    printf("%d",i);
}
```

【2-20*】请修改下面的程序,使其能够将输入的浮点数在屏幕上正确显示。

```
void main()
{
    double x;
    scanf("%lf",x);
    printf("%d",x);
}
```

解:略。

【2-21】请修改下面的程序,使其能够正确运行。

```
void main();
{
    float f=7.12;
    char c="c";
    printf("%d\n",int(f%3));
    printf("%c",c);
}
```

解:

该程序有以下几点错误:“void main()”后面的分号是错误的;c是字符型变量,不能用字符串常量赋值;f是浮点数,不能进行取模运算,应该将f强制转换为整型数据。

正确的程序是:

```
#include"stdio.h"
void main()
{
```