

新世纪高等院校计算机应用技术系列教材

C 语言程序设计

(第二版)

王声决	罗 坚	主 编
徐文胜	李雪斌	副主编
黄明和	聂承启	主 审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

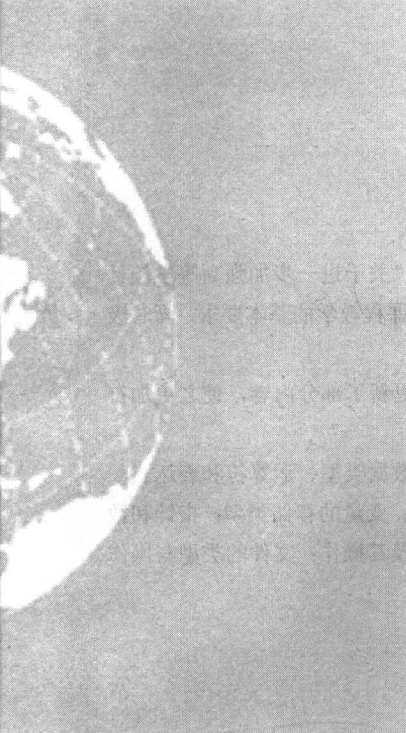
清华大学出版社

C语言程序设计

(第二版)

—— 面向大学非计算机专业
—— 清华大学出版社

清华大学出版社



新世纪高等院校计算机应用技术系列教材

C 语言程序设计（第二版）

王声决 罗 坚 主 编

徐文胜 李雪斌 傅清平 副主编

黄明和 聂承启 主 审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是在第一版的基础上,根据教育部高等学校计算机教育指导委员会“关于进一步加强高等学校计算机基础教学的意见”的计算机基础教育白皮书的精神和计算机程序设计基础课程教学的基本要求,同时汲取了广大读者的意见,修订而成。

本书在保持与原书的体系结构和编写原则一致的情况下,修改、补充和更新了部分内容,使之更加符合教学规律,具有更强的实用性。

全书共分8章,分别介绍了C程序的基本构成与Turbo C的使用;基本数据类型、运算符和表达式、基本输入与输出操作;算法的概念和结构化程序设计的3种方法;函数的使用、变量的存储类型;指针和数组的使用、动态内存分配与动态数组和字符串函数;结构和联合的使用、链表及其操作;文件的类型与操作以及面向对象的概念和C++的基本语法现象。

本书适合作为高等院校的教科书,也可作为广大编程爱好者的自学读物。

图书在版编目(CIP)数据

C语言程序设计/王声决,罗坚主编. —2版. —北京:
中国铁道出版社,2005.11(2006.12重印)
(新世纪高等院校计算机应用技术系列教材)
ISBN 7-113-06794-8

I.C... II.①王... ②罗... III.C语言—程序设计
—高等学校—自学参考资料 IV.TP312

中国版本图书馆CIP数据核字(2005)第141488号

书 名: C语言程序设计(第二版)

作 者: 王声决 罗 坚 等

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 马 建

责任编辑: 严 力 赵 轩 贾 星

封面设计: 薛 为

封面制作: 白 雪

责任校对: 李 旸

印 刷: 北京京海印刷厂

开 本: 787×1092 1/16 印张: 20.5 字数: 495千

版 本: 2003年1月第1版 2005年12月第2版 2006年12月第6次印刷

印 数: 18 001~20 000册

书 号: ISBN 7-113-06794-8/TP·1659

定 价: 29.00元

版权所有 侵权必究

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。

第二版前言

自本书第一版出版以来,得到了全国许多高校专家、教师 and 广大同学们的支持,大家在使用过程中给予了我们许多热情洋溢的鼓励和中肯的意见,这些使我们深感鼓舞,对大家的关心与支持表示衷心的感谢!

