

第 2 章 数据类型、运算符和表达式

C 语言程序设计

第 2 章

清华大学出版社

THINKING PRESS



教育部高职高专规划教材

C 语言程序设计

廖雷 主编

高等教育出版社

内容提要

本书是教育部高职高专规划教材,依据教育部《高职高专教育结构化程序设计课程教学基本要求》编写。在全面介绍 ANSI C 的语言成分和标准库函数的同时,对 Turbo C 2.0 集成环境的使用、Turbo C 特有的 BIOS 和 DOS 功能调用函数、屏幕处理函数、图形处理函数等常用库函数作了讲解,并将同时具备了 Visual Basic 的易用性和 C/C++ 的高效性的 Windows 下的软件开发工具 Borland C++ Builder 4.0 推荐给读者。在介绍上述内容的过程中,穿插讲解了相应的程序设计技巧、常用算法和具有实用价值的程序实例。

编者力求体现概念准确、编排合理、循序渐进、深入浅出、讲解通俗、便于自学的特色,读者可以不具备其他高级语言和程序设计的基础知识。本书注重技术应用性,语言与程序设计并重,经典实例和实用程序并重;强化实践环节,有专门章节介绍上机步骤、调试技巧,精选了较多的习题和实习题;反映最新技术进展,介绍了大众化软件开发工具 Borland C++ Builder。

本书可作为高等职业学校、高等专科学校、成人高等学校及本科院校举办的二级职业技术学院工科各专业的教材,也可作为计算机培训和等级考试辅导的教学用书,还可供程序开发人员和自学者参考。

图书在版编目(CIP)数据

C 语言程序设计/廖雷主编. —北京:高等教育出版社,2000
教育部高职高专规划教材

ISBN 7-04-008713-8

I. C... II. 廖... III. C 语言-程序设计-高等学校:
技术学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2000)第 26438 号

C 语言程序设计

廖 雷 主编

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

电 话 010-64054588

网 址 <http://www.hep.edu.cn>

邮政编码 100009

传 真 010-64014048

经 销 新华书店北京发行所

印 刷 国防工业出版社印刷厂

开 本 787×1092 1/16

印 张 17.25

字 数 410 000

版 次 2000 年 7 月第 1 版

印 次 2000 年 7 月第 1 次印刷

定 价 14.90 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

前 言

近十年来, 计算机技术得到飞速发展, 计算机已在各行各业得到了广泛应用, 对计算机应用人员 (特别是那些非计算机专业出身的技术人员) 的技术要求, 已不再停留在操作员的层次, 而是随之提出了更高的要求, 即要求他们能掌握一定的程序设计技术, 以解决更为复杂和专业的问题。

掌握程序设计的前提是掌握程序设计语言, 在众多的程序设计语言中, C 语言以其灵活性和实用性受到了广大计算机应用人员的喜爱。同时 C 语言是既得到美国国家标准学会 (ANSI) 标准化, 又得到工业界广泛支持的计算机语言之一, 几乎任何一种机型 (大型机、小型机、工作站、PC 机)、任何一种操作系统 (Windows, Unix, Linux, Netware) 都支持 C 语言开发。C 语言在巩固其原有应用领域的同时, 又在拓展新的应用领域, 支持大型数据库开发和 internet 应用, 并且 C 语言本身也拓展了面向对象的语言成分, 发展成为目前最有影响力的面向对象的程序设计语言 C++。C/C++ 语言几乎具备现代程序设计语言的所有语言成分, 一旦掌握了 C/C++ 语言, 就可以较为轻松地学习其他任何一种程序设计语言, 但反过来就未必如此。

本书的读者对象主要是高职高专的工科各专业的学生。根据高职高专的课程设置, C 语言被推荐为工科工科各专业程序设计语言的必修课程, 其他专业也可酌情选学。同时, 社会上众多计算机应用人员也对 C 语言有浓厚的兴趣。然而, 与其他程序设计语言相比, C 语言学习难度要大一些, 因此, 要学好 C 语言, 提供一本合适的教材是使读者能顺利地掌握和运用 C 语言一个重要因素。为此, 本书根据读者对象的性质, 力图体现以下编写特色:

1. 起点较低, 不需具备程序设计语言基础知识。

很多 C 语言的教材都要求读者先前学过一门程序设计语言。但我们认为, 本书的主要读者是高职高专工科各专业学生, 他们学制只有三年, 要求他们先学一门其他语言, 总课时就占的较多, 学生学业负担就较重。因此, 我们从程序设计的最基础知识讲起, 把一些经典算法的来龙去脉交代清楚, 读者就不一定非要有其他程序设计语言的基础。

