

microsoft®

windows®xp

C 语言程序设计

C YUYAN CHENGXU SHEJI

张明林 主编

西北工业大学出版社

C 语言程序设计

主 编 张明林

西北工业大学出版社

【内容简介】 C语言是目前广泛应用于软件开发的一种编译型程序设计语言。自从它问世以来,成功地开发出 Windows 操作系统,被越来越多的软件工作者所喜爱,是目前国内外广泛使用的计算机语言之一。

全书共分为 10 章,第 1 章介绍程序设计语言的发展过程,C 语言和其他高级语言的共同特点及 C 语言程序结构特点。第 2 章为基本数据类型(整型、实型、字符型、字符串型)及运算符和表达式。第 3 章介绍与流程控制的三种基本程序结构(顺序、选择、循环)相关的语句和程序设计方法。第 4 章内容是数组的使用及程序设计方法。第 5 章阐述了指针的使用及程序设计方法。第 6 章介绍函数的定义、调用及程序设计方法。第 7 章为宏、文件包含、条件编译预处理方法。第 8 章介绍结构体、共用体、枚举型数据的特点、定义、使用及程序设计方法。第 9 章内容是文件处理及程序设计方法。第 10 章介绍位运算的概念。

与本书配套的教材有《C 语言程序设计上机实验指导与习题集及章末习题解答》。

本书可作为高等院校本科理工类专业的正式教材,也是一本可供自学的好教材。

图书在版编目(CIP)数据

C 语言程序设计/张明林主编. —西安:西北工业大学出版社,2005. 12

ISBN 7-5612-1752-8

I. C… II. 张… III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 016399 号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路 127 号 邮编:710072 电话:(029)88493844

网 址:www.nwpup.com

印刷者:陕西友盛印务有限责任公司印刷

开 本:787 mm×1 092 mm 1/16

印 张:16.375

字 数:396 千字

版 次:2005 年 12 月第 1 版 2005 年 12 月第 1 次印刷

定 价:22.00 元

前言

C 语言是目前广泛用于软件开发的一种编译型程序设计语言。和其他高级语言相比,C 语言具有丰富的数据类型和运算符,语句极其简单,源程序清单简洁清晰;具有汇编语言的功能,可以直接处理硬件系统和外围设备接口的控制;是一种结构化程序设计语言;为中大型软件设计中采用模块化程序设计方法提供了基础。自从它问世以来成功地开发出 Windows 操作系统,被越来越多的软件工作者所喜爱,目前是国内外广泛使用的计算机语言之一。它既具有高级语言的优点,又具有汇编语言与计算机硬件操作关系密切的许多特点。因此,C 语言是当前软件开发的强有利工具,同时还能兼顾科学计算功能。原来许多用汇编语言编写的软件,现在可以在 C 语言环境下编写了。C 语言是目前全国各高校理工专业必修课程之一。学好这门课不仅为后续的专业基础课和专业课学习奠定基础,同时 C 语言的基础也是以后学习其他科学计算语言,如 C++,VC++,Java 等语言的基础。

全国计算机等级考试、全国计算机应用技术证书考试(NIT)和全国各地组织的大学生计算机统一考试都将 C 语言列入了考试范围。

由于 C 语言涉及的概念比较复杂,规则繁多,使用灵活,容易出错,初学者迫切希望有一本易于入门、容易学习的 C 语言教材。针对初学者的特点,本书在内容组织上总结作者多年 C 语言程序设计教学经验,全面介绍了 C 语言的基本语法和程序设计方法。教材内容层次条理性强,定位准确,没有讲到的概念不在当前章节中出现,所有例题都在 Turbo C 2.0 环境下调试通过。概念清晰,实用性强,例题丰富,每章后都配有一定量的习题,深入浅出。内容循序渐进,体系合理,逻辑性强,文字流畅,通俗易懂,是学习 C 语言的理想教材。本书可作为高等院校本科理工类专业的正式教材,也是一本可供自学的好教材。