进入 21 世纪以后,我国高校计算机基础教育面临了新的机遇与挑战。教育部高等学校计算机教育指导委员会发表了“关于进一步加强高等学校计算机基础教学的意见”的计算机基础教育白皮书(俗称白皮书)。白皮书在深入分析我国计算机基础教育现状的基础上,就进一步改进我国高校的计算机基础教育提出了 11 点意见,对指导我国高校的计算机基础教育有着重要的现实意义。

本书作者根据“白皮书”的精神和计算机程序设计基础课程教学的基本要求,同时汲取了广大读者的意见,对《C 语言程序设计》进行了修订,改编成《C 语言程序设计》(第二版)。第二版保持了第一版的体系结构和编写原则,同时在以下几个方面做了较大的修改补充。

(1) 对“算法与程序设计基础”一章例题中原来用传统流程图表示的算法,改用 N-S 结构流程图来表示,同时增强了对例题的讲解。为尽快引导读者学会编写 C 语言程序,给出了一个具体的例题来描述结构化程序设计方法。

(2) 对“函数”一章在参数传递和递归方面进行了更加详细的叙述;将分程序纳入“变量的作用域”一节,使内容衔接更流畅。

(3) 突出了数据类型的概念,以全新的视角介绍数组、指针、结构、联合等复杂数据类型,为面向对象等高级程序设计方法的学习以及计算机专业理论课程的学习奠定了基础。

(4) 强调了文件的顺序访问与随机访问的概念,补充了如何在 MS-DOS 方式下进行文件操作的上机步骤,并增删了本章的部分示例。

(5) 补充了附录 B 的内容。满足了学生在 Visual C++ 平台下调试 C 程序的需要。

(6) 对各章的文字仔细地做了一些修改;例题程序进行了修订和补充,使得整书的内容更加循序渐进,使读者更容易理解。

为配合本书的教学,我们还编写了《C 语言程序设计学习指导与实验教程》一书。书的内容除了包含了本书全部习题的详细解答之外,还根据教学内容安排了上机实验材料和模拟试题,已由铁道出版社出版。

本书第 1 章、第 2 章、附录 A 和附录 B 由王声决编写,第 3 章由傅清平编写,第 4 章和附录 C 由李雪斌编写,第 5 章、第 6 章由徐文胜编写,第 7 章、第 8 章和附录 D 由罗坚编写。全书由黄明和教授、聂承启教授担任主审。

在本书的编写过程中,杨印根、敖小玲、刘申之、汪浩、李建元、吴克捷、熊刚、王昌晶、傅玲莉、聂伟强、王丽君、王围纬、刘洪、廖云燕、徐培、张捷等教师对本书提出了许多宝贵的意见,桂训泉、尹红、郭奇峰、王祖勤、张建平、李渊姗、黄惠、胡敏、熊剑等同志对本书的成稿提供了很大帮助,在此一并表示衷心感谢。还要感谢中国铁道出版社的领导 and 编辑们对我们编写工作的帮助!更要感谢广大读者的关心和支持!

由于编者水平有限,书中难免存在错误与不足,恳请读者批评指正。

编者

2005 年 9 月于江西师范大学

第一版前言

C 语言是一种非常出色的程序设计语言，它精练、灵活、应用领域广泛，虽然走过了 30 个春秋，至今依然在计算机专业教学和计算机应用程序设计中起着重要作用。

本书根据教育部提出的非计算机专业计算机基础教学三个层次的要求，参照《全国计算机等级考试大纲（2002 年版）》和《全国高等学校计算机等级考试（江西考区）考试大纲》组织编写。作者长期从事高等学校 C 语言课程教学，亲身感受到学生在学习过程中遇到的各种困难。为了使學生能够在 C 语言的学习过程中始终保持强烈的学习兴趣，领悟到程序设计的奥妙，掌握并使用 C 语言解决本专业的实际问题，作者对书中的内容和写作方法作了精心考虑，使得本书具有以下一些特色：

（1）系统全面。内容安排由浅入深，循序渐进。全书围绕结构化程序设计方法，全面展开 C 语言教学内容，示例丰富，习题难易适中，既有便于语法理解的内容，又有提高学习兴趣和实践编程能力的例子。书中同时引用了部分数据结构的算法实例，有利于读者深入学习计算机相关课程。作为 C 语言的发展，本书最后一章介绍 C++ 的基本内容和面向对象程序设计思想。