2. 概念准确, 编排合理。

由于历史的原因, 与其他程序设计语言相比, C 语言显得不是很严谨, 因此, 在理解 C 语言时容易产生偏差。为此, 我们认真研读、消化了最具权威的由 C 语言设计者 K&R 按照 ANSI C 标准所著的 “The C Programming Language” (第二版) 一书以及微软和 Inprise/Borland 公司最新的 C/C++ 语言产品的联机手册, 以此为基础讲解 C 语言。同时, 在内容的编排上, 注意分散难点, 便于读者循序渐进。

3. 详略得当, 重点突出。

为体现高职高专教学中 “必需、够用为度” 的原则, 本书主要讲解 C 语言最基本、最常用的内容, 有意回避一些 C 语言中出现频率很低或与语言的实现版本有关的内容。把重点放在语言本身的难点 (如指针) 和程序设计技巧方面。

4. 深入浅出, 讲解通俗。

根据高职高专教学和社会读者自学的特点, 避免一般本科教材中引出概念, 解释概念, 举例说明的传统讲解模式, 而采用根据知识结构要求, 提出问题, 分析问题, 解决问题, 最后总结出概念并推广到一般的写作方法。同时, 在讲解中强调通俗性。

5. 两个并重, 强化实践, 重视应用。

所谓两个并重, 即程序设计语言和程序设计技巧并重, 经典实例和实用程序并重。本书力求使读者学完 C 语言后, 不仅能懂 C 语言的语法、语义, 更重要的是具备编程解决实际问题的能力。读者不仅要了解一些经典实例和经典算法, 还应研究一些来自实际工作和工程实践的实用案例, 加以借鉴、模仿、改写。同时, 本书重视实践环节, 有专门章节介绍上机编程和程序调试技巧, 附录中有中英文出错信息对照表, 并提供了较多的习题和上机实习题。

6. 反映最新技术进展。

当前的程序设计技术, 已从结构化程序设计技术向面向对象程序设计技术过渡。具体地说, 对一个规模较大的应用程序, 总体框架是由面向对象程序设计构搭而成, 而在局部实现时仍需采用结构化程序设计技术。因此, 我们在本书最后一章从实用的角度简要介绍了面向对象的程序设计语言 C++。同时我们认为, 面向对象的程序设计可分为两个层次, 第一层次是利用一些开发工具提供的类体系(如微软 Visual C++ 中的 MFC, Inprise/Borland C++ Builder 中的 VCL)进行开发, 第二层次是根据实际应用, 自己构造一个类体系。我们认为, 就目前技术发展水平, 高职高专工科各专业的学生只需具备第一层次的技能。

自从 Windows 出现以来, 用 C 语言开发程序的技术人员比例一度比 DOS 时代低了, 究其原因, 无论微软的 Visual C++ 还是 Inprise/Borland 的 Borland C++ 的学习难度都较大, 因此一部分技术人员改用诸如 Visual Basic, Delphi, PowerBuilder 等快速应用程序开发工具(RAD)进行开发。而如今局面终于开始改观, Inprise/Borland 推出的 C++ Builder 是一个全新的真正可视化编程工具, 它采纳了微软公司 Windows GUI 的许多先进特性和设计思想, 采用了标准化的面向对象程序设计语言 C++, 可以说, C++ Builder 既有 Visual Basic 的易用性, 又有 C/C++ 的高效性, 将 Windows 下的 C/C++ 程序设计带入了大众化时代。因此, 本书在最后一章中简要介绍了 C++ Builder 4.0。

本书由廖雷主编, 袁璟, 陈立参编。其中主编负责拟定全书框架并对参编所写章节仔细阅读, 提出了明确和具体的修改意见。各章具体分工如下: 廖雷编写第 1 章、第 2 章、第 6 章、第 8 章、第 9 章、第 10 章, 附录 2、附录 3, 袁璟编写第 3 章、第 4 章、第 5 章、第 9 章图形处理部分内容和附录 1, 陈立编写第 7 章、第 11 章、第 12 章。深圳职业技术学院的余苏宁副教授在百忙之中抽出时间认真细致地审阅了全书, 并提出了宝贵意见, 作者深表感谢, 同时对余教授在审稿过程中体现出的优秀的业务素质和严谨的工作作风表示由衷的钦佩。