在学校学习程序设计的目的是进行程序设计的最基本训练,而不是要求学生立即编写大型程序,因此,学习程序设计课程时,应该把精

力放在这门课的基本语法规则以及典型的、最常见的数学中的基本算法程序设计,例如,从求简单的几何图形的面积、体积、周长开始,到找出一定规律的数(素数、完数、数据排序等)、用迭代法或二分法解出一元高次方程近似根、求定积分的面积(数值积分)等。通过这些简单程序设计使学生掌握编程思路 and 技巧。学习程序设计,一定要灵活运用基本概念,忌死记而不会用,要举一反三。同一道题可以考虑用不同算法形成程序清单,如同数学中的一题多解一样,任何一道题,由于算法不同,思考问题的思维方法不一样,设计出的程序也绝对不会是完全一样的,但运行结果(答案)应是一样的。

程序设计是一门实践性很强的课程,既要掌握概念,又要多动手编程,还要多上机调试程序。不要满足于能答出填空题和选择题,这只是检验一下你对这门语言的语法概念掌握得是否到位,而应当把重点放在编写程序和调试程序上,因为语言学习要落实到解决实际问题。

全书共分为 10 章,第 1 章介绍程序设计语言的发展过程,C 语言和其他高级语言的特点及 C 语言程序结构特点。第 2 章介绍基本数据类型(整型、实型、字符型、字符串型)及运算符和表达式。第 3 章介绍与流程控制的三种基本程序结构(顺序、选择、循环)相关的语句和程序设计方法。第 4 章介绍数组的使用及程序设计方法。第 5 章介绍指针的使用及程序设计方法。第 6 章介绍函数的定义、调用及程序设计方法。第 7 章介绍宏、包含文件、条件编译等编译预处理方法。第 8 章介绍结构型、共用型、枚举型数据的特点、定义、使用及程序设计方法。第 9 章介绍文件处理及程序设计方法。第 10 章介绍位运算的概念。

与本书配套的教材有《C 语言程序设计上机实验指导与习题集及章末习题解答》。

本书由张明林教授主编。各章编写分工如下:第 1,2 章由郭向东编写,第 3,4 章由张冬梅编写,第 5,6 章由张明林编写,第 7,8 章由周大明编写,第 9,10 章由张忠彬编写。全书由张明林定稿。

最后,对多年来关心支持本书和本书作者的西北工业大学明德学院教学部领导和所有教师表示衷心的感谢,同时,对本书在编写过程中所参考文献的作者致以诚挚的谢意。由于作者水平有限,书中肯定会有不少缺点和不足,热切期望得到专家和读者的批评指正。

编 者

2005 年 10 月

目 录

part 1 基础知识

第 1 章 程序设计语言概述	1
1.1 程序设计语言的发展	1
1.2 C 语言程序结构(框架)特点	2
1.3 各种高级语言解题步骤	4
习题	5
第 2 章 用 C 语言表达式描述数学表达式	6
2.1 数的表示方法	6
2.2 数据类型	7
2.3 变量	11
2.4 运算符与表达式	13
2.5 数据的输入和输出	21
习题	30
第 3 章 C 语言程序中的流程控制语句	35
3.1 用流程图表示算法	35
3.2 用 C 语言的语句来描述程序中的分支算法	36
3.3 用 C 语言的语句来描述重复算法	44
习题	62
第 4 章 数组	70
4.1 一维数组的定义和引用	70
4.2 多维数组	75
4.3 字符数组与字符串	80
4.4 字符串处理函数	85
4.5 数组应用程序设计实例	89
习题	90

第 5 章 指针及指针变量	95
5.1 指针和指针变量	95
5.2 数组的指针和指向数组的指针变量	100
5.3 指针数组的概念	114
5.4 指向指针的指针型变量	117
5.5 字符指针变量和字符数组使用中应注意的问题	119
5.6 指针应用程序设计实例	122
习题	124
第 6 章 函数	128
6.1 模块化程序设计概念	128
6.2 变量的作用域和变量的生存期	132
6.3 函数调用中参数传递方式	138
6.4 调用一个函数取得返回值方式	148
6.5 使用全局外部变量的传递方式	150
6.6 函数的指针和指向函数的指针变量	151
6.7 函数的嵌套调用和递归调用	153
6.8 带形参的主函数	160
习题	161
第 7 章 编译预处理	166
7.1 宏	166
7.2 “文件包含”处理	171
7.3 条件编译	173
习题	175
第 8 章 结构体数据类型及共用体数据类型	178
8.1 结构体数据类型的定义	178
8.2 结构体数据类型变量的定义和引用	180
8.3 结构体数据类型数组的定义和引用	183
8.4 结构体型变量的指针和指向结构体型变量的指针变量	185
8.5 在函数间传递结构体型数据	189
8.6 用指向结构体型指针变量处理链表(结构体型数据应用程序设计实例)	194
8.7 共用体型	203
8.8 枚举型	207
8.9 用户自定义类型	209
习题	211