（2）实践性强。本书从一开始就强调学习 C 语言最好的方法是上机编写程序，这个观点始终贯穿全书。书中既介绍了使用 Turbo C 调试程序的技术，又介绍了如何在 MS Visual C++ 环境下调试 C 程序。

（3）通俗易懂。在写作方式上既注意到概念的严谨，又考虑到语言叙述的通俗易懂，对一些难以理解的算法和容易混淆的概念使用了图解，使得本书不仅适用于课堂讲授，也便于自学。

全书共分 8 章。第 1 章介绍了 C 程序的基本构成与 Turbo C 的使用。第 2 章介绍了基本数据类型、运算符和表达式、基本输入与输出操作。第 3 章介绍了算法的概念和结构化程序设计的 3 种方法。第 4 章介绍了函数的使用、变量的存储类型。第 5 章介绍了指针和数组的使用、动态内存分配与动态数组和字符串函数。第 6 章介绍了结构和联合的使用、链表及其操作。第 7 章介绍了文件的类型与操作。第 8 章介绍了面向对象的概念和 C++ 的基本语法现象。

本书第 1 章、第 2 章、附录 A 和附录 B 由王声决编写，第 3 章由傅清平编写，第 4 章和附录 C 由李雪斌编写，第 5 章、第 6 章由徐文胜编写，第 7 章、第 8 章和附录 D 由罗坚编写。全书由黄明和教授、聂承启教授担任主审。

在本书的编写过程中，杨印根、敖小玲、刘申之、汪浩、李建元、吴克捷、熊刚、王昌晶、傅玲莉、聂伟强等教师对本书提出了许多宝贵的意见，桂训泉、尹红、郭奇峰、王祖勤、张建平、李渊姗、黄惠、胡敏等同志对本书的成稿提供了很大帮助，在此一并表示衷心感谢。

由于编者水平有限，书中难免存在错误与不足，恳请读者批评指正。

编 者

2002 年 10 月于江西师范大学

目 录

第 1 章 简单的 C 程序设计.....	1
1-1 几个简单的 C 程序.....	1
1-2 C 语言常用符号	7
1-2-1 C 语言的关键字	7
1-2-2 标识符.....	8
1-2-3 其他的符号	8
1-3 C 语言程序的上机调试步骤.....	8
1-4 Turbo C 集成开发环境.....	9
1-4-1 Turbo C 2.0 的安装.....	10
1-4-2 Turbo C 2.0 集成化操作界面	10
1-4-3 Turbo C 2.0 简单操作.....	10
1-5 C 语言的概况	18
1-5-1 C 语言的发展过程	18
1-5-2 C 语言的特点和用途	18
1-5-3 学习建议.....	19
习题一	19
第 2 章 数据类型、运算符和表达式.....	21
2-1 常用的进位制.....	21
2-1-1 二进制数、八进制数和十六进制数.....	21
2-1-2 十进制数、二进制数、八进制数和十六进制数之间的换算	22
2-2 数与字符在计算机内存中的表示方法.....	23
2-2-1 机器数和真值.....	23
2-2-2 原码、反码和补码	23
2-2-3 定点数和浮点数.....	24
2-2-4 字符编码.....	25
2-3 常量.....	25
2-3-1 整型常量.....	25
2-3-2 实型常量.....	26
2-3-3 字符常量.....	27
2-3-4 字符串常量.....	28
2-3-5 符号常量.....	28
2-4 变量.....	29
2-4-1 整型变量.....	29
2-4-2 实型变量.....	32

