

C语言程序设计

主编 姜仲秋 副主编 江林升 许卫林
主审 白少布

712

Q/1

出版社

南京大学出版社

JS396/06

C 语 言 程 序 设 计

主编 姜仲秋

副主编 江林升 许卫林

主审 白少布

•

南京大学出版社出版

(南京大学校内, 邮政编码: 210093)

江苏省新华书店经销 江苏地质印刷厂印刷

•

开本 787 × 1092 1/16 印张 15.25 字数 376 千

1998 年 1 月第 1 版 1998 年 1 月第 1 次印刷

印数: 1 — 8000

ISBN 7-305-03124-0/TP • 168

定价: 18.50 元

前 言

C 语言已为广大的计算机工作者所普遍接受和使用,这源于 C 语言的两个基本特点:一是它具有一般的计算机高级语言接近人们思维习惯的特点;二是它具有低级语言直接操纵计算机硬件的能力。这使得 C 语言适于编写系统软件,也适于编写一般数据管理等应用程序,这是 C 语言为计算机专业人员与非计算机专业人员所共同喜爱的根本原因。

本书宗旨是为计算机教学或自学提供合适的教材。考虑读者在学习 C 语言之前一般都接触过其他高级语言,所以,本书内容中 C 语言独具特色的模块设计、指针应用占有较大的篇幅,从教学难点分解的角度,设立了模块设计(一)、模块设计(二)、指针(一)、指针(二)等章节,因而使读者在学习过程中能够对 C 语言的精华有充分的理解。

一般的 C 语言书籍对基本数据、基本语句、基本编程讲解都比较详细,但实际应用指导较少,使得一般读者认为学了 C 语言不实用或认为 C 语言是计算机专业人员使用的,鉴于此,本书专门设立高级程序设计一章,作为在 C 语言基本内容学习完成后实际应用的指导。考虑面向对象程序设计思想(OOP)逐渐为人们所接受,本书的第十三章介绍了面向对象的程序设计的基本知识,作为 C 语言学习者进一步学习的指导。

本书适合于一般的非计算机专业的大专院校学生,计算机专业的中专、技校学生及其他计算机爱好者学习使用。

为适应不同层次的读者学习和不同学时的教学,本教材采用台阶式深度编写方法,共分七级台阶,每一级从知识结构上看都是相对独立完整的。分级情况如下:

- 第一级 第一章—第五章 基本 C 语言程序设计
- 第二级 第一章—第六章 C 语言程序设计
- 第三级 第一章—第七章 C 语言程序设计与指针知识
- 第四级 第一章—第八章 C 程序设计与构造类型数据
- 第五级 第一章—第十章 C 语言模块化程序设计
- 第六级 第一章—第十一章 C 程序设计及应用
- 第七级 第一章—第十三章 C 程序设计及图形编程

以上分级可供学习和教学时参考。

本书由姜仲秋任主编,江林升、许卫林任副主编,白少布主审。其中,第一章、第十三章及附录由李甲林编写,第二章由吴延凤编写,第三、四、十一章由江林升编写,第五、六、十二章由许卫林编写,第七、八、九章由姜仲秋编写,第十章由姜大庆编写。白少布、唐瑞庭等同志在本书编写过程中给予了大力支持,在此深表感谢!