第 9 章 文件.....	215
9.1 文件概述	215
9.2 文件的打开与关闭操作	217
9.3 文件中的数据顺序读写操作	220
9.4 对文件中的数据随机读写操作	228
9.5 错误测试函数	230
习题.....	232
第 10 章 位运算	235
10.1 位运算符和位运算.....	235
10.2 位运算应用程序设计实例.....	238
习题.....	239
附 录.....	241
附录 I 常用字符与 ASCII 代码对照表	241
附录 II 关键字及其用途.....	242
附录 III 运算符的优先级别和结合方向.....	243
附录 IV C 库函数	244
参考文献.....	254

第 1 章 程序设计语言概述

1.1 程序设计语言的发展

随着计算机的不断普及和推广,面向科学计算方面的程序设计语言也伴随着计算机技术的不断发展而更新换代。

1. 面向机器语言

(1) 机器语言(二进制语言)。程序设计只能用计算机直接可识别并执行的指令(一串二进制代码)来表示操作功能和操作数。每种型号的计算机(不同的 CPU),设计时都有自己的指令系统,这样就造成机器语言的不通用性。如 8086 CPU 芯片、51 系列单片机、Pentium 系列 CPU 芯片都有自己独立的指令系统。

下面是 DJS_130 小型计算机两条指令:

1011011000000000 加法运算指令

1011010100000000 减法运算指令

其中:指令中的操作功能是由指令第 9,8 两位决定,即 10 为加法运算、01 为减法运算;第 15,14 两位决定的寄存器保存被加数、被减数,第 13,12 两位决定的寄存器保存加数、减数;运算的结果保存在由指令第 15,14 两位决定的寄存器中。

用机器语言编程序,就是从所使用的 CPU 的指令系统中挑选合适的指令,组成一个指令系列。这种程序虽然可以被机器直接理解和执行,却由于它们不直观、难记、难理解、易错,只能被少数计算机专业人员掌握,同时程序设计的效率很低。这种繁重的手工方式编程与高速、自动工作的计算机极不相称。但它的运算速度要远远高于其他高级语言,正因如此,很多实时自动控制系统,如宇宙飞船、洲际导弹、工业生产中的压力、温控等实时控制系统都采用机器语言进行编程。

(2) 汇编语言。用“助记符”来代替机器语言中的指令代码,也称为符号语言。如前面的两条机器指令在 8086 宏汇编下可以写为

ADD AX,BX 加法运算

SUB AX,BX 减法运算

将这样对应的机器语言指令加、减、乘、除、移位、传送等设计成汇编指令集。

这种用助记符号描述的指令系统,称为符号语言或汇编语言。

汇编语言要比机器语言前进一大步。克服了机器语言指令(一长串二进制代码)不好记忆、易错的缺点。用汇编语言编程,程序的设计效率及质量都有所提高。但新的问题是人为的规定的助记符,汇编指令是机器不能直接识别并执行的指令。要有翻译软件将它翻译成机器能够直接可识别并执行的指令。这个翻译转换过程称为“代真”。代真后得到的机器语言程序称为目标程序(object program),代真以前的程序,称为源程序(source program)。

汇编语言与机器语言,都是依 CPU 的不同而异,它们都称为面向机器的语言。用面向机器的语言编程,可以编写出效率极高的程序。但是程序员用它们编程时,不仅要考虑解题思路,还要熟悉机器的内部结构,并且要“手工”地进行存储器分配。这种编程的难度仍然很大,给计算机的普及推广造成很大的障碍。

2. 高级语言

由于面向机器语言受到机型的限制,指令集不可能统一起来,从科学计算及工程计算角度来看,程序设计的关键是将问题及解决问题方法的算法过程用计算机语句描述出来,使程序设计者把精力集中于解题思路和方法上;并使程序设计语言开始与解题方法相结合。称之为高级语言。如 FORTRAN, Basic, Pascal, C, C++, Visual Basic, Visual Foxpro, Visual C++ 等各种高级语言相继出现。

(1) 面向过程的语言。将解题过程看做是数据被加工的过程。数据和加工数据的代码程序分开存放于内存不同位置。基于这种方法的程序设计语言称为面向过程的设计语言。C, FORTRAN, Basic, Pascal 等语言都是一种面向过程的程序设计语言。

(2) 面向对象的程序设计语言。面向对象的程序设计是一种结构模拟方法,它把客观世界看成是由许多对象(object)所组成,对象之间通过互相发送和接收消息进行联系;消息激发对象本身的运动,形成对象状态的变化。从程序结构的角度,每个对象都是一个数据和方法(程序代码)的封装体——抽象数据类型。

从分类学的观点看,客观世界中的对象都是可以分类的。也就是说,所有的对象都属于特定的“类”(Class),或者说每一个对象都是类的一个实例。因而,面向对象的程序设计的一个关键是定义“类”,并由“类”生成“对象”。

面向对象的程序比面向过程的程序更清晰、易懂,更适宜编写更大规模的程序,已成为现代程序设计的主流。面向对象的程序设计语言有 C++, Visual Basic, Visual Foxpro, Visual C++, Java 等。这里还要指出的是由 C 语言派生出的 C++ 语言,它是一种不仅可以支持面向对象的程序,还是一种可用来编写面向过程的程序。

1.2 C 语言程序结构(框架)特点

一种语言之所以能存在和发展,并具有生命力,总是有其不同于(或优于)其他语言的特点。从纯工程计算角度来看 FORTRAN 语言要比 C 语言功能更强些,如复型数运算的功能。

C语言除具有一定的计算功能外,还与机器硬件联系密切,可兼顾软件开发功能,如UNIX操作系统就是用C语言开发出来的。所以教学上选择的程序设计语言在20世纪90年代初由FORTRAN语言转向C语言。

下面通过一个小例题来看C语言程序结构(框架)特点。

例 1.1 在主函数中从键盘上接收3个实型数据,调用一个求平均值的函数计算出平均值,并在主函数中输出。

```
float ave(x1,x2,x3)                /* 函数头 */
float x1,x2,x3;                    /* 形式参数数据类型描述 */
{float a;                          /* 定义 a 为实型变量 */
a=(x1+x2+x3)/3;                   /* 将表达式的值赋给 a */
return a;                         /* 返回 a 的值 */
}

main( )                            /* 主函数 */
{float x,y,z,a1;                   /* 定义 x,y,z,a1 为实型变量 */
scanf("%f,%f,%f",&x,&y,&z);        /* 从键盘上接收 3 个实型数据,分别赋给 x,y,z */
a1=ave(x,y,z);                    /* 调用求平均值的函数 */
printf("%f\n",a1);                /* 输出 a1 的值 */
}
```

这样的程序学生目前接触尚早,因这种语言很多的语法规则未讲述,举这个例题的目的是通过它总结出C语言程序结构(框架)的特点:

(1)任何一个C语言程序是由若干个函数构成的,其中必须有一个函数名为main()的主函数(唯一的,不可多,也不能少)。

(2)函数(自定义、标准库函数)。

函数结构:由函数头和函数体两部分构成。

函数头:在函数名前有函数的存储类型符及数据类型符,函数名后面跟一对括号,括号内为形式参数表,包括跟在后面一条形式参数数据类型描述语句。

函数体:在函数头下面一对花括号内的内容称为函数体。

(3)一条语句以“;”号为结束标志。一行可写多条语句,长语句也可占多行(不必加续行符)。但为了可读性好,通常采用缩进、对齐方式书写或编辑源程序。

(4)注释内容是以“/*”开头直到“*/”之间的内容。注释内容不是程序的组成部分,不参加程序编译。

(5)一个C语言程序运行总是从主函数开始,并在主函数中结束。各函数在程序内的位置可以任意,但最好把主函数安排在最后,免得要在主函数中声明被调函数。

(6)主函数可以调用其他函数,其他函数间也可以相互调用,但不能调用主函数。

(7)除符号常量外,程序一律用小写字母书写及编辑。

1.3 各种高级语言解题步骤

各种高级语言采用编译方式将源程序转换为二进制的目标代码。编写好一个程序到完成运行,一般经过以下几个步骤。

1. 编辑建立源程序

所谓编辑,是按某种高级语言的语法规则书写的程序逐个字符输入到计算机内存;修改源程序;将修改好的源程序保存在磁盘文件中(源程序建立)。编辑的对象是源程序,它是以 ASCII 代码的形式输入和存储的,不能被计算机执行。

各种高级语言源程序的区分是通过文件的扩展名来实现。

语言名	文件名
C	文件标识名 . C
Pascal	文件标识名 . pas
FORTRAN	文件标识名 . FOR
Basic	文件标识名 . bas
C++	文件标识名 . cpp
.....	

注:所有高级语言源程序的扩展名都是在存盘时,由这种语言的支持软件自动加上的,程序设计者只须输入程序的标识名即可。

2. 编译源程序

计算机无法识别源程序,不能直接运行,必须借助本语言翻译软件将已编辑好的源程序(已存储在磁盘文件中)翻译成二进制的目标代码。在编译时,还要对源程序进行语法检查,如发现有错,则在屏幕上显示出错信息,此时应重新进入编辑状态,对源程序进行修改后再重新编译,直到通过编译为止。编译后得到的二进制代码在 UNIX 下是后缀为“. o”的文件,在 MS-DOS 下是后缀为“. obj”文件。应当指出,经编译后得到的二进制代码还不能直接执行,因为每一个模块往往是单独编译的,必须把经过编译的各个模块的目标代码与系统提供的标准模块(如 C 语言中的标准函数库)连接后才能运行。

3. 连接

将各模块的二进制目标代码与系统标准模块经连接处理后,产生可执行文件,它是计算机能直接执行的文件。在 UNIX 下文件扩展名以“. out”为后缀(例如, f. out),在 MS-DOS 下文件扩展名以“. exe”为后缀(例如, f. exe)。

4. 执行

执行一个经过编译和连接的用户可执行的文件。只有在操作系统的支持和管理下才能执行它。

Turbo C 将编辑、编译、连接、运行的过程集成为一体化,以菜单形式给出。关于 Turbo C

集成环境及使用见与本教材配套的《C 语言程序设计上机实验指导与习题集及章末习题解答》一书。

习 题

1. 什么是面向机器的语言？
2. 什么是高级语言？
3. 什么是面向过程的模块化程序设计语言？
4. 什么是面向对象的程序设计语言？
5. 在计算机上运行一个 C 语言编写的程序，必须经过哪几个操作步骤才能得到结果？
6. 指出 C 语言程序结构(框架)特点。

第 2 章 用 C 语言表达式描述数学表达式

一个大型工程项目的设计,首先要建立描述物理过程的数学模型。对数学模型进一步处理,如飞机六自由度非线性联立方程,在一定范围内进行线性化处理,进而产生飞机在空中运动规律的线性六自由度联立方程。所以科学计算或工程计算是从已有的公式开始,利用计算机高速精确的解题能力,得到接近理论值的精确结果。

组成数学方程式的基本要素是表达式。本章中心问题是介绍怎样用 C 语言的表达式来描述数学中的表达式。

C 语言的表达式组成有三个成分:常量、变量、运算符。本章分别介绍这三个成分在 C 语言中的具体规定及使用。

2.1 数的表示方法

1. 数学中数值的表示

(1) 一般形式。如 38.56(整数部分、小数点、小数部分)

0.52 纯小数

35 纯整数

(2) 指数形式(科学计数法)。如 $3.141\ 592\ 657\ 5 \times 10^0$

$0.314\ 159\ 265\ 75 \times 10^1$

$31.415\ 926\ 575 \times 10^{-1}$

2. C 语言中数值的表示

(1) 一般形式。整数,如 35

实数,如 38.56 或 0.52

(2) 指数形式。如 $3.1415926575 \times 10e0$

$0.31415926575 \times 10e1$

$31.415926575 \times 10\ e-1$

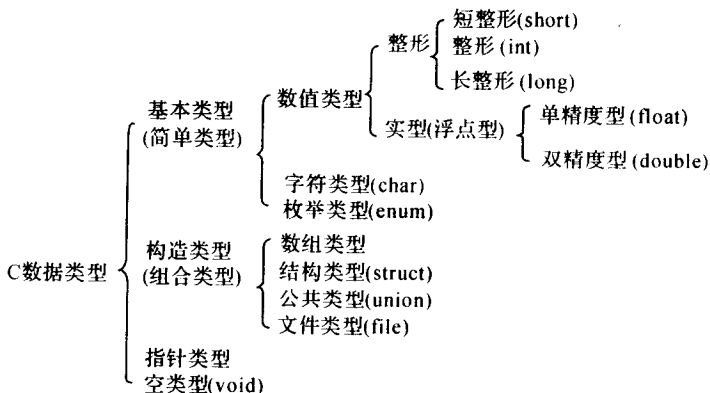
3. 数值在计算机中浮点表示

由于数据采用指数形式表示,小数点位置不定,小数点位置是浮动的这种表示形式称为数值的浮点形式。

在计算机中浮点数存储分配 4 个字节(bytes)。其中,数字部分为 3 个字节;指数部分为 1 个字节。

2.2 数据类型

在程序设计中,核心工作是对数据加工处理,数据是程序处理的对象。因此程序中用到的数据应描述。C语言规定数据类型如下:



构造类型数据是指由基本类型数据按一定的规律构造而成的。例如,有内在关联的若干个同类型数据组成的有序排列就是数组。

本章只介绍基本数据类型(枚举类型除外,在涉及结构体型一章中介绍),其他类型则在后续各章中陆续介绍。

2.2.1 常量

常量也称为常数,在程序运行期间具有固定值并且值不变的数据,称为常量。

常量是表达式中运算符(操作)操作的对象(数据)成分之一。例如,数学中的 π 代表圆周率,它的值为3.141 592 6,在程序中将PI定义为符号常量并且值为3.141 592 6,这样在程序中任何表达式里出现的PI就代表圆周率 π 的值为3.141 592 6。

C语言规定,常量类型为整型、实型、字符型、字符串型、符号常量。

1. 整型常量

整型常量就是整数(正整数、负整数、0)。

Turbo C规定整型常量为带符号的数据类型与无符号的数据类型两种。在内存中存储整数时,一般以其最高位(即最左边一位)表示数的符号,以0表示正,以1表示负。在计算机中数值是以补码形式存放的(好处是将减法运算转为加法运算,这样计算机的线路简单,造价低)。一个正数的补码就是该数的二进制数原码(如10的补码为0000000000001010),一个负数由它的原码求补码的过程是:①将原数转化为二进制表示形式;②这个负数的原码符号位不变,其他各位取反,然后再加1。这样就得到该负数的补码。例如求-10的补码可以这样求:①-10的二进制原码为1000000000001010;②-10的符号位不变(即最高位“1”不动,其他各位取反得到该数的反码11111111 11110101,再加1,就得到-10的补码11111111 11110110。

下面是一10的16位补码在计算机内存中的存储形式。

11 11 11 11	11 11 01 10
-------------	-------------

显然,所有负数的二进制补码的最高位必然是 1。从该位的状态(0 或 1)可以判定该数的正或负。有关补码的知识不属于本书的范围,不作详述。在此读者只需要一般了解什么是“补码”以及如何求得一个数的补码即可。以后的章节中有时要用到“补码”的概念。

(1) 带符号的整型常量。带符号的整型常量又分短整型(signed short int)和长整型(signed long int)两类。存储空间规定短整型常量分配 2 个字节(bytes);长整型常量分配 4 个字节。存取值的范围短整型常量为 $-32\,768 \sim 32\,767$ (15 位二进制所能保存的最大值);长整型常量为 $-2\,147\,483\,648 \sim 2\,147\,483\,647$ (31 位二进制所能保存的最大值)。

省略 signed(带符号),默认为隐含整型。所以下列三组描述是等价的:

int	}	等价
signed int		
signed		
short	}	等价
signed short		
long	}	等价
signed long		

(2) 无符号的整型常量。C 语言还允许使用无符号整数,同样无符号的整型常量也分为短整型(unsigned short int)和长整型(unsigned long int)两类。存储空间规定短整型常量分配为 2 个字节(bytes);长整型常量分配为 4 个字节(bytes)。

它将二进制形式存储 2 个字节或 4 个字节的最左端一位不作为符号位处理,而与右面各位一起用来表示数值。也就是它只能存放正整数,而不能存放负整数,与同样长度的有符号正整数相比,它所能表示的数值的最大范围扩大了 1 倍。一个无符号 2 个字节整数表示数值的范围为 $0 \sim 65\,535$,而一个有符号的 2 个字节整数表示正整数数值的范围最大数为 32 767。一个无符号的 4 个字节整数表示正整数数值的范围最大数为 $0 \sim 4\,294\,967\,295$ 。它用来表示那些只有正值的数值(例如人口、存款、库存、考分等)是比较适宜的。

C 语言中规定正整数前的“+”可以省略,默认值为正。

(3) 一个整型常量可以用不同的进制来表示。在 C 语言中,整型常量可以使用十进制数、八进制数、十六进制数等几种形式书写及编辑。C 语言规定,程序中凡出现以数字 0(零)开头,后面跟若干位的数字,C 系统一律作为八进制数处理;凡出现以 0x(零 x)开头,后面跟若干位的数字,C 系统一律作为十六进制数处理;以数字 1~9 开头的整型常量一律作为十进制数处理。例如,人们习惯的十进制整型常量 20,下面的书写及编辑都是合法的:

20, 024, 0x14

在计算机中二进制各位上的数字都是相同的,所以它们代表的是同一个数。另外长整型常量在书写及编辑时,在数的末尾加一个小写的“l”字母来与短整型常量区别,C 系统编译时一律按长整型常量处理。

例如 201,0241,0x141 分别表示十进制的 20 为长整型常量,八进制的长整型常量 24,十六进制的长整型常量 14。

2. 实型常量

实数在 C 语言中又称浮点数。浮点数只采用十进制数表示,实数有两种表示形式:

(1) 数的一般表示形式。它由数字和小数点组成(注意必须有小数点)。如 0.123,.123,123.,123.0,0.0 都是数的一般(十进制)表示形式。

(2) 指数表示形式。如 123e3 或 123E3 都代表 123×10^3 。注意字母 e(或 E)之前必须有数字,且 e 后面指数必须为整数,如 e5,2.1e5.8,.e9,e 等都是不合法的指数形式。而 123.45e-12 是合法的指数形式。指数表示形式输出有效数字 10 位,其中指数部分占 3 位。

C 语言规定:

1) 实型常量只使用十进制表示数值。

2) 不采用无符号形式表示数值。

3) 单精度(float)在内存中存储分配 4 个字节;双精度(double)在内存中存储分配 8 个字节。

3. 字符常量

C 语言规定用一对单撇号括起来的内容称为字符常量,如 'a','A','%','\#','5' 都是字符常量。

注意,单撇号只是一种标记,它不属于字符型数据自身的组成部分,或者说是字符常量的定界符。

C 语言规定字符常量在内存中存储分配 1 个字节,存储不是字符本身,而是这个字符对应的 ASCII 码(美国通讯采用的标准码)。

如 'A'...'Z'	'a'...'z'	'0'...'9'	'\r'	'\41'	'\x3d'
65...96	97...128	48...57	13	33	61

注:下一行的数字是上一行字符对应的 ASCII 码。

标准 ASCII 字符的允许范围为 0~127,用二进制 7 位表示就可以了,最左一位补 0。例如字符 'A' 的 ASCII 码为 65,ASCII 码的二进制存储形式为“01000001”。这时第一位是否作为符号位其作用都是一样的(因为一个数前面加一个零或多个零其值不变)。但是有些计算机系统(例如 IBM),除了使用 ASCII 码为 0~127 的字符外,还扩充使用 128~255 的字符(见本书附录 I)。这些字符需要 8 位二进制来表示,它们多是图形字符。例如字符“≥”,其 ASCII 码为 242,即二进制数 11110010(八进制数 362),其第一位为 1。怎样处理这个最左端的位,标准 C 无统一规定,有的系统(如 Turbo C,PDP,VAX-11 等)把 char 型变量隐含指定为 signed 型,即将其最左端一位作为符号位。某些系统把字符型数据处理为带符号的,与带符号的整型数据处理方法一致,便于按同一原则运算处理。因此如果有一个字符变量被赋值为八进制数 0362(即二进制数 11110010),那么按十进制数形式输出时,屏幕上看到的不是十进制数 242,而是一个负数-140。

4. 字符串常量

C 语言规定,把用一对双撇号括起来的内容称为字符串常量。如:"hello","Programming C","A"等都是字符串常量。

字符串以双撇号为定界符,但双撇号并不属于字符串自身,只是一种标志。