2-4-3	字符型变量.....	33
2-5	常用运算符与表达式.....	34
2-5-1	算术运算符与算术表达式.....	35
2-5-2	赋值运算符和赋值表达式.....	36
2-5-3	强制类型转换运算符.....	37
2-5-4	加 1、减 1 运算符.....	37
2-5-5	逗号运算符和逗号表达式.....	38
2-5-6	位运算.....	38
2-6	基本输入/输出操作的实现.....	40
2-6-1	字符的输入和输出.....	40
2-6-2	有格式的输入与输出.....	41
	习题二.....	46
第 3 章	算法与程序设计基础.....	51
3-1	算法概述.....	51
3-1-1	算法的概念.....	51
3-1-2	算法的特性.....	52
3-2	算法的常用表示方法.....	53
3-2-1	自然语言表示法.....	53
3-2-2	流程图.....	54
3-2-3	N-S 结构流程图.....	55
3-2-4	伪代码表示法.....	56
3-2-5	用计算机语言表示算法.....	57
3-3	结构化程序设计方法.....	58
3-4	C 语句概述.....	60
3-5	选择结构程序设计.....	63
3-5-1	关系运算符和关系表达式.....	63
3-5-2	逻辑运算符和逻辑表达式.....	63
3-5-3	if 语句.....	66
3-5-4	if 语句的嵌套.....	69
3-5-5	条件运算符和条件表达式.....	72
3-5-6	switch 语句.....	73
3-5-7	选择结构程序设计举例.....	75
3-6	循环程序设计.....	80
3-6-1	goto 语句以及用 goto 语句构成的循环.....	81
3-6-2	while 语句.....	81
3-6-3	do-while 语句.....	83
3-6-4	for 语句.....	85

3-6-5 循环的嵌套.....	88
3-6-6 break 语句.....	89
3-6-7 continue 语句.....	90
3-6-8 循环程序设计举例.....	92
3-7 综合程序应用举例.....	95
习题三.....	103
第4章 函 数	106
4-1 函数概述.....	106
4-2 函数的定义.....	107
4-3 函数的调用与返回值.....	108
4-3-1 实参与形参的区分.....	108
4-3-2 函数的调用.....	110
4-3-3 对被调用函数的声明.....	111
4-3-4 函数的返回语句与返回值.....	113
4-4 函数的参数传递方式.....	115
4-4-1 值传递方式.....	115
4-4-2 地址传递方式.....	116
4-5 函数的嵌套与递归.....	118
4-5-1 函数的嵌套调用.....	118
4-5-2 函数的递归调用.....	120
4-6 变量的作用域.....	123
4-6-1 局部变量.....	123
4-6-2 全局变量.....	124
4-6-3 分程序.....	126
4-7 变量的生存期.....	127
4-7-1 自动变量 (auto).....	127
4-7-2 静态变量 (static).....	128
4-7-3 外部变量 (extern).....	129
4-7-4 寄存器变量 (register).....	130
4-8 内部函数和外部函数.....	131
4-8-1 内部函数.....	131
4-8-2 外部函数.....	132
4-9 编译预处理命令.....	133
4-9-1 宏定义.....	133
4-9-2 文件包含.....	136
4-9-3 条件编译.....	137
习题四.....	138

第 5 章 指针与数组类型	146
5-1 数据类型的定义	146
5-2 指针类型的定义与使用	147
5-2-1 指针与指针类型的定义	147
5-2-2 指针常量与变量	149
5-2-3 指针参数	152
5-2-4 函数指针	153
5-2-5 void 指针	155
5-3 数组类型的定义与使用	157
5-3-1 数组与数组类型的定义	157
5-3-2 一维数组变量与基本操作	157
5-3-3 数组参数	161
5-3-4 二维数组与二级指针	164
5-3-5 动态数组	170
5-3-6 字符数组与字符串	173
5-3-7 main 函数的数组参数	179
习题五	180
第 6 章 结构与联合类型	183
6-1 结构与联合类型的概述	183
6-2 结构类型	185
6-2-1 结构类型与结构变量	185
6-2-2 结构变量的基本操作	187
6-2-3 结构指针与结构数组	190
6-3 链表及其操作	195
6-3-1 链表及其实现	195
6-3-2 链表的基本操作	200
6-3-3 链表的应用	203
6-4 位域与联合类型	207
6-5 枚举类型	210
习题六	212
第 7 章 文 件	215
7-1 文件概述	215
7-1-1 文件的概念	215
7-1-2 文件的分类	216
7-1-3 文件的访问方式	217
7-1-4 文件类型指针	218
7-1-5 文件的处理过程	218

