



新大纲

电子工业出版社隆重推出

新编全国计算机等级考试教材

C语言 程序设计(二级)

黄维通 孙宏昌 安颖莲 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

14-1111
73.87429
C818

图章 8 清

新编全国计算机等级考试教材

C 语言程序设计(二级)

黄维通 孙宏昌 安颖莲 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

C语言是目前国内外最为广泛使用的结构化程序设计语言之一。它功能丰富,表达能力强,使用方便灵活,执行效率高,可移植性强,兼具高级语言与汇编语言的特点。本书是按照面向应用、重视实践、便于自学的原则编写的。从C语言最基本的概念入手,由浅入深,综合大量的编程实例,引导初学者从入门到掌握C语言。同时,每一章后面都有大量的习题,在本书的附录中提供了习题的答案。此书全面地阐述了C语言的基本内容及其程序设计技术,并对结构化程序设计技术作了较深入的讨论。其特点是通俗易懂,便于初学者学习和掌握。因此,本书作为计算机等级考试的C语言教材,不仅适用于成人高等教育,也适用于函授大学、夜大学、广播电视大学、职工大学等计算机的C语言程序设计课程,还可供普通高校师生和广大学习C语言程序设计的技术人员参考。

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,翻版必究。

图书在版编目(CIP)数据

新编全国计算机等级考试教材:C语言程序设计(二级)/黄维通等编著. -北京:电子工业出版社,1999.7
新编全国计算机等级考试教材
ISBN 7-5053-5318-7

I. 新… II. 黄… III. C语言-程序设计-水平考试-自学参考资料 IV. TP312

中国版本图书馆CIP数据核字(1999)第08133号

丛 书 名:新编全国计算机等级考试教材

书 名:C语言程序设计(二级)

编 著 者:黄维通 孙宏昌 安颖莲

策 划:胡毓坚

责任编辑:周宏敏

排版制作:电子工业出版社计算机排版室

印 刷 者:北京李史山胶印厂

装 订 者:

出版发行:电子工业出版社 URL:<http://www.pke.com.cn>
北京市海淀区万寿路173信箱 1

经 销:各地新华书店

开 本:787×1092 1/16 印张:18.5 字数:474千字

版 次:1999年7月第1版 1999年7月第1次印刷

书 号:ISBN 7-5053-5318-7
TP·2645

印 数:10100册 定价:24.00元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。
若书店售缺,请与本社发行部联系调换。电话:68279077

前 言

C语言是目前国内外最广泛使用的第三代程序设计语言之一。它处理功能丰富,表达能力强,使用方便灵活,执行程序效率高,可移植性强;兼具高级语言与汇编语言的特点。它具有较强的系统处理能力,可直接实现对系统硬件和外部接口的控制。C语言是一种结构化程序设计语言,它支持自顶向下逐步求精的结构化程序设计技术。另外,C语言程序的函数式结构也为实现程序的模块化设计提供了强有力的保障。因此,它被广泛地用于系统软件和应用软件的开发。著名的UNIX操作系统就是用C语言开发的。因而,C语言也就随着UNIX的出名而闻名。

目前的社会已进入信息化社会,社会经济已向知识型经济发展。为了适应社会的发展,使我国计算机的开发和应用进一步向深度和广度发展,在我国成人教育和函授教育等领域中广泛地开设C语言课程是非常必要的。同时,学习C语言也为进一步学习面向对象的C++程序设计语言和可视化编程打下良好基础。本教材就是基于这种需要而编写的。

本书是按照面向应用、重视实践、便于自学的原则编写的。C语言与其他语言(如BASIC和FORTRAN等)相比,涉及的概念多,规则复杂,书写灵活,容易出错,使初学者感到不易掌握。本书就是为了适应成人教育和其他业余教育学生的情况而写的。其特点如下:

1. 本书通俗易懂,对读者没有特殊要求,使初学者易于学习和掌握本书的基本内容。
2. 本书的内容是按照循序渐进、逐步深入的原则来安排的。把难点分散,使读者学习本书不会感到有太多的困难。
3. 为了帮助读者更好掌握本书的基本内容,特提供了大量程序例题。而且这些例题不要求读者必须具备其他更多的计算机硬件和软件知识就能理解和掌握。这就能使读者更快掌握C语言及其程序设计技巧,以便更快地将它用于实际中。
4. 本书介绍的C语言及其程序例题具有通用性,基本上适合任何计算机系统和C语言的版本。但是应注意,不同的C语言版本是有差别的。
5. 本书介绍的是C语言的基本内容,省略了其他更多的细节(如绘图、系统调用等),这对于初学者抓住主要矛盾,掌握基本内容是重要的。当然,此时用C语言设计大型的程序时还会遇到一些困难,不过到时再参考其他参考书也不会有太多困难。

因此,本书不仅适用于成人高等教育,也适用于函授大学、夜大学、广播电视大学、职工大学等计算机的C语言程序设计课程,也可供普通高校师生和广大学习C语言程序设计的技术人员参考。

本书参照美国国家标准C语言(87 ANSI C)编写,全面地阐述了C语言的基本内容及其程序设计技术,并对结构化程序设计技术作了较深入的讨论。对C语言的指针概念、指针与数组的关系、函数间数据的传递以及结构和联合数据类型等较难理解的内容做了详细和深入的描述,最后一章介绍了用C语言处理动态数据结构的编程技术,以提高读者的实际编程能力。

本书由黄维通、孙宏昌和安颖莲共同编写。在写作过程中得到了清华大学计算机与信息管理中心王行言教授的大力支持和指导,在此表示衷心感谢。

由于作者水平有限,书中错误和缺点在所难免,恳请广大读者批评指正,不胜感激。

黄维通

目 录

第1章 C语言的基本概念	(1)
1.1 C语言的发展与特点	(1)
1.1.1 C语言的发展	(1)
1.1.2 C语言的特点	(1)
1.2 C语言程序的结构特点和书写风格	(2)
1.2.1 构成C语言的基本字符和标识符	(2)
1.2.2 C程序的实例	(4)
1.2.3 C程序的结构特点	(6)
1.2.4 C程序的书写风格	(6)
1.3 C语言程序的编译和执行	(8)
1.4 小结	(9)
1.5 练习题	(10)
第2章 数据类型、运算符、表达式及基本输入输出语句	(11)
2.1 数据类型的一般概念	(11)
2.2 常量	(12)
2.2.1 数	(12)
2.2.2 字符常量	(13)
2.2.3 字符串常量	(13)
2.2.4 转义字符	(14)
2.2.5 符号常量	(14)
2.3 变量及其说明	(15)
2.3.1 变量	(15)
2.3.2 基本数据类型	(16)
2.3.3 变量的定义	(16)
2.3.4 变量的初始化	(17)
2.4 数据类型转换	(17)
2.4.1 隐式类型转换	(17)
2.4.2 显式类型转换	(19)
2.5 运算符和表达式	(20)
2.5.1 运算符和表达式概述	(20)
2.5.2 算术运算符及算术表达式	(22)
2.5.3 赋值运算符和赋值表达式	(23)
2.5.4 关系运算符和关系表达式	(24)
2.5.5 逻辑运算符和逻辑表达式	(25)

2.5.6	三项条件运算符	(26)
2.5.7	其他运算符	(27)
2.6	位运算符	(28)
2.6.1	按位取反运算符	(28)
2.6.2	移位运算符	(28)
2.6.3	按位“与”、按位“或”、按位“异或”	(29)
2.7	C语言基本输入/输出函数	(30)
2.7.1	字符输入/输出函数	(30)
2.7.2	字符串输入/输出函数	(32)
2.7.3	格式化输入/输出函数	(33)
2.8	小结	(37)
2.9	练习题	(38)
第3章	C语言程序的控制结构和结构化程序设计	(41)
3.1	算法及结构化程序设计	(41)
3.1.1	算法及其特征	(41)
3.1.2	算法的类型与结构	(42)
3.1.3	结构化程序设计	(44)
3.1.4	构造结构化程序应注意的问题	(44)
3.2	顺序结构程序设计	(46)
3.2.1	赋值语句	(46)
3.2.2	顺序程序设计及举例	(47)
3.3	分支结构程序设计	(49)
3.3.1	if-else分支	(49)
3.3.2	if分支	(50)
3.3.3	条件分支的嵌套	(51)
3.3.4	if-else if结构	(52)
3.3.5	switch(开关)分支	(54)
3.3.6	条件分支程序设计举例	(57)
3.4	循环结构程序设计	(60)
3.4.1	while语句	(61)
3.4.2	do-while语句	(61)
3.4.3	for语句	(62)
3.4.4	三种循环的比较	(64)
3.4.5	多重循环	(65)
3.4.6	循环和开关(switch)分支的中途退出	(65)
3.4.7	goto语句	(68)
3.5	结构化程序举例	(69)
3.6	小结	(72)
3.7	练习题	(73)