由于时间仓促,本书缺点在所难免,恳请广大读者批评指正。

作 者

1997 年 10 月

目 录

第一章 程序设计与 C 语言	(1)
§ 1.1 计算机语言	(1)
§ 1.2 C 语言的发展与特点	(2)
§ 1.3 数据在计算机中的存储	(3)
§ 1.4 C 语言程序简介	(4)
§ 1.5 在 Turbo C 环境下运行 C 程序	(5)
第二章 数据与基本操作	(7)
§ 2.1 C 语言的数据类型	(7)
2.1.1 数值型数据的表示及存储.....	(7)
2.1.2 字符型数据的表示及存储.....	(8)
2.1.3 无符号数据类型.....	(9)
§ 2.2 C 语言的标识符、常量与变量	(9)
2.2.1 标识符.....	(9)
2.2.2 常量的表示	(10)
2.2.3 变量的表示、定义及初始化	(12)
§ 2.3 C 语言的运算符与表达式.....	(13)
2.3.1 算术运算与表达式	(13)
2.3.2 自增与自减运算符	(14)
2.3.3 逗号运算与逗号表达式	(14)
2.3.4 关系运算与逻辑运算	(15)
2.3.5 赋值运算与表达式	(15)
2.3.6 位运算	(16)
§ 2.4 运算的优先级、结合性	(18)
§ 2.5 不同类型数据间的转换.....	(19)
第三章 C 程序设计初步	(24)
§ 3.1 C 程序的基本组成.....	(24)
3.1.1 程序的基本组成	(24)
3.1.2 C 程序的构成	(24)
§ 3.2 C 语言中的基本语句.....	(25)
3.2.1 C 语言中输出数据的语句	(25)
3.2.2 C 语言中提供数据的语句	(29)
3.2.3 C 语言中数据处理过程的控制语句	(33)
3.2.4 C 语言中的其他功能语句	(33)
§ 3.3 程序举例	(34)
第四章 C 程序的控制流程设计	(37)

§ 4.1	结构化程序设计概述	(37)
§ 4.2	分支程序设计	(38)
4.2.1	if 语句	(38)
4.2.2	条件运算符	(41)
4.2.3	switch 语句	(41)
§ 4.3	循环程序设计	(43)
4.3.1	while 循环结构	(43)
4.3.2	do-while 循环结构	(45)
4.3.3	for 循环	(45)
4.3.4	循环的嵌套	(48)
4.3.5	break、continue、goto 和 exit 语句	(48)
§ 4.4	程序举例	(51)
第五章	数组及其应用	(56)
§ 5.1	一维数组	(56)
5.1.1	一维数组的定义	(56)
5.1.2	一维数组的初始化	(57)
5.1.3	一维数组元素的引用	(58)
5.1.4	一维数组的应用举例	(58)
§ 5.2	二维数组	(60)
5.2.1	二维数组的定义	(60)
5.2.2	二维数组的初始化	(61)
5.2.3	二维数组元素的引用	(62)
5.2.4	二维数组应用举例	(62)
§ 5.3	字符数组和字符串	(64)
5.3.1	字符数组与字符串	(64)
5.3.2	字符数组的定义和初始化	(64)
5.3.3	字符串的输入和输出	(65)
5.3.4	字符串处理函数	(66)
5.3.5	字符数组的应用举例	(68)
第六章	模块设计 (一)	(71)
§ 6.1	模块化软件	(71)
§ 6.2	函数定义和函数声明	(72)
6.2.1	函数定义	(72)
6.2.2	函数声明	(73)
§ 6.3	函数的调用和值的传递	(74)
6.3.1	函数调用的一般形式	(74)
6.3.2	传值调用	(75)
6.3.3	数组作为参数	(76)
6.3.4	嵌套调用	(76)
6.3.5	递归	(77)

§ 6.4 变量的存储类型及作用域·····	(78)
6.4.1 自动变量 auto 及其作用域·····	(78)
6.4.2 寄存器变量 ·····	(80)
6.4.3 静态变量(static)及其作用域 ·····	(81)
6.4.4 外部变量(extern)及其作用域 ·····	(82)
6.4.5 存储类别小结 ·····	(85)
§ 6.5 编译预处理·····	(86)
6.5.1 宏替换 ·····	(86)
6.5.2 文件包含 ·····	(89)
6.5.3 条件编译 ·····	(90)
§ 6.6 程序设计举例·····	(91)
第七章 指针(一) ·····	(95)
§ 7.1 指针与简单变量·····	(95)
7.1.1 指针概述 ·····	(95)
7.1.2 指针变量的定义及赋值 ·····	(96)
7.1.3 指针变量的引用与应用 ·····	(97)
7.1.4 指针运算 ·····	(99)
§ 7.2 指针与数组 ·····	(100)
7.2.1 一维数组的指针及数组单元的访问·····	(100)
7.2.2 二维数组的指针及数组单元的访问·····	(105)
7.2.3 字符串指针·····	(111)
§ 7.3 指针作为函数的参数 ·····	(113)
7.3.1 变量的指针作函数参数·····	(113)
7.3.2 指向数组的指针变量作函数参数·····	(114)
7.3.3 字符串指针作函数参数·····	(121)
第八章 构造类型数据 ·····	(127)
§ 8.1 概述 ·····	(127)
§ 8.2 结构体类型数据 ·····	(127)
8.2.1 结构体类型的定义·····	(127)
8.2.2 结构体类型变量的定义·····	(128)
8.2.3 结构体类型变量的引用·····	(130)
8.2.4 结构体变量的初始化·····	(131)
8.2.5 结构体数组·····	(132)
§ 8.3 共用体类型数据 ·····	(135)
8.3.1 共用体的概念·····	(135)
8.3.2 共用体变量的引用方式·····	(137)
8.3.3 共用体类型变量的应用·····	(137)
8.3.4 共用体类型变量应用的注意点·····	(138)
8.3.5 共用体与结构体的嵌套使用·····	(138)
§ 8.4 枚举类型数据 ·····	(140)