7-2 文件的打开和关闭	221
7-2-1 打开文件的函数	221
7-2-2 关闭文件的函数	222
7-3 文件的顺序读写	223
7-3-1 文本文件的顺序读/写	223
7-3-2 二进制文件的顺序读/写	234
7-4 文件的定位与随机读/写	239
7-4-1 rewind()函数	239
7-4-2 fseek()函数	240
7-4-3 ftell()函数	241
7-5 文件状态检查函数	242
7-5-1 文件读/写结束检查函数 feof()	242
7-5-2 文件出错检查函数 ferror()	244
7-5-3 文件出错复位函数 clearerr()	244
7-6 文件输入/输出小结	245
习题七	245
第 8 章 面向对象技术与 C++	248
8-1 C++的起源和特点	248
8-2 简单的 C++程序	249
8-3 C++程序的上机实现	251
8-4 C++的输入和输出	252
8-4-1 用 cout 输出	252
8-4-2 用 cin 输入	253
8-5 设置函数参数的默认值	254
8-6 内联函数	255
8-7 重载函数	257
8-8 变量的引用	259
8-8-1 引用的概念	259
8-8-2 引用做函数参数	261
8-9 面向对象的概念和思想	262
8-9-1 面向对象的概念	262
8-9-2 面向对象方法与结构化方法的比较	263
8-9-3 面向对象系统的特性	264
8-10 类和对象	265
8-10-1 类的定义	265
8-10-2 对象的定义	268
8-10-3 对象的成员表示	268

8-11 构造函数	269
8-12 析构函数	273
8-13 继承与派生类	274
8-13-1 继承与派生类的概念	275
8-13-2 派生类的定义格式	276
8-13-3 公有派生类	277
8-13-4 私有派生类	278
8-13-5 保护成员	278
8-13-6 派生类的构造函数	279
习题八	281
附录 A 美国标准信息交换码表	284
附录 B Visual C++集成环境下调试标准 C 程序的方法	285
附录 C 常用库函数介绍	290
C-1 输入/输出函数	290
C-2 字符和字符串函数	292
C-3 数学函数	296
C-4 动态地址分配函数	300
C-5 图形函数	301
C-6 其他函数	307
附录 D C 语言编译错误信息	308
D-1 致命错误	308
D-2 一般错误	308
D-3 警告性错误	315
参考文献	317

第 1 章 简单的 C 程序设计

学习新的程序设计语言的最佳途径是尽早地用它编写程序、进行程序的调试，解决实际问题。本章从最简单的第一个程序开始，逐步介绍了 C 语言基于函数的程序结构，变量与常量、算术运算、循环结构、选择结构和基本输入/输出标准函数，使读者了解到一个 C 语言程序的基本框架和书写格式。至于详细的语法规则细节，读者先不必深究，学到有关章节时自然会理解。通过介绍 Turbo C 集成环境的使用，要求读者掌握一个 C 程序的编写、编译、连接、调试，直到成功运行的全过程，并能够上机操作。由于所举的例子并没有用到 C 语言的所有特性，不可能完整地表达出使用 C 语言编程的特点，所以本章最后简要阐述了 C 语言的主要特点，给读者以下各个章节的学习起一个点睛的作用，相信读者通过以下各章节的学习，能够真正的理解到这些特点。

1-1 几个简单的 C 程序

【例 1.1】文字显示：在 DOS 屏幕上显示“hello,world!”。

```
/*-----第一个 C 语言程序举例-----*/
#include <stdio.h>
/*-----定义名为 main 函数，它不接收实参值-----*/
main()
{
    /*main 的语句括在花括号中*/
    /*main 函数调用库函数 printf 在计算机屏幕上显示 hello , world! */
    printf("hello,world! \n");
}
```

程序分析

(1) 注解：夹在“/*”与“*/”之间的字符序列，用于解释该程序是做什么的，目的是使程序更易于理解和提起记忆，在编译时被忽略掉。它们可以在程序中自由地使用，可以出现在程序中的任何位置。字符序列中可以使用空格、制表符或换行符。读者应重视使用注解，养成良好的编程习惯。