本书得以出版, 与教育部高教司高职高专处、高等教育出版社的指导和支持, 作者所在学校及教务处、系、教研室领导和同事的关心和帮助是分不开的。同时, 作者在与北京联合大学、深圳职业技术学院、上海第二工业大学、南京动力高等专科学校、承德石油高等专科学校的有关专家教授的交流中获益非浅, 在此一并致谢。

虽然作者从 1988 年起即用 C 语言进行科研和科技开发工作, 又有 10 年的 C 语言教学经验,

但由于水平有限，加之时间仓促，错漏之处恐在所难免，热诚欢迎广大读者把你们的意见、建议和要求反馈给我们，以便作进一步的完善，作者的电子信箱是：liaolei@sina.com。

作 者

2000年3月于南京

第 1 章 C 语言概述

1.1 为什么要学习 C 语言

1.1.1 C 语言的历史

Fortran 语言是历史上的第一门计算机高级语言,它主要用于科学计算,随着 Fortran 的出现,越来越多的计算机专家和工程技术人员对高级语言的研究、设计和使用产生了浓厚的兴趣。诞生于 20 世纪 60 年代的 Algol 语言是一门结构良好、逻辑严谨、简单易学的算法语言,但由于它的应用面较窄和人们对 Fortran 的依恋,结果没有能得到推广。20 世纪 70 年代初期出现的 PASCAL 语言是第一门反映了结构化程序设计思想的高级语言,它在大学和研究所中流传较广,一度成为国内外计算机(应用)专业学生的入门语言。

几乎与 PASCAL 语言诞生的同时,C 语言在美国著名的贝尔实验室中酝酿并诞生了。与 Fortran、Algol 和 PASCAL 语言不同,C 语言诞生之时并没有什么研制报告和语言报告,而是在设计 Unix 操作系统时不断地得到更新和完善。因此,人们把 C 语言称为程序员设计的语言,而把 Fortran、Algol 和 PASCAL 语言称为计算机科学家设计的语言。

Unix 的早期版本是用汇编语言写的,而用 C 编写的 Unix 比原先的版本更易于理解、修改和扩充,更重要的是,具有良好的可移植性。作为一个优秀的操作系统,Unix 在世界范围内得到了广泛的应用,它的设计者因此于 1983 年获得了计算机科学的最高奖——图灵奖。要使用 Unix,就必须掌握 C 语言,因此,C 语言也为越来越多的人所熟知,人们进一步认识到 C 语言是一门极有生命力的程序设计语言。渐渐地 C 语言已不完全依赖 Unix,成为程序员的首选语言之一。在国内,很多高校将 C 语言作为第一门程序设计语言开设。

1978 年,贝尔实验室的 B.W.Kernighan 和 D.M.Ritchie(简称 K&R)出版了“The C Programming Language”一书,建立了所谓的 C 语言的 K&R 标准,它一度成为 C 语言的事实标准。目前,美国国家标准学会已对 C 语言标准化,称为 ANSI C 标准。ANSI C 标准与 K&R 标准之间有一些差别,本书在讲解 C 语言时以 ANSI C 标准为主,再加上 Borland 公司 Turbo C 2.0 的一些扩充。

C 语言本身也在发展,20 世纪 80 年代中期,出现了面向对象程序设计的概念。贝尔实验室的 B. Stroustrup 博士借鉴了 Simula 67 中的类的概念,将面向对象的语言成分引入到 C 语言中,设计出了 C++ 语言。C++ 语言赢得了广大程序员喜爱,不同的机器不同的操作系统几乎都支持 C++ 语言,如对 PC 机,微软公司先后推出了 MS C++、Visual C++ 等产品,Borland 公司先后推出了 Turbo C++、Borland C++、C++ Builder 等产品。同时,C++ 语言也得到了国际标准化组织(ISO)的认可,为此,国际标准化组织(ISO)已对 C++ 语言实现标准化。本书的最后一章简要介绍了 C++ 语言及 Borland 公司的 C++ Builder 编程环境。

C/C++语言对新语言的形成也有较大的影响力。20 世纪 90 年代中期以来, Internet 日益普及, 用于 Internet 开发的 Java 语言日益为人们所熟知, 事实上, Java 语言与 C++语言极为相似, 熟悉 C++语言的程序员在很短的时间内就能掌握 Java。

1.1.2 C 语言的特点

C 语言之所以能成为程序员的首选语言之一, 是因为具有如下特点:

(1) 具有现代程序设计语言的特征

C 语言具有丰富的数据类型, 众多的运算符, 体现结构化程序设计的优良的控制结构, 以及具备抽象功能和体现信息隐蔽思想的函数。

(2) 用途广泛

C 语言的应用几乎遍及了程序设计的各个领域, 如科学计算、系统程序设计、字处理软件和电子表格软件的开发、信息管理、计算机辅助设计、图形图像处理、数据采集、实时控制、嵌入式系统开发、网络通讯、Internet 应用、人工智能等。

(3) 语言简洁, 具备底层处理功能、可执行代码质量高

C 语言简洁, 为完成某一功能所写的源程序往往比用其他语言来得短, 使得程序输入工作量减少。C 语言能直接访问物理地址和端口, 并能进行位操作, 因此能实现汇编语言的大部分功能。另一方面, 由 C 语言生成的可执行代码内存容量少, 执行效率高, 因此, C 语言有可移植的汇编语言的美称。

(4) 可移植性好

若程序员在编写程序时严格遵循 ANSI C 标准, 则其源代码可不作修改, 即可用于各种型号的计算机和各种操作系统, 因此, C 语言具备良好的可移植性。

1.2 C 语言的一个简单实例

下面是 C 语言的一个简单实例, 其功能是从键盘上读入两个整数, 并计算两个数和 6 三者之和, 最后将结果输出。

【例 1.1】C 语言的一个简单实例。

```
#include <stdio.h>

main()
{ int a,b,sum;
  printf("Enter Two Numbers:");
  scanf("%d%d",&a,&b);
  sum=a+b+6;
  printf("The sum is %d\n",sum);
}
```

程序运行结果如下:

```
Enter Two Numbers:64 28
The sum is 98
```


一个 C 语言程序的执行总是从被称为 main 的主函数处开始, 在【例 1.1】中, main 函数对变量 a、b、sum 作了说明, 它们的类型是整型, printf 是一个标准输出函数, 因此, main 函数中的第一个 printf 函数输出一行提示信息: Enter Two Numbers, 要求用户输入两个整数, scanf 是一个标准输入函数, 它完成 a、b 两个变量的输入工作, 即从键盘上输入两个数, 使得 a、b 分别取值 64 和 28, 语句 sum=a+b+6 计算 a+b+6 的值并将它送给 sum 变量, 第二个 printf 函数调用完成 sum 的打印, 即将文字 The sum is 和运算结果 98 一起输出。

举这个例子, 主要是为了使读者对 C 语言有一个大致的了解, 具体细节不必完全弄懂。

1.3 编辑、编译、连接、运行一个 C 语言程序

我们用 C 语言书写的程序又称为 C 的源程序, 它是不能直接运行的。为此, 必须生成与之对应的可执行程序。具体过程如下:

(1) 编辑源程序, 完成后将源程序以扩展名.c 存盘。

(2) 对源程序进行编译, 即将源程序转换为扩展名为.obj 的二进制代码, 此二进制代码仍不能运行。若源程序有错, 必须予以修改, 然后重新编译。

(3) 对编译通过的源程序连接, 即加入库函数和其他二进制代码生成可执行程序。连接过程中, 可能出现未定义的函数等错误, 为此, 必须修改源程序, 重新编译和连接。

(4) 执行生成的可执行代码, 若不能得到正确的结果, 必须修改源程序, 重新编译和连接; 若能得到正确结果, 则整个编辑、编译、连接、运行过程顺利结束。

在 Turbo C 2.0 中, 提供了一个集成开发环境, 上述过程步骤均能在 Turbo C 2.0 集成环境中完成。有关集成环境的具体使用详见后继章节。

1.4 学习 C 语言所需的必备知识

学习 C 语言, 除学习理论知识外, 还必须通过上机编制和调试程序。事实上, 不同的机器在不同的操作系统下有不同的 C 语言编译器, 本书主要用 Borland 公司的 Turbo C 2.0 讲解 C 语言的实际编程。Turbo C 2.0 是 DOS 下的一个集成开发环境, 在 Windows 下需进入 DOS 窗口运行, 因此, 熟悉常用 DOS 命令或 Windows 基本操作是学习本课程的前提条件。

学习 C 语言, 还必须弄清数据在内存中的表示方法。数据在计算机内部是以二进制来表示, 而为了便于问题描述, 我们还常常用到十六进制和八进制, 实际上, 它们都可以认为是二进制的缩写方式。

1.4.1 数制

1. 十进制数

十进制是我们日常使用的进制, 由 0~9 十个数字组成, 运算规则是逢十进一。

2. 二进制数

二进制数在日常生活中也会出现, 如两只鞋子称为一双, 这就是二进制。二进制的基数是 2, 因此, 在二进制中出现的数字只有二个: 0 和 1。

二进制的规则是逢二进一。因此有：

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10$$

$$0 \times 0=0 \quad 0 \times 1=0 \quad 1 \times 0=0 \quad 1 \times 1=1$$

3. 八进制数

八进制的基数是 8，所使用的数字为 0、1、2、3、4、5、6、7。其运算规则是逢八进一。

4. 十六进制数

十六进制的基数是 16，它使用的数字字符为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。其中 A~F 分别代表十进制数的 10~15。其运算规则是逢十六进一。

5. 数制标记方法

为了区分不同进制的数据，我们可以用圆括号将数据括起来，在括号的右下角以数字 2、8、10、16 表示该数代表的进制。如

$(100)_2$, $(261)_8$, $(192)_{10}$, $(1FA)_{16}$ 分别代表的是二进制的 100、八进制的 261、十进制的 192、十六进制的 1FA。

1.4.2 数制之间的转换

下面通过一些实例来说明不同进制数相互转换的方法。

1. 二进制数转换成十进制数

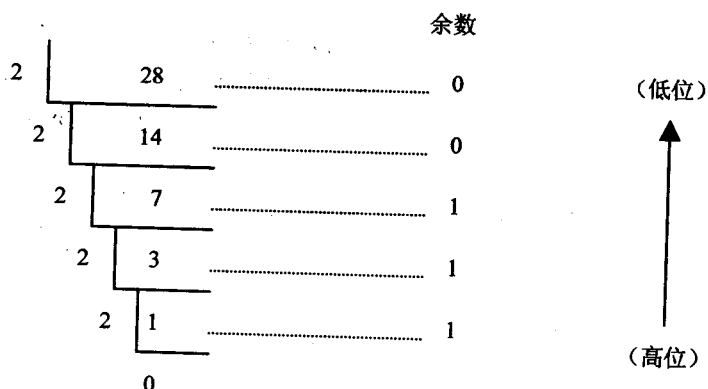
$$(111.011)_2 = (1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3})_{10} = (7.375)_{10}$$

2. 十六进制数转换成十进制数

$$(2AB.C)_{16} = (2 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 + 12 \times 16^{-1})_{10} = (683.75)_{10}$$

3. 十进制整数转换成二进制数

十进制整数转换成二进制数可采用“除 2 取余法”，直至商为 0。如将十进制数 28 转换成二进制数：



注意，最后得到的余数是高位数，因此

$$(28)_{10} = (11100)_2$$

由以上实例可知，我们可以实现任何数制之间的转换，例如要将二进制数转换成十六进制数，可先将二进制数转换成十进制数，再将十进制数用“除 16 取余法”得到十六进制数。

事实上,二进制、八进制、十六进制数之间的相互转换有更简便的方法,因限于篇幅没有展开,读者可自行查阅有关书籍。

1.4.3 整数的原码、补码、反码表示

一般计算机中用 16 位或 32 位来表示整数,位数越多,能表示数的范围就越大。

我们知道,整数有正负之分,为此,我们可以用一个二进位作为符号位,一般总是最高位,当符号位为“0”时表示正数,符号位为“1”时表示负数。例如,当我们用 16 位来表示一个整数时,有

$0000000000101011=+43$

$1000000000101011=-43$

上述表示法,称为整数的原码表示法。

整数也可采用反码表示法,对于负整数来说,符号位作为“1”,但绝对值部分正好与原码相反(即 0 变为 1,1 变为 0)。因此:

$(-43)_{\text{原}}=1000000000101011$

$(-43)_{\text{反}}=1111111111010100$

而实际上,整数在机器内大多用补码表示,对负整数而言,符号位仍为 1,但绝对值部分却是反码的最低位加 1 得到的结果,因此

$(-43)_{\text{补}}=1111111111010101$

注意,对正整数而言,其原码、反码、补码均相同。

关于补码的知识,是讲述 C 语言的必备知识。至于实数在计算机内的表示法,因限于篇幅,不再讲述。

习 题

1. 简述 C 语言有哪些特点。
2. 简述编辑、编译、连接、运行一个 C 语言程序的步骤。
3. Inprise (Borland) 公司和微软公司的 C/C++ 语言产品是 PC 机上最有影响力的产品,试问这二家公司先后推出了哪些 C/C++ 语言产品?

4. 写出整数 8 与 -8 在机内的补码表示。

5. 有二个整数,它们在机内的表示分别

111111111110100

000000000001010

试问这是哪二个整数的补码表示?

第2章 数据类型、运算符、表达式、赋值语句、输入输出

2.1 C 语言的词法单位

C 语言是一种程序设计语言，由 C 语言编写的程序是由各种不同的词法单位构成的。所谓词法单位，是指程序中具有独立含义的不可进一步分割的单位，具体地说，C 语言的词法单位可分成 6 类：

- (1) 关键词
- (2) 标识符
- (3) 常量
- (4) 字符串
- (5) 运算符
- (6) 分割符

以下面程序为例。

【例 2.1】C 语言的一个简单例子。

```
#include <stdio.h>

main()
{ int a,b,sum;
  printf("Enter two numbers:");
  scanf("%d%d",&a,&b);
  sum=a+b+6;
  printf("The sum is %d\n",sum);
}
```

该程序由不同的词法单位组成，其中：

int 是关键词，sum 是标识符，6 是常量，"Enter two numbers"是字符串，+ 是运算符，(是分割符。

下面先介绍关键词、标识符和分割符的含义。

关键词是 C 语言中预定义的符号，它们有固定的含义，用户定义的任何名字不得与它们冲突。现将所有的关键词罗列如下：

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

其具体含义将在后继章节陆续介绍。

除关键词外，还有准关键词，即它们也有固定的含义，主要作为库函数名和预处理命令。

C 语言允许这些符号另作它用，使这些符号失去原先含义。但为避免出现不必要的麻烦，建议不对这些准关键词另作它用。

出现在预处理命令中的准关键词有：

`define` `endif` `include` `ifdef` `ifndef` `line` `undef`

其余准关键词均为系统函数名，如 `scanf`、`printf` 等。

标识符是用户自行定义的符号，用来标识常量、变量、函数、标号等。C 语言要求，所有符号必须先定义/说明而后使用。ANSI C 规定，标识符是由字母或下划线(`_`)开头，后跟零个或多个字母、下划线和数字中任何一个符号所组成的符号序列，因此，下列词法单位都是合法的标识符：

`i` `count` `num_day` `round2` `_ax`

而下列标识符是不合法的：

`586pc` `b-c` `#abc`

Turbo C 中，标识符最长可达 32 个字符。

注意，C 语言是区分大小写的，因此，`int` 是关键词，而 `Int` 则是标识符。原则上，我们用大写字母来表示常量标识符，而变量等其他类型标识符则用小写字母表示。

在给标识符起名时，最好选择相应的英文单词或汉语拼音或它们的缩写，这样可以增加程序的可读性。

常用的分隔符有：

[] () { } * . : = ; #

至于常量、字符串、运算符，将在后面陆续介绍。

2.2 数据类型

C 语言中，为解决具体问题，要采用各种类型的数据，数据的类型不同，它所表达的数据范围、精度和所占据的存储空间均不相同，本章首先讨论 C 语言的三种基本数据类型：整型、浮点型、字符型。

2.2.1 整型

整型可分为四种：基本型、短整型、长整型和无符号型。

1. 基本型：以 `int` 表示。
2. 短整型：以 `short int` 或 `short` 表示。
3. 长整型：以 `long int` 或 `long` 表示。
4. 无符号型：无符号型的整数必须是正数或零。无符号型又可细分为：无符号整型、无符号短整型和无符号长整型，分别以 `unsigned int`、`unsigned short` 和 `unsigned long` 表示。

以上各种类型数据所占的内存因机器而异。对 PC 机，各种类型数据所占位数和数的范围如表 2-1。

表 2-1 各种整型数据所占位数及数的范围

数据类型	所占位数	数的范围
int	16	-32 768 ~ 32 767
short	16	-32 768 ~ 32 767
long	32	-2 147 483 648 ~ 2 147 483 647
unsigned	16	0 ~ 65 535
unsigned short	16	0 ~ 65 535
unsigned long	32	0 ~ 4 294 967 295

无符号数在内存中存放时二进制最高位不是符号位，而有符号整数在内存中以补码表示，其最高位是 1 时表示负数，最高位是 0 时表示非负数。

2.2.2 浮点型

浮点型用来表示实型数据。浮点型分为两种：单精度和双精度，分别以 float 和 double 表示。对 PC 机，浮点型数的有关参数见表 2-2。

表 2-2 各种实型数据所占位数及数的范围

数据类型	有效数字	所占位数	数的绝对值范围
float	7	32	$10^{-38} \sim 10^{38}$
double	15	64	$10^{-308} \sim 10^{308}$

2.2.3 字符型

字符型用 char 表示，一般用 8 位来存放一个字符，实际上存放的是该 ASCII 码值（即一个整数），因此，字符型和整型关系非常密切，可把字符型看作一种特殊的整型。事实上，字符型数据和整型数据之间经常混合使用。

2.3 常量和变量

2.3.1 常量

常量是指在程序运行过程中，其值不能被改变的量。常量包括整型常量、浮点型常量、字符型常量和符号常量。

整数常量即整常数，由一个或多个数字组成，可以带正负号。其中以数字 0 开头的为八进制数，以 0x 或 0X 开头的为十六进制数，否则就是十进制数。

下列常量都是合法常量：

32767 -314 0 0516 017 0x1c0d 0X4600

在整型常数后跟有 l 或 L 时，表示该整型常量是长整型常量。如：

49876L 0X4F8DAL

浮点常量的表示有十进制形式和指数形式两种，例如下列常量都是合法的浮点常量：

3.14 -9.568 1.5E5 2.9E-7

其中，后两个数采用了指数形式，分别表示 1.5×10^5 和 2.9×10^{-7} 。注意，指数形式的浮点常量 E 或 e 前面必须有数字，E 后面必须为整数。因此，E8、6E7.8 都不是合法的浮点常量。

字符型常量是用单引号括起来的单一字符，如 'S'、's'，它们分别代表字母 s 的大小写。一个字符常量的值就是该字符的 ASCII 码值，如 '2' 的值就是 50。

C 还允许一种特殊形式的字符常量，就是以 \ 开头的字符序列。如 '\n' 表示一个换行符。常用的以 \ 开头的特殊字符见表 2-3。

表 2-3 转义字符表

字符形式	含 义
\n	换行
\t	横向跳格 (Tab)
\v	竖向跳格
\b	退格
\r	回车
\f	换行但不回车
\\	反斜杠 \
\'	单引号
\xhh	1 到 2 位 16 进制数所代表的字符
\ddd	1 到 3 位 8 进制数所代表的字符

由上表可知，\030 代表 ASCII 码为 8 进制数 30 的字符，即 $\uparrow$ ，而 \x16 代表 ASCII 码值为 16 进制 16 的字符，即 $\leftarrow$ 。

字符串常量是双引号括起来的零个、一个或多个字符序列。如 "I am a student"。编译程序自动地在每一个字符串末尾添加串结束符 \0，因此，所需要的存储空间比字符串的字符个数多一个字节，故上述字符在内存中如下存放：

I		a	m		a		s	t	u	d	e	n	t	\0
---	--	---	---	--	---	--	---	---	---	---	---	---	---	----

必须注意，字符常量和单字符的字符串是不一样的，如 'x' 与 "x" 是两个不同类型的常量。常量还有另一种表示方法，称为符号常量，后继章节将予以介绍。

2.3.2 变量

顾名思义，其值可以改变的量称为变量。一个变量由一个名字来标识，此名字称为该变量的标识符（应多以有意义的名字标识变量，以使程序更加明了）。变量在内存中占据一定的存储单元，该存储单元中存放变量的值。注意，变量名和变量值是两个不同的概念，变量名与内存中的某一存储单元相联系，而变量值是指存放在该存储单元中的数据的值。这样，同一个变量名对应的变量在不同的时刻可以有不同的值。

变量在使用之前必须加以说明，变量说明的一般形式是：

<类型标识符> <变量名> { , <变量名> } ;

其中用{ }括起来的内容可以重复零次或多次。

变量说明的一些实例如下：

```
int i;                /* 说明 i 为整型变量*/
short c,d,e;          /*说明 c,d,e 为短整型*/
unsigned long p,g;     /* 说明 p,g 为无符号长整型*/
float x,y;             /* 说明 x,y 为单精度型*/
double z;             /* 说明 z 为双精度型*/
char ch1,ch2;         /* 说明 ch1,ch2 为字符型*/
```

在 C 语言中，夹在/* */中的内容表示注解（程序运行时并不执行，但必要的注解可提高程序的可读性，应充分予以重视）。

上述变量被说明后，根据其类型的不同，拥有不同大小的存储单元。

2.4 运算符和表达式

2.4.1 算术运算符和算术表达式

算术运算符包括+（加）、-（减）、*（乘）、/（除）、%（取模）、++（增一运算符）以及--（减一运算符）。

请注意乘、除运算符在 C 语言中的表达方式与数学中的区别。对于/运算符，若除数和被除数均为整数，则只取整数部分，舍弃小数部分，如 $6/4=1$ ；而若除数或被除数中只要一个为实数，则结果是 double 型，如 $6/4.0=1.5$ 。

取模运算符 %，又称求余运算符，它求出运算符 % 两侧的两个整型数据整除后的余数，如 $6\%4=2$ ，因为 6 除以 4 的余数 2。

下面是由算术运算符构成的算术表达式：

```
2*3.14*r
(5+6)/2.0-u*v+p%q
```

增一运算符 ++ 和减一运算符--是两个独特的单目运算符，其独特之处在于它们既可以用作变量的前缀，又可以作为变量的后缀。这两种用法的效果都是使变量增一或减一，但用作前缀时，++ 和--表示先增加和先减少再取其值，而用作后缀时，++ 和--表示先取其值再增加和再减少，如：

++i, --i（在使用 i 之前，先使 i 的值增/减 1）

i++, i--（在使用 i 之后，使 i 的值增/减 1）

粗看起来，++i 和 i++ 的作用都相当于 $i=i+1$ ，但细细研究一下，++i 是先执行 $i=i+1$ ，再使用 i 的值，而 i++ 是先使用的 i 值后，再执行 $i=i+1$ 。若 i 的原先值为 8，则

j=++i 先使 i 的值加 1，变为 9，然后再送给 j，因此，j 的值为 9

j=i++ 先将 i 的值送给 j，这样，j 的值为 8，然后 i 值再加 1，变为 9。

必须说明的是 ++、--运算符只能用于变量，而不能用于常量或表达式。

2.4.2 关系运算符和关系表达式

关系运算符有 > (大于)、>= (大于等于)、< (小于)、<= (小于等于)、== (等于) 和 != (不等于)，它们的优先级低于算术运算符的优先级。在由关系运算符构成的关系表达式中，若规定的关系成立，则表达式值为非零整数（系统实现时往往为 1）；若不成立，则表达式值为零。可以看出，在 C 语言中用整型变量替代了逻辑量。需要指出的是，C 语言中的等于关系用 == 表示，而不是 = (赋值运算符)。下面是由关系运算符构成的关系表达式的几个实例：

```
i>5
ch<='z'
i==count
time!=p+q
```

读者很可能把最后一个关系表达式写成 `time!=(p+q)`，但因为 C 语言中，+ 运算符的优先级高于 != 运算符，故圆括号是多余的，这也从一个方面反映了 C 语言简洁的特点。

2.4.3 逻辑运算符和逻辑表达式

逻辑运算符有 && (逻辑与)、|| (逻辑或) 和 ! (逻辑非) 三种。下面是由逻辑运算符构成的逻辑表达式：

```
a>b && c<d
x==1 || !b
!(sex=='M' && age<20)
```

对于 &&，若其左右两个操作数均为非零（真），返回值才为非零（真），否则返回值为零（假）。对于 ||，只要它左右两边的操作数有一个为非零（真），返回值就为非零（真），否则返回零。对于 !，它把非零操作数转换为零，而把零操作数转换为非零。就运算符优先级而言，逻辑运算符!的优先级高于算术运算符，而 && 和 || 的优先级低于关系运算符。

C 语言中，&& 和 || 是短路运算符。即在一个或多个 && 相连的表达式中，只要有一个操作数为零，就不做后面的 && 运算，整个表达式的结果为零。如对于表达式 <条件 1> && <条件 2>，若 <条件 1> 取值为零，则不需再判定 <条件 2>，立即可判定整个表达式的值为零。而由一个或多个 || 连接而成的表达式中，只要碰到第一个不为零的操作数，就不再继续进行后继运算，整个表达式的值不为零。

2.4.4 赋值运算符和赋值表达式

最基本的赋值运算符是 =，由赋值运算符组成的表达式称为赋值表达式。其一般形式为：

<变量> <赋值运算符> <表达式>

其含义是先求出 <表达式> 的值，然后将此值送入 <变量> 对应的存储单元，整个赋值表达式的值就是 <变量> 的值。如：

```
t=x+3
```