8.4.1	枚举类型的定义与枚举型变量的定义	(140)
8.4.2	枚举类型数据与枚举型变量的说明	(140)
8.4.3	枚举类型数据的应用	(141)
§ 8.5	自定义类型数据	(142)
第九章	指针(二)	(146)
§ 9.1	多级指针	(146)
§ 9.2	指针数组	(147)
§ 9.3	指向函数的指针	(150)
9.3.1	函数的指针	(150)
9.3.2	指向函数的指针变量	(150)
9.3.3	用函数指针实现函数的调用	(150)
§ 9.4	指向结构类型数据的指针	(151)
9.4.1	结构指针	(151)
9.4.2	结构数组的指针	(152)
§ 9.5	动态存储分配与链表	(153)
9.5.1	内存的动态分配	(153)
9.5.2	结构体的自引用与链表	(154)
第十章	模块设计(二)	(163)
§ 10.1	结构体作为函数参数	(163)
10.1.1	将结构元素递给函数	(163)
10.1.2	将整个结构变量传递给函数	(164)
§ 10.2	指针数组作为函数参数	(166)
§ 10.3	函数的指针作为函数参数	(170)
§ 10.4	指向结构体的指针作为函数参数	(172)
§ 10.5	函数返回值类型	(174)
10.5.1	返回值为结构体的函数	(174)
10.5.2	返回指针的函数	(176)
10.5.3	返回结构体指针的函数	(178)
第十一章	文件操作	(184)
§ 11.1	文件、文件系统和流	(184)
11.1.1	文件的概念	(184)
11.1.2	流	(185)
§ 11.2	文件类型指针	(185)
§ 11.3	文件的打开与关闭	(186)
11.3.1	文件的打开(fopen 函数)	(186)
11.3.2	文件的关闭(fclose 函数)	(187)
§ 11.4	文件的读写	(187)
11.4.1	fputc 函数和 fgetc 函数	(187)
11.4.2	fread 函数和 fwrite 函数	(190)
11.4.3	fprintf 函数和 fscanf 函数	(192)

11.4.4 其他读写函数	(192)
§ 11.5 文件定位函数	(193)
11.5.1 rewind 函数	(193)
11.5.2 fseek 函数	(193)
11.5.3 ftell 函数	(194)
§ 11.6 出错检测	(194)
11.6.1 ferror 函数	(194)
11.6.2 clearerr 函数	(194)
§ 11.7 应用举例	(195)
第十二章 Turbo C 屏幕控制与作图	(199)
§ 12.1 文本模式与图形模式	(199)
12.1.1 文本模式	(199)
12.1.2 图形模式	(201)
§ 12.2 菜单程序设计	(205)
第十三章 面向对象程序设计简介	(211)
§ 13.1 面向对象程序设计技术的知识	(211)
§ 13.2 Turbo C++与 Visual C++简介	(213)
§ 13.3 Windows 环境下编程简介	(214)
附录 A Turbo C 集成开发环境简介	(216)
附录 B 常用字符与 ASCII 编码表	(222)
附录 C C 语言关键字及语法提要	(223)
附录 D 运算符优先级和结合方向	(227)
附录 E 常用的 Turbo C 库函数	(229)