(2) main 函数：函数是 C 语言的基本单位，每一个 C 程序，不论大小，都是由一个或多个函数组成的。函数是一个单独的程序模块，用于完成指定的功能。在本例中有两个函数，一个是名字为 main 的函数。一般而言，可以给函数任意命名，但 main 函数是一个特殊的函数名，一个 C 程序不论由多少个文件组成，都只有一个且只能有一个 main 函数，通常称做主函数。任何一个 C 程序都从它开始运行。main 函数常常还要调用其他的函数来协助其完成某些工作，被调用的函数有些是由编程人员自己编写的，有些则由系统函数库提供。本例中的另一个函数就是由系统函数库提供的名为 printf 的函数。函数中的语句用一对花括号{}括起来。本例中的 main 函数只包含一条语句 printf("hello, world! \n");，语句最后有一个分号。

(3) 函数的调用：当要调用一个函数时，先要给出这个函数的名字，然后考虑与调用函数的数据交换。在函数之间进行数据交换的一种方法是让调用函数向被调用函数提供一串叫做实参（变元）的值。函数名后面的一对圆括号用于把这一串实参（实参表）括起来。在本例中，main 函数不要求任何实参，故用空实参表“()”表示。main 函数用“hello, world!\n”作实参调用 printf 函数。

(4) printf 是一个格式化输出库函数，在本例中，它用于在 DOS 屏幕上照原样显示双引号内的字符序列“hello, world! \n”（不包括双引号）。用双引号括住的字符序列叫做字符串。本例中仅使用字符串作为 printf 的实参。

(5) 字符串中的字符序列\n 表示换行符，在显示时它用于表示从下一行的左边开始显示字符。

注意：\n 只表示一个字符。除此之外，C 语言中还有：表示制表符的\t，表示回退符的\b，表示双引号的\"，表示反斜杠符本身的\\。

(6) 文件包含命令#include <stdio.h>：这里的#include 称为文件包含命令，其意义是把尖括号<>内指定的文件包含到本程序中，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为“.h”。因此也称为头文件或首部文件。如果使用了系统提供的库函数，一般应在文件的开始用#include 命令，将被调用的库函数信息包含到本文件中。本例中的#include <stdio.h>是因为调用了标准输入/输出库中的 printf 函数。需要说明的是，C 语言规定对 scanf 和 printf 这两个函数可以省去对其头文件的包含命令。所以在本例中也可以删去第二行的包含命令#include <stdio.h>。

【例 1.2】数值计算：编写程序，计算 t 的值。

$$t = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}$$

这个程序本身只由一个名为 main 的函数组成。它要比上面例子长，增加了一些新的内容，包括变量说明、算术表达式以及格式输出。该程序如下：

```
#include <stdio.h>
main()
{
    int sum;                                /*定义整型变量*/
    sum = 1+1/2+1/3+1/4+1/5;                /*计算累加和的语句*/
    printf("The sum is %d\n", sum);          /*在计算机屏幕上按整型数显示计算结果*/
}
```

程序分析

(1) 变量说明：本例子的主函数体中分为两部分：一部分是说明部分，另一部分是执行部分。在说明部分说明了函数所用到的变量的类型，通常放在函数开始处的可执行语句之前。上一个例子没有使用任何变量，因此没有说明部分。C 语言规定，程序中所有用到的变量都必须先说明，后使用，否则将会出错。说明语句由一个类型名与若干所要说明的变量名组成，int sum; 中的 int 是类型名，sum 是变量名。类型 int 表示所列变量为整型变量（整数不包含小数部分）。float 表示所列变量为浮点变量（浮点数可以有小数部分）。除 int 与 float 之外，C 语言还提供了其他一些基本数据类型，包括：char（字符型）、short（短整型）、long

(长整型)、double (双精度浮点型)。另外, 还有由这些基本类型构成的数组、结构与联合类型, 指向这些类型的指针类型以及返回这些类型的函数, 本书将在后面适当的章节再分别介绍它们。

(2) 赋值表达式: 与 Basic、Pascal、Fortran 语言一样, 数学值的计算使用赋值表达式实现。在 C 中 “=” 称为赋值号, 含义不同于数学中的等号, 而是将其右边的表达式的值赋给左边的变量。