第4章 数组及其应用	(78)
4.1 一维数组	(78)
4.1.1 一维数组的定义	(78)
4.1.2 一维数组的存储形式	(78)
4.1.3 一维数组的引用	(79)
4.1.4 一维数组的初始化	(79)
4.1.5 一维数组的应用举例	(80)
4.2 多维数组	(81)
4.2.1 多维数组的定义	(81)
4.2.2 多维数组的存储形式	(82)
4.2.3 多维数组的引用	(82)
4.2.4 多维数组的初始化	(83)
4.2.5 多维数组的应用举例	(84)
4.3 字符型数组与字符串	(86)
4.3.1 字符型数组的概念	(86)
4.3.2 字符型数组的初始化	(86)
4.3.3 字符型数组的输入/输出	(87)
4.3.4 字符型数组的应用举例	(88)
4.4 综合应用举例	(89)
4.5 小结	(91)
4.6 练习题	(92)
第5章 指针	(94)
5.1 指针的基本概念	(94)
5.1.1 什么叫指针	(94)
5.1.2 指针的目标变量	(94)
5.1.3 指针运算符	(95)
5.2 指针的定义与初始化	(95)
5.2.1 指针的定义	(95)
5.2.2 指针的初始化	(96)
5.3 指针的运算	(97)
5.3.1 指针的算术运算	(98)
5.3.2 指针的关系运算	(99)
5.3.3 指针的赋值运算	(100)
5.4 指针与数组	(100)
5.5 字符指针和字符串	(102)
5.6 指针数组	(104)
5.6.1 指针数组的概念	(104)
5.6.2 指针数组的应用	(106)
5.7 多级指针	(109)

5.7.1	多级指针的概念	(109)
5.7.2	多级指针应用举例	(111)
5.8	应用举例	(112)
5.9	小结	(113)
5.10	练习题	(113)
第6章	函数	(117)
6.1	函数的定义和引用	(117)
6.1.1	函数的定义	(117)
6.1.2	函数的引用	(118)
6.1.3	C程序的执行过程	(121)
6.2	变量的存储类型及作用域	(122)
6.2.1	自动型变量	(122)
6.2.2	寄存器变量	(123)
6.2.3	静态变量	(124)
6.2.4	外部变量	(126)
6.3	函数间的通讯方式	(130)
6.3.1	传值方式	(130)
6.3.2	地址复制方式	(131)
6.3.3	利用参数返回结果	(133)
6.3.4	利用函数返回值传递数据	(134)
6.3.5	利用全局变量传递数据	(135)
6.4	数组与函数	(136)
6.5	字符串和函数	(138)
6.6	指针型函数	(140)
6.6.1	指针型函数的定义和引用	(140)
6.6.2	指针型函数的应用举例	(141)
6.7	指向函数的指针	(142)
6.7.1	函数指针的概念	(142)
6.7.2	函数指针的应用	(143)
6.8	递归函数与递归程序设计	(146)
6.8.1	递归函数的概念	(146)
6.8.2	递归程序设计	(149)
6.9	命令行参数	(150)
6.10	小结	(152)
6.11	练习题	(154)
第7章	结构体、联合和枚举	(157)
7.1	结构体的说明和定义	(157)
7.1.1	什么叫结构体	(157)
7.1.2	结构体的说明及结构体变量的定义	(157)

7.2	结构体成员的引用与结构体变量的初始化	(160)
7.2.1	结构体成员的引用	(160)
7.2.2	结构体变量的初始化	(161)
7.3	结构体数组	(162)
7.3.1	结构体数组的定义及初始化	(162)
7.3.2	结构体数组的应用举例	(162)
7.4	结构体指针	(165)
7.4.1	结构体指针及其定义	(165)
7.4.2	通过指针引用结构体成员	(166)
7.4.3	结构体指针的应用举例	(167)
7.5	结构体在函数间的传递	(169)
7.5.1	结构体变量的传递	(170)
7.5.2	结构体数组在函数间的传递	(173)
7.6	结构体型和结构体指针型函数	(174)
7.6.1	结构体型函数	(174)
7.6.2	结构体指针型函数	(175)
7.7	结构体嵌套	(177)
7.7.1	什么是结构体嵌套	(177)
7.7.2	嵌套结构体类型变量的引用	(178)
7.7.3	结构体嵌套应用举例	(180)
7.8	联合	(181)
7.8.1	联合的说明及联合变量的定义	(182)
7.8.2	使用联合变量应注意的问题	(185)
7.9	枚举类型	(187)
7.9.1	什么是枚举类型	(187)
7.9.2	枚举类型的说明	(187)
7.9.3	枚举型变量的定义	(187)
7.9.4	如何正确使用枚举型变量	(188)
7.10	自定义类型	(190)
7.10.1	自定义类型(typedef)的含义及表示形式	(190)
7.10.2	自定义类型的优点	(191)
7.11	位字段结构体	(193)
7.11.1	位操作方式	(193)
7.11.2	位字段结构体方式	(194)
7.11.3	位字段结构体的应用	(197)
7.12	动态存储分配及其应用	(199)
7.12.1	动态存储分配	(199)
7.11.2	动态数据结构及链表	(203)
7.13	小结	(209)

7.14 练习题	(213)
第8章 标准库函数和文件系统	(217)
8.1 文件概述	(217)
8.1.1 C语言的文件概念	(217)
8.1.2 文件类型指针	(218)
8.1.3 文件的处理过程	(218)
8.2 一般文件的打开和关闭	(219)
8.2.1 文件打开函数	(219)
8.2.2 文件关闭函数	(221)
8.3 一般文件的读写	(221)
8.3.1 一般文件的字符输入/输出函数	(221)
8.3.2 一般文件的字符串输入/输出函数	(225)
8.3.3 一般文件的格式化输入/输出函数	(227)
8.3.4 二进制形式的输入/输出函数	(231)
8.3.5 文件状态检查函数	(235)
8.3.6 文件定位函数	(237)
8.4 小结	(239)
8.5 练习题	(240)
第9章 C语言的预编译程序	(243)
9.1 “文件包括”语句	(243)
9.2 宏定义	(244)
9.2.1 符号常量的定义	(244)
9.2.2 带参数的宏定义	(246)
9.3 条件编译	(249)
9.4 预定义的宏名和其他预编译语句	(251)
9.4.1 预定义的宏名	(251)
9.4.2 #line	(251)
9.5 小结	(252)
9.6 练习题	(252)
附录A C语言的标准库函数	(254)
附录B ASCII字符图表	(259)
附录C C语言二级水平考试样卷(1998年)	(261)
附录D 二级考试大纲	(277)
附录E 练习题答案	(281)

第 1 章 C 语言的基本概念

1.1 C 语言的发展与特点

1.1.1 C 语言的发展

C 语言是目前国际上广泛流行的一种通用的结构化程序设计语言,它不仅是开发系统软件的理想工具,而且也是开发应用软件的理想程序设计语言。因此,它深受广大程序设计者的欢迎。

C 语言是在 70 年代初由美国贝尔实验室的 D. M. Ritchie 设计的,最初的 C 语言是为了描述和实现 Unix 操作系统提供的一种工作语言而设计的,到了 1973 年, K. Thompson 和 D. M. Ritchie 两个人合作把 Unix 的 90% 以上内容用 C 语言进行了改写,即 Unix 第五版。C 语言和 Unix 操作系统可以说是一对孪生兄弟,它们在发展过程中相辅相成。C 是为开发 Unix 操作系统而研制的,它随着 Unix 的出名而闻名。C 的广泛应用又不断推出新的 C 语言版本,其性能也越来越强。到了 1975 年 Unix 第六版的推出和随着面向对象程序设计的出现,C 语言的突出优点就引起了人们的普遍关注,20 多年来又几经修改和完善,发展到了目前可在微机上运行的 Microsoft C/C++、Turbo C、Quick C、Borland C、Visual C/C++ 等版本。

Brian W. Kernighan 和 Dennis M. Ritchie 根据 1978 年发表的 Unix 第 7 版中的 C 语言写了著名的《The C Programming Language》一书,该书中介绍的 C 语言称为标准 C。1983 年美国国家标准协会(ANSI)根据 C 问世以来的发展制定了一个新的标准,这个 C 语言称为 ANSI C。1988 年 K&R 按照 ANSI C 重新编写了《The C Programming Language》一书。因此,目前称 ANSI C 为新标准,而把 K&R 在 1978 年的书中所介绍的 C 称为旧标准。

1.1.2 C 语言的特点

C 语言之所以能被推广并被广泛使用,概括地说主要有如下特点:

- 中级语言

C 语言是介于高级语言和汇编语言之间的中级语言,利用 C 语言,既可以实现高级语言的程序设计,也能对系统的内存地址进行操作,因此它比其他高级语言更接近硬件系统。

- 结构化程序设计

C 语言是一种结构化的程序设计语言,它具有顺序、选择和循环三种基本结构,使程序设计人员便于使用自顶向下逐步求精的结构化程序设计技术。使用 C 语言来编写的程序易读、易维护。

- 运算灵活

C 语言具有类型丰富和使用灵活的运算符,这些运算符不仅具有一般高级语言所具有的算术运算和逻辑运算的功能,而且还具有位运算和复合运算等功能。

- 数据类型丰富

C 语言具有比较丰富的数据类型,它不仅具有字节型、单精度整数、双精度整数、单精度和双精度实数等基本数据类型;而且还有数组、结构、联合、指针、位域、枚举和用户自定义一类的复合数据类型,这使 C 语言具有很强的数据处理能力。

- 模块化程序设计

C 语言是便于进行模块化程序设计的语言,C 语言程序是由一个个函数所组成,这种结构便于把一个大型程序划分为若干相对独立的模块,模块间通过函数调用来实现相互连接。

- 高效性

C 语言程序可以通过 # define、# include 等预编译程序语句,来使用“宏定义”和实现外部文件的读取和合并;还可使用 # if、# else 等来实现条件预编译等,总之可通过使用预编译功能来提高开发效率。

- 可移植性

C 语言程序具有较高移植性,这是因为 C 语言不同于 FORTRAN 等一类语言。C 语言不包含依赖硬件的输入/输出机制,其输入/输出功能是由独立于 C 语言的库函数来实现。这样就使 C 语言程序本身不依赖于硬件系统,于是便于在不同的机器系统间移植。

1.2 C 语言程序的结构特点和书写风格

任何一种计算机语言,都有特定的语法规则和表现形式,程序的构成规则和书写格式则是其表现形式的重要方面。熟悉 C 程序的构成规则和书写格式则是编写一个好的和正确的程序的前提。

1.2.1 构成 C 语言的基本字符和标识符

C 语言根据其特点,规定了其所需的基本符号和标识符。

(1) 字符集

满足 C 语言文法要求的字符集如下:

① 英文字母 a~z, A~Z

② 阿拉伯数字 10 个(0~9)

③ 特殊符号 28 个:

+ - * / % _ (下划线) = < > & | ^ ~ () [] 空格
 { } ; ? : ' (单引号) " ! #

(2) 标识符

标识符是只起标识作用的一类符号,C 语言的标识符主要用来表示常量、变量、函数和类型等的名字。

C 语言的标识符包括如下三类:

① 保留字

所谓保留字,就是这样一类标识符,其每一个都有特定含义,不允许用户把它们当作变量

名使用,C语言的保留字都用小写英文字母表示,共有如下33个保留字,如表1.1所示。

表1.1 C语言的保留字

auto	break	case	char	const	continue	default
do	double	else	enum	entry	extern	float
for	goto	if	int	long	register	return
short	signed	sizeof	static	struct	switch	typedef
union	unsigned	void	volatile	while		

其中 entry 目前还未由任何编译程序来实现,保留作以后使用。

② 预定义标识符

在C语言中,除了上述保留字外,还有一类具有特殊含义的标识符,它们被用作库函数名和预编译命令,这类标识符我们叫预定义标识符。对于这类标识符,虽然C语言准许程序设计者做其他使用(但这时已不具有系统原先规定的含义),但为了避免混淆和增强程序的可读性,建议读者还是不要把这类标识符再定义为其他标识符(用户定义标识符)使用。

预定义标识符包括预编译程序命令和C编译系统提供的库函数名。其中预编译程序命令有:

define undef include ifdef ifndef endif line

③ 用户定义标识符

用户定义标识符是程序员根据自己的需要定义的一类标识符,用于标识变量、符号常量、用户定义函数、类型名和文件指针等。这类标识符的构成规则如下:

- a. 由英文字母、数字组成,但开头字符一定是字母。
- b. 下划线()起字母的作用,它还可用于一个长名字的描述,如:

checkdiskspaceavailable(specifieddiskdrive)

可写为

check _ disk _ space _ available(specified _ disk _ drive),把名词用下划线隔开,以增加可读性。

- c. 大小写英文字母含义不同,TOTAL、Total、...、total等是完全不同的名字。

d. 一个名字可由许多字符组成,但其长度是有限的,对于ANSI C只有前31个字符有效。对旧标准是前8个字符有效,例如编译程序把userpassword和userpass视为同一个名字。

e. C语言的习惯是:变量名用小写字母,常数名用大写字母,函数名和外部变量用6个字符组成。

f. 不允许把C语言的保留字再定义为用户定义标识符;建议也不要将预定义标识符再定义为用户定义标识符,以增强程序的可读性。

为了使程序清晰、易读,建议在定义标识符时,应注意如下几点:

- 名字要有明确含义,应尽量选用具有一定含义的英文单词来命名,使读者“见其名而知其意”。例如代表总和的标识符用total要比用t好,代表平均数的标识符用average而不用a等。如果所选用的英文单词太长,可采用公认的缩写方式。例如表示职员识别字的标识符可用empid来命名。
- 标识符一般采用常用取简专用取繁的原则。即常用标识符的定义应当既简单又明了。

- 对于由多个单词描述的标识符,建议用下划线将各单词隔开,以增强可读性。例如 average_salary。
- 对于标识变量的标识符,可用特定的字符作其前缀来表示变量的数据类型。例如用“i”表示整数,“l”表示长整数,“c”表示字符型,“sz”表示串类型等。

1.2.2 C 程序的实例

为了说明 C 程序的结构特点,首先我们先看几个简单的 C 程序实例,以便使读者有一个初步的感性认识。

【例 1-1】编写显示字符串“Tsinghua University”的 C 语言程序

具体程序代码如下:

```
#include <stdio.h>
main()
{
    printf("Tsinghua University \n");
}
```

这是一个最简单的 C 程序,它把字符串“Tsinghua University”显示在屏幕上。该程序由一个函数 main() (叫主函数)构成。任何一个程序都必须有此函数,花括号 {} 所括的内容是 main 的函数体,每个 C 语言程序的函数都至少有一对 {}。

printf (...) 是由系统提供的标准库函数,它完成输出功能,C 语言的输出是由函数来完成的,而与系统无关,这是它的特点之一。“Tsinghua University”是要输出的内容。“\n”表示换行字符,它是由“\”和“n”两个字符构成,属转义字符,有关转义字符,在后面将会具体介绍。printf () 后的分号是语句结束符,C 的每一个语句都以“;”终止。

#include 是预编译程序命令,它把头文件“stdio.h”的内容展开在 #include <stdio.h> 所在的行位置处。其中, #include <stdio.h> 也可以写成 #include "stdio.h",“stdio.h”文件中定义了 I/O 库所用到的某些宏和变量。因此,在每一个引用标准库函数的程序中都必须带有该 #include <stdio.h> 命令行。

【例 1-2】计算 5 个数平均值的 C 语言程序

具体程序代码如下:

```
#include <stdio.h>          /* 计算平均值的 C 语言程序 */
main()
{
    float a,b,c,d,e,average;    /* 说明 x,y,z,average 为实数 */
    printf("Please input the five datas \n:");
    scanf("%f %f%f%f%f",&a,&b,&c,&d,&e);    /* 输入 a,b,c,d,e 5 个数 */
    average = (a+b+c+d+e)/5;    /* 计算平均值 */
    printf("\n average = %f \n",average);
}
```

运行该程序时,首先提示你输入 5 个数 a、b、c、d 和 e,然后计算出平均值,并把平均值以如

下形式显示在屏幕上:

```
average = ...
```

在此程序中, `/* */` 是一个注释语句, 其中包含注释的内容, 它在程序的编译过程中不产生任何执行代码, 只是在编程中起到备忘录的作用。"float a,b,c,d,e,average;" 是数据类型说明语句, 它把 a,b,c,d,e 和 average 定义为实型数。值得注意的是, 所有 C 语言程序中的变量, 在使用之前都要定义其数据类型。

"scanf(...);" 是输入语句, scanf() 是格式化输入函数, 它是一个由系统提供的标准库函数, 其后的圆括弧内为参数表, "%f%f%f%f%f" 为格式串, %f 表示实型数格式, 指明给 a,b,c 等要求输入实型数。执行该语句时, 数据从键盘上输入。

"average = (a + b + c + d + e) / 5;" 是赋值语句(或表达式语句), 等号(=)是赋值运算符, 表示把右边表达式的运算结果赋给 average。

"printf("\n average = %f \n", average);" 为输出语句, 它首先在新的一行上输出字符串 "average =", 然后按实型数格式(%f)输出变量 average 的值, 并使光标移至下一行。

【例 1-3】求 a、b 和 c 三个数中的最小值

具体代码如下:

```
#include <stdio.h>                                /* 计算最小值的 C 语言程序 */
main()                                             /* 主函数 */
{
    int a,b,c,imin;                                /* 定义变量类型 */
    printf("please input three datas a,b,c: \n");
    scanf("%d %d %d",&a,&b,&c);                    /* 在此输入 a,b,c 的值 */
    imin = min(a,b,c);                             /* 调用求最小值的函数 */
    printf("\n minimum is %d",imin);              /* 输出结果 */
}

int min(x,y,z)                                    /* 求最小值函数 */
int x,y,z;
{
    int m;
    if(x < y)
        m = x;
    else
        m = y;
    if(m > z)
        m = z;
    return(m);                                     /* 向主程序返回最小值结果 */
}
```

此程序由两个函数组成, 除了主函数 main() 之外, 还有一个计算最小值的函数 min()。"int min(x,y,z)" 说明函数的返回值类型为 int(整型), 函数名字为 min, 函数的参数表为 x,y,z。

"int x,y,z;" 说明各参数的类型。

return(m)将求解结果返回给主函数。

该程序的执行是从 main()函数开始,当主函数执行到 `imin = min(a,b,c)` 语句时,控制被传递给 min() 函数,当执行 `return(m)` 语句时,则结束 min() 函数,控制又被传递给 main() 函数,并把 min() 的计算结果带给 main() 函数。当主函数执行结束时,整个程序的执行也就结束了。

1.2.3 C 程序的结构特点

由上面几个简单的 C 程序实例,我们可以看出 C 程序的结构有如下几个特点:

(1) 一个 C 程序是由一个或多个函数所组成

一个 C 程序中必须有一个主函数,主函数名为 main。其余函数的名字由程序设计者自定。程序的执行是从主函数开始,其他函数都是在开始执行 main 函数以后,通过函数调用或嵌套调用而得以执行的。因此,主函数实际上是整个程序的控制部分。主函数以外的其他函数可以是系统提供的库函数,也可以是用户根据自己的需要而编制的函数。所以,我们说 C 语言是函数式语言。为了便于程序设计,各种 C 语言的版本都提供了大量的库函数,供程序设计者引用。

(2) 函数组成

C 函数的定义包括函数说明和函数体两个部分。函数说明指明函数的类型、属性、函数名、参数和参数说明等,如例 1-3 中的 min 函数的说明部分为:

```
int min(x,y,z)
int x,y,z;
```

函数体是花括号所括的部分,它包括局部变量的说明语句和一组执行语句。每个语句都由分号“;”结束。综上所述,一般函数的结构如下:

```
数据类型标识符    函数名(形参表)
形参说明;
{
    局部变量说明语句;
    可执行语句;
}
```

(3) 可包含外部说明

在函数定义之外还可包含一个说明部分,该说明部分叫外部说明,它可包括预编译命令(如上例中的 #include)、外部量的说明等。

1.2.4 C 程序的书写风格

为了写出一个层次清楚、可读性强的程序,我们推荐如下的 C 程序风格:

① 锯齿形

C 程序的风格比较自由,每个语句可从任意列开始。但是,为了使程序层次清楚,应当使