第一章 程序设计与 C 语言

§ 1.1 计算机语言

电子数字式计算机是一种能自动地、高效地、精确地进行信息处理的现代化电子设备。不管计算机用在科学计算、数据管理、工业控制的哪一方面，我们都可将计算机看成是一部信息处理机，送入计算机之前的信息称为外部信息（字母、汉字、图表、控制信号的变化等），而送入计算机的信息经转换后已变为计算机的内部信息，计算机内部的信息是由二进制数构成的。外部信息转换为内部二进制数是有严格规定的，英文及半角符号按“美国标准信息交换码(ASCII)”的规定实施转换，汉字及全角符号按“中国国家标准信息交换用汉字编码基本字符集 GB-2312-80”实施转换，而外部计算数据在计算机中有原码、反码、补码等表示方法，还有二进制、十进制、八进制、十六进制等进位制的转换，请在学习本书前参考有关书籍，熟悉以上内容。

一个完整的计算机系统通常包括硬件和软件两部分。

软件和硬件相辅相成，是计算机发挥作用不可缺少的组成部分，两者既有分工又有配合，缺一不可。软件以程序为基础，程序又由计算机语言编写而成，计算机语言分低级语言(含机器语言、汇编语言)和高级语言。

一台计算机通常都配有一套它能识别和执行的由二进制数组成的指令，称作指令系统。计算机也只能识别和执行这些二进制代码组成的指令。由于这种二进制指令是面向计算机的，所以也叫作机器指令。用机器指令编写的程序叫机器语言程序。用机器语言编写程序，要求程序编制者熟悉指令系统和计算机硬件方面的一些有关知识，程序编制工作量大，枯燥繁琐，并且编制的程序难于阅读和修改。由于每一种计算机都有自己的一套指令，导致不同机器之间不能通用，我们称为可移植性差或兼容性差；同时机器语言与人的语言差别太大，因此增加了计算机推广使用的难度。

为使机器语言指令便于记忆，人们用指定的符号代替由二进制代码组成的机器指令，即形成所谓的汇编语言。汇编语言指令由于同指定的机器指令相对应，所以也存在可移植性差或兼容性差的缺点。当然，由于机器语言和汇编语言接近机器硬件，程序运行效率高。

为了解决机器语言和汇编语言的上述缺陷，也为了方便程序的编制，现在计算机系统基本都配有高级程序设计语言，例如 PASCAL、BASIC、C 语言等。高级语言与人们的思维习惯相近，不要求程序编制者深入了解计算机硬件方面的知识，只要按照相应的语言规则，用接近人们习惯的语言和数学公式的形式书写程序(称为源程序)。用高级语言编写的程序易于调试和修改，易于阅读和交流。

但是，计算机不能直接接受和运行高级语言源程序。为了使计算机能识别和接受，必须在计算机中预先配备相应的“编译程序”(或“解释程序”)。编译程序(或解释程序)的作用是充当“翻译”，把人们易于理解的高级语言程序“翻译”成计算机能接受的机器指令代码(称

作目标程序)，以便在计算机上运行。一般高级语言源程序运行步骤如图 1-1 所示。

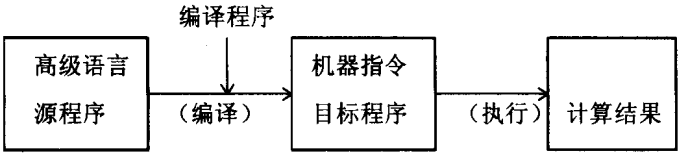


图 1-1 高级语言程序运行过程

§ 1.2 C 语言的发展与特点

C 语言是目前最流行的系统程序设计语言。它具有一般高级语言的特点，面向用户，面向解题的过程，编程者不必熟悉具体的计算机内部结构和指令，易于编程，又可以像汇编语言一样，面对硬件内存单元的位、字节直接进行操作，程序运行效率高。由于这些特性，使 C 语言非常适合用来编写系统软件，并且也可用于编写应用软件。

C 语言发展与中小型计算机上的操作系统 UNIX 是密不可分的。UNIX 系统的早期版本是用汇编语言编写的，由于可移植性差、效率低、编程困难等原因，促使 UNIX 的开发——美国贝尔实验室的 K·汤普逊(Ken.Thompson)和 D·M·里奇(Dennis.M.Ritchie)开发一种高级语言来描述 UNIX 系统。1970 年，他们实现了一种称为 B 的语言，并用 B 语言编写了 UNIX 操作系统和绝大多数上层实用程序。B 语言的主要思想源于 1967 年 M·理查德(M.Richards)提出的 BCPL 语言。BCPL 语言又是基于 ALGOL60 的(见图 1-2)。但由于 B 语言依赖于机器，过于简单，功能有限，又无数据类型，未能流行开来。为了克服 B 语言的局限，D·M·里奇在此基础上设计出了 C 语言，并于 1972 年正式投入运行。C 语言既保持了 BCPL 和 B 语言的优点(精练、接近硬件)，又克服了它们的缺点(过于简单、无数据类型)。

下图反映了 C 语言的发展历史：

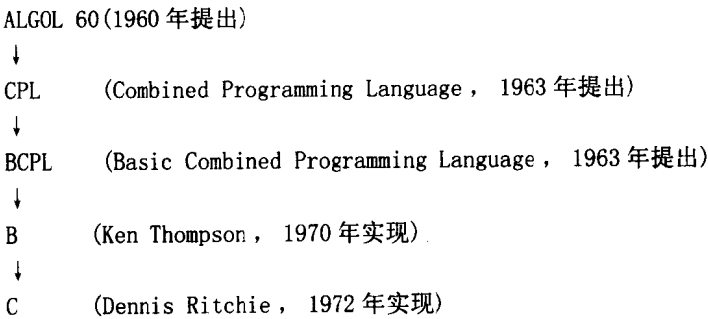


图 1-2 C 语言演变历史

目前，公认的 C 语言版本有 Brian W.Kernighan 和 Dennis M.Ritchie(简称 K&R)合著的《The C Programming Language》，称为标准 C；1983 年美国国家标准化协会(ANSI)根据 C 问世以来的各种版本进行了发展和扩充制定了新的标准，称为 ANSI C。1988 年 K&R 修改了他们的经典著作《The C Programming Language》，按照 ANSI C 的标准重写了该书，1987 年，ANSI 又公布了新标准——87ANSI C，目前流行的 C 语言版本都是以它为基础的。目前

广泛使用于 PC 的 C 语言编译系统有 MS-DOS 支持下的 Turbo C 和 Microsoft C 及 Quick C 等, 随着面向对象编程技术 (OOP) 的出现, 又出现了 Turbo C++ 和 Visual C 等。本书的例题和习题都在 Turbo C 2.0 的编译环境下运行。

C 语言相对于其他高级语言, 有很多比较突出的优点, 概括起来有:

(1) 语言简洁紧凑。

C 语言是一种小型语言, 它一共只有 32 个关键字, Turbo C 多包含了 11 个关键字, 用以实现各种增强和扩展功能, 它们构成了 C 语言的全部指令。虽然 C 语言的关键字比 Pascal 少, 但功能上却毫不逊色。与其对比, IBM PC 机上的 BASIC 语言的关键字竟多达 159 个。C 程序比 Pascal 简炼, 源程序短, 采用的表示法尽量简洁, 如以 { } 代替通常的 Begin、End 作复合语句或函数体的“括号”, 以 ++ 表示加 1, -- 表示减 1, 运算符省写等等。

(2) 表达式能力强。

通常要由机器指令来实现的普通的算术及逻辑运算都可以由 C 语言来完成。它可直接处理字符、数字、地址, 能进行位操作, 能实现汇编语言的大部分功能。正因为兼顾高级语言与汇编语言的优点, 最适宜用来开发系统软件。因此, C 语言又常被称为中级语言。

(3) C 语言数据结构丰富, 并具有很强的控制流结构。

C 语言具有丰富的数据结构, 它可以在基本数据类型 (如: 字符型、整型、实型、无类型等) 的基础上构造新的数据类型 (如: 数组类型、指针类型、结构类型等)。另外, 各种控制语句, 如 if、while、do...while、switch 等, 功能很强, 足以描述结构良好的程序。

(4) C 语言是结构化、模块化的编程语言。

结构化是现代程序编制的基本要求, C 语言具有功能很强的分支、循环等结构化控制语句。C 语言的主要结构成分是函数, 每一个函数都是功能独立的程序, 可以单独编译, 便于程序分块及调试; 用函数作为程序模块则实现程序的模块化。这也使得每个 C 语言编程者都可以根据需要建立和维护属于自己的函数库, 这一函数库可用于许多不同的程序中。不同版本 C 语言还提供标准函数库, 编程时可直接调用相应的库函数, 从而节省大量的编程时间。函数与函数之间除了必要的信息交流以外, 彼此互不影响, 相互隔离。这种方式集中体现了结构化、模块化语言特点: 代码与数据的分隔化。如果使用一个分隔化的函数, 仅需要知道它实现的功能, 而无须知道其功能是如何实现的。

(5) C 语言可移植性能好。

相对于汇编语言, C 语言程序从一个环境不加或稍加修改就可以搬到另一个完全不同的环境上运行。汇编语言依赖于机器硬件, 所以根本不可移植。这种可移植性不仅节省时间, 而且节省费用。

正是基于上述特点, 使得 C 语言得到迅速推广, 成为人们编写大型软件特别是系统软件的首选语言之一。程序员使用 C 语言编写程序会感到限制少、灵活性大、功能强, 是强有力的不可多得的工具。

§ 1.3 数据在计算机中的存储

计算机中的存储器由许许多多的存储单元组成, 每一个单元称为一个“位”(bit, 又称“比特”)。它有两个稳定的工作状态, 分别表示二进制数中一个 bit 的 0 和 1。计算机中的

信息是以二进制形式存储的。内存储器通常由集成电路组成，内存储器容量的大小是衡量微机性能的主要指标之一。为了便于管理，通常取 8 个二进制位作为一个单元，叫做字节(Byte)，

地址	字节的内容
0000H	01000101
0001H	11100001
0002H	10110110
0003H	11000101

它是微机存储器的基本单位。数据在内存中存储时一般是以字节为单位的。例如，一个 ASCII 字符通常占一个字节，一个汉字占两个字节，一个整数占两个字节，一个实数占四个字节等。

内存的每一个字节都被赋予一个唯一的序号，称为地址。CPU 凭借地址，准确地对每个单元的数据存取。图 1-3 为存储单元地址与内容示意图。

图 1-3 存储单元地址与内容示意图

§ 1.4 C 语言程序简介

用 C 语言的语句编写的程序称为 C 语言源程序，一个具有独立功能的程序段可组织为函数，一个 C 语言源文件由一个或多个函数构成，函数是 C 语言的基本构成单位，每个 C 程序都必须有一个名为 main 的主函数，程序的执行总是从主函数开始的。

C 语言程序一般用小写字母，仅在一些宏定义中，将常量名用大写字母表示，对一些有特殊含义的变量，偶而也用大写表示。C 语言中把大小写字母视作两个不同的量。C 语言程序中，一行可以有多个语句，只要每个语句用分号作为结尾即可，多个语句还可用花括号{、}括起来，形成复合语句。C 语言源程序的一般形式如下所示，其中 f1() 至 fN() 代表用户定义的函数。

```
全程变量说明
main()
{ 局部变量说明
  程序段
}
f1()
{ 局部变量说明
  程序段
}
f2()
{ 局部变量说明
  程序段
}
.....
fN()
{ 局部变量说明
  程序段
}
```

其中每个函数名后的一对花括号中的内容为函数体。为了使读者对 C 程序有一个感性认识，下面来看一个简单的 C 程序。

【例 1-1】 在显示器上输出 “How are you!”。

```
main()
{ printf("How are you!"); }
```

本程序即一个 C 语言的源程序，它由一个主函数构成，函数体内只有一个 printf 语句，本程序的功能是在屏幕上输出指定的字符串。

【例 1-2】 求和程序。

```
#include "stdio.h"           /*包含头文件 stdio.h*/
main( )                     /*主函数*/
{ int a,b,c,d;              /*局部变量说明*/
  a=0,b=20,c=30;            /*a,b,c 赋值*/
  d=sum(a,b,c);              /*调用 sum 函数，返回求和值赋给 d */
  printf("sum is %d",d);     /*输出 d 的值*/
}

int sum(x,y,z)               /*定义 sum 函数，函数值为整形， a,b,c 为形参*/
int x,y,z;                   /*对形参 x,y,z 作说明*/
{ int w;                     /*局部变量说明*/
  w=x+y+z;                   /*将 x,y,z 的和赋给 w */
  return(w);                 /*将 w 值返回*/
}
```

本程序中包含了两个函数：主函数 main() 和被调用函数 sum()。sum 函数的作用是将 a,b,c 的和赋给变量 w，用 return 语句将 w 的值返回给主调函数 main。在 main 函数中第 5 行调用 sum 函数，在调用时将实参 a,b,c 分别传送给 sum 函数中的形参 x,y,z。经过执行 sum 函数得到一个返回值(即 sum 函数中变量 w 的值)，把这个值赋给变量 d，然后输出 d 的值。该程序用 /*……*/ 对 C 语言进行注释，在对源程序进行编译时，程序注释将被忽略。恰当使用注释，可增加程序的可读性。

C 语言程序中，有一类函数是编译系统本身提供的，即库函数，在使用这些库函数时，需在程序最前面用 C 语言预处理命令中的包含语句定义这些库函数相关的头文件。本程序中使用了格式化输出库函数 printf()，该函数的头文件是 stdio.h，所以在程序中第一行要用预处理命令 #include 包含此头文件。

§ 1.5 在 Turbo C 环境下运行 C 程序

C 语言程序必须经历编辑、编译、连接过程，才能变成一个可单独执行的程序，即变成一个由机器代码组成的目标程序。

Turbo C 是一个集程序编辑、编译、连接、调试为一体的 C 语言程序开发软件，具有速度快、效率高、功能强等优点，使用非常方便。

Turbo C 一般安装在硬盘 TC 目录下，在 DOS 提示符下键入 TC，回车，即可加载 Turbo C 的集成开发环境(参见附录 A)。

运行一个 C 程序的具体步骤如下：

(1) 输入源程序文件名。选择 File 菜单中 Load 或 New 子菜单, 选择已有文件名或输入新建的源程序文件名, 扩展名为 .C。如果源文件存在, 则将其调入显示在屏幕上。如不存在, 建立一个新文件, 此时进入编辑 (Edit) 状态。

(2) 编辑源文件。在全屏幕编辑方式下, 可以非常方便地完成输入、修改、删除等编辑工作。编辑完成, 为防止程序丢失, 选择 File 菜单中 Save 选项将源程序存入磁盘。

(3) 执行程序。按 Ctrl-F9 键或选择 Run 选项, 可编译、连接、执行输入的源程序。此时, 屏幕上会显示运行结果。根据程序需要, 可能需要输入有关数据, 然后接着执行程序, 输出结果。如来不及看结果, 可用 Alt-F5 或 Run 菜单中的 user screen 看运行结果。如果运行结果不对, 屏幕会提示错误, 这时可按 ALT-E 进入编辑状态查找错误, 直到得到正确结果为止。

(4) 退出 Turbo C。按 Alt-X 或选择 File 菜单中的 Quit 选项, 可退出 Turbo C 环境, 返回 DOS 提示符状态。

习 题 一

1.1 说明计算机内部信息和外部信息的主要差别。

1.2 简述 C 语言主要特点。

1.3 说明 C 语言源程序的一般构成。

1.4 用 C 语言编写程序, 在屏幕上显示如下图形。

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

1.5 说明使用 Turbo C 编程环境完成一个程序编辑及调试的过程。

第二章 数据与基本操作

变量、常量和函数等都是程序加工处理的对象，统称为数据。数据有各种类型，不同类型的数据在机器内存中占有不同的存储长度，具有不同类型的运算，在程序中必须说明数据的类型。本章主要介绍 C 语言的数据类型及标识符、常量与变量的表示方法，各种运算符的运算规则、取值范围和优先级，以及不同类型数据间的转换。

§ 2.1 C 语言的数据类型

C 语言的数据类型通常分为三类，如表 2-1 所示。本节只介绍基本类型，其他类型在后续章节中陆续介绍。

2.1.1 数值型数据的表示及存储

一、整数类型

C 语言中，整数类型数据有不同的长度，分别为短整型 short、普通整型 int、长整型 long。整型数据包括正、负整数和零，一般表示可数的个数、次序等。在数学中，整数是一个无限的集合，即 $-\infty \sim +\infty$ 的范围，而在计算机中

类型名	所占字节数	数 值 范 围
short	2	-32768 ~ +32767
int	2	-32768 ~ +32767
long	4	-2147483648 ~ +2147483647

基本类型	构造类型	派生类型
整数类型	数组类型	指针类型
实数类型	结构类型	无类型
字符类型	联合类型	
	枚举类型	

整数的取值范围受限于机器所能表示的范围。表 2-2 列出了计算机中（以 IBM-PC 为例）对以上各种数据所分配的存储空间（以字节为单位）及数值范围。

【例 2-1】求 653541698 与 1414892245 之和。

```
main()
{
    long a, b, sum;          /*变量说明，说明 a, b, sum 为长整型变量*/
    a=653541698;
    b=1414892245;
    sum=a+b;
    printf("%ld+%ld=%ld\n", a, b, sum);    /*长整型显示格式*/
}
```

运行结果：

653541698+1414892245=2068433943

可见长整型数据的范围较大。

二、实数类型

C 语言中的实数类型数据分为单精度类型(标识符为 float)和双精度类型(标识符为 double),在机器内部,C 语言中的实型数是以浮点数形式存储的,数值都是近似的,而且误差累积很快,为此,引进双精度类型,即以两倍单精度的存储空间存放数据。它可以减小累积的截断误差,大大改善计算的精度。实型数通常有两种表示形式:十进制形式和指数形式。注意在表示实型数时,字母 E 可以用 e 代替,指数部分必须是整数(若为正整数时,可以省略“+”号)。表 2-3 列出了 C 语言中实型数据的长度和数值范围。

表 2-3 实数数值范围

类 型 名	所占字节数	数 值 范 围
float	4	约 6 位有效数字(3.4E-38 ~ 3.4E+38)
double	8	约 12 位有效数字(1.7E-308 ~ 1.7E+308)
long double	16	约 24 位有效数字

2.1.2 字符型数据的表示及存储

C 语言的字符类型数据代表 ASCII 字符集里的一个字符,在程序中用单引号括起来,在内存中占一个字节长度,取值范围是-128 ~ +127。C 语言中还有一种特殊形式的字符类型数据是以一个“\”开头的字符序列,但它们只代表某个特定的具有控制功能的 ASCII 字符,在程序中使用时要括在一对单引号内,称为转义字符,见表 2-4。

表 2-4 转义字符功能

字符形式	功 能	字符形式	功 能
\n	换行	\\	反斜杠字符“\”
\t	横向跳格	\'	单引号字符
\v	竖向跳格	\"	双引号字符
\b	退格	\ddd	1 到 3 位 8 进制数所代表的字符
\r	回车	\xhh	1 到 2 位 16 进制数所代表的字符
\f	走纸换页		

【例 2-2】 用转义字符方法输出指定的符号。

```
main()
{ char ch;
  ch= '\361' ;
  printf("%c\n",ch);
}
```

运行结果:

±

上例中转义字符“\361”对应的 ASCII 字符是“±”(241)。可见利用转义字符可以将 ASCII 中有的而键盘上没有的字符打印出来。

2.1.3 无符号数据类型

前面介绍的几种数据类型,除字符型外都是指带符号的,C语言中也允许使用不带符号的数据,这时二进制数的最左边一位不再作为符号位,而是与右边各位一起用来表示数值,使得表示的数最大值扩大了一倍。例如:

unsigned int	取值范围: 0 ~ 65535
unsigned short	取值范围: 0 ~ 65535
unsigned long	取值范围: 0 ~ 4294967295
unsigned char	取值范围: 0 ~ 255

【例 2-3】 两数的求和。

```
main()
{ int a,b;
  unsigned int c;
  a=30000;
  b=30000;
  c=a+b;
  printf("ld\n",c);
}
```

运行结果:

60000

上例对变量 c 进行 unsigned int 型说明,否则不能正确存储整数 60000。

§ 2.2 C 语言的标识符、常量与变量

2.2.1 标识符

标识符是一个字符序列,是给程序中的实体——变量、常量、函数、数组结构以及文件所起的名字。标识符可分为三类:

一、保留关键字

C 语言共有 32 个保留关键字,每个关键字在 C 程序中都代表着某个固定含义,所有保留关键字都要用小写英文字母表示,且这些关键字都不允许在程序中作为用户标识符使用。

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

二、预定义标识符

预定义标识符在 C 语言中都具有特定的含义,如 C 语言提供的库函数的名字和预编译处理命令。C 语言语法允许用户把这类标识符作其他用途,但这将使这些预定义标识符失去系统规定的原意。鉴于目前各种计算机系统的 C 语言已经把这类标识符作为统一的库函数名或