(3) 算术表达式: 笔写的数学计算式在 C 语言中要写成合法的 C 语言算术表达式。在 C 中, 算术运算符包括 “+”、“-”、“*”、“/” 以及取模运算符 “%”。“/” 整数除法要截掉结果中的小数部分。表达式 “x%y” 的结果是 “x” 除以 “y” 的余数, 要求 “x” 和 “y” 均为整数, 余数的符号与被除数相同。顺便提醒的是, C 语言中的乘号要写成 “*”, “1.0/2.0” 表示浮点数 1 除以浮点数 2, 而 “1/2” 表示整型数 1 除以整型数 2, 它们的结果是不一样的。进一步的内容在下一章中介绍。

(4) 格式输出: 这个例子使用了 printf 函数更多的功能。printf 函数是一个通用格式化输出函数, 本书将在下一章做详细介绍。本例中 printf 函数具有两个变元, 第 1 个变元是要打印的字符串, 其中百分号 “%” 表示用第 2 个变元 sum 对其进行替换, “d” 表示按整型数打印 sum 的值。

(5) 本例运行以后, 在 DOS 屏幕上显示如下:

```
The sum is 1
```

显然, 结果是不精确的。为什么呢? 原因在于 1/2、1/3、1/4 和 1/5 在 C 中计算结果都是 0。为了得到更加精确的计算结果, 必须用浮点数代替上面的整型数。

【例 1.3】修改后的程序。

```
#include <stdio.h>
main()
{
    float sum;                                /*定义单精度浮点型变量*/
    sum=1.0+1.0/2.0+1.0/3.0+1.0/4.0+1.0/5.0;    /*计算累加和语句*/
    printf("The sum is %f\n", sum);            /*在计算机屏幕上按浮点型数显示计算结果*/
}
```

程序分析

(1) 上一例结果不精确是因为 1/2 按整数除法, 截取后结果为 0。1.0/2.0 是两个浮点数的除法, 不作截取处理。在 C 语言程序中, 浮点数最好写成带小数点的, 即使该浮点数取的是整数值, 因为这样使程序比较清晰。

(2) %f 对应于单精度的浮点数 sum。表示按单精度浮点数格式显示计算结果。还有其他几种指定显示宽度的格式串:

%8d 显示十进制整数, 至少 8 个字符宽。

%8f 显示浮点数, 至少 8 个字符宽。

%.2f 显示浮点数, 小数点后有两位小数。

%8.2f 显示浮点数, 至少 8 个字符宽, 小数点后有两位小数, 小数点也占 1 个字符。

此外, printf 函数还可以识别如下格式说明: 表示八进制数的 %o、表示十六进制数的 %x、表示字符的 %c、表示字符串的 %s 以及表示百分号本身的 %%。

printf 函数第 1 个变元中的各个%分别对应于第 2 个、第 3 个、…、第 n 个变元，它们在数目和类型上都必须匹配，否则将出现错误。

【例 1.4】计算 t 的值的另一个 C 程序。

对于一个特定任务，可以用多种方法来编写程序。上面的程序直接按照算式写法表达，语句很长，可以想像，如果累加 100 项，累加 1 000 项呢？就写不成了。下面是另一版本的 C 程序，完成同样的计算任务。

```
#include <stdio.h>
main()
{
    int i;
    float sum ;
    sum = 1.0;
    for(i=2;i<6;i++)
        sum=sum+1.0/i;
    printf("The sum is %f\n",sum);
}
```

程序分析

(1) for 循环语句：本例的特点是累加计算，所以可以采用循环语句来实现。for 是一种循环语句，for 后面的圆括号内共包含 3 个部分，它们之间用分号隔开。第 1 部分 $i=2$ 是初始化部分，仅在进入循环前执行一次。第 2 部分是用于控制循环的条件测试部分 $i<6$ ，这个条件要进行求值。如果所求得的值为真，那么就执行循环体（本例循环体中只包含一条语句 $sum = sum + 1.0/i$ ；然后再执行第 3 部分 $i++$ ($i=i+1$)，加步长，并再次对条件求值。一旦求得的条件值为假，那么就终止循环的执行。for 循环语句的循环体可以是单条语句，也可以是用花括号括起来的一组语句。

(2) 条件测试：用于控制循环执行次数的条件测试 $i<6$ 在 C 中称做关系表达式。如果 i 小于 6，其结果是“真”（true），否则是“假”（false），除小于（<）外，还有大于（>）、小于等于（<=）、大于等于（>=）、等于（==）和不等于（!=）等关系运算符，例如 i 不等于 10 写作 $i!=10$ 。

关于循环结构的用法将在第 3 章介绍。

(3) 如果某个算术运算符的运算分量都是整数类型，那么就执行整数运算。如果某个算术运算符的运算分量有一个是浮点类型，而另外一个为整数类型，那么这个整数类型分量在开始运算之前会被转换成浮点类型。所以 $1.0/i$ 不会影响结果的精度。

【例 1.5】扩充上一个程序功能的例子。

编写 C 程序，计算 t 的值，其中 m 由键盘输入

$$t = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{m}$$

C 程序如下：

```
#include <stdio.h>
float func(int x); /*函数声明*/
main()
{ int m;           /*说明部分，定义变量*/
  float c;
```

```

/* 提示用户从键盘输入一个整数*/
printf("\n Please enter a integer number m(m>5):");
scanf("%d",&m);          /*由键盘输入变量 m 的值*/
if(m>5)                   /*如果 m 大于 5 计算累加和*/
{   c=func(m);            /*调用 func 函数计算累加和, 将得到的值赋给 c*/
    printf("\n the sum = %f\n",c); /*输出 c 值*/
}

    return 0;
}

float func(int x)          /*定义 func 函数, 函数值为浮点型, 变元 x 为整型数据*/
{ /*func 函数中的说明部分, 定义本函数中用到的变量 i,sum*/
    /*i 为整型, sum 为浮点型, 同时 sum 赋初值 1.0*/
    int i;
    float sum=1.0;
    for(i=2;i<=x;i++)
        sum=sum + 1.0/i;
    return sum;            /*将 sum 的值返回, 通过 func 带回调用处*/
}

```

程序分析

(1) 由功能划分确定函数划分: 本程序进一步表达了 C 语言基于函数的基本结构。上例中程序的执行过程是, 首先在屏幕上显示提示字符串, 请用户输入一个整数, 回车后计算出累加和并在屏幕上显示。程序需要处理三件事: 从键盘接受输入 m ; 如果 $m>5$ 则计算累加和 (其功能是接受 `main` 函数传递的 m , 计算出 $1+\frac{1}{2}+\frac{1}{3}+\dots+\frac{1}{m}$ 的值, 并把它返回给 `main` 函数); 输出计算结果。所以程序除了一个主函数 `main` 以外, 还包含计算累加和的函数 `func`、处理键盘输入的函数 `scanf` 和处理输出的函数 `printf`, 后两个函数都是系统库函数, 由 Turbo C 系统库提供, 用户只要按规定使用而不必自己编写, 但累加和计算函数 `func` 是一个由用户自己编写的自定义函数。

(2) 键盘输入函数 `scanf`: 本程序中的 `scanf` 语句的作用是从键盘上输入一个整型数给整型变量 m 。`%d` 的含义与前面介绍的 `printf` 的用法一致, 表示按十进制整型数输入, `&m` 的含义是将输入的数赋给变量 m , 注意不要漏写 “&”, 其含义在下一章中将详细介绍。

(3) `if` 选择语句: 在求解过程中, 往往需要根据输入的数据做出逻辑判断, 对不同的结果作不同的处理。例如本题如果输入的 m 大于 5 则作累加和计算, 否则不作计算。简单的 `if` 语句格式为:

```

if(<表达式>)
{
    ...
}
...

```

如果表达式的值不为 0, 表示结果为真, 执行花括号内的语句, 否则直接执行花括号下面的语句。作条件判断用的表达式通常可以用关系表达式。

关于选择结构的用法将在第 3 章介绍。

(4) 自定义函数 `func`:

函数定义的一般形式为: