

高等学校教材
Textbook for Higher Education

C 语言程序设计教程

谷建华 主编
王备战 副编



西北工业大学出版社

高等学校教材

C 语言程序设计教程

主 编 谷建华

副主编 王备战

编写组成员 (以姓氏笔划为序)

王备战 谷建华 杨志义

张盛兵 胡 飞 谭 毅

西北工业大学出版社

【内容简介】 本书是按照面向应用、重视实践、便于自学的原则编写的,其特点是通俗易懂,便于初学者学习和掌握。

正文部分的主要内容为 C 语言概述、数据类型、运算符、表达式及基本的 I/O 函数、流程控制语句、数组、函数、指针、结构体与联合体、编译预处理、文件、面向对象的程序设计 C++ 等。附录部分介绍了 Turbo C 的基本概念、环境、使用以及部分函数。

本书可作为高等工科院校本、专科学生的教学用书,也可供广大 C 语言程序设计人员参考。

图书在版编目(CIP)数据

C 语言程序设计教程/谷建华主编. —西安:西北工业大学出版社,2001. 1

ISBN 7-5612-1322-0

I. C... II. 谷... III. C 语言-程序设计-教材 IV. TB312

中国版本图书馆 CIP 数据核字(2000)第 85569 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—8493844

网 址: <http://www.nwpup.com>

印 刷 者:西安市向阳印刷厂

开 本:787 mm×1 092 mm 1/16

印 张:13

字 数:315 千字

版 次:2001 年 1 月第 1 版 2001 年 1 月第 1 次印刷

书 号:ISBN 7-5612-1322-0/TP·200

印 数:1~3 000 册

定 价:20.00 元

前 言

C 语言的学习、应用从来都是计算机教学与实践的关键和难点。C 语言是目前国内外最广泛使用的程序设计语言之一, C 语言功能丰富、表达能力强、使用方便灵活、应用面广、目标程序效率高、可移植性好, 既具有高级语言的优点又具有低级语言的许多特点, 因此特别适合于编写系统软件。著名的 UNIX 操作系统就是用 C 语言开发的, 因此 C 语言也就随着 UNIX 的出名而闻名。

本书介绍的是 ANSI C——美国国家标准协会 (ANSI) 和国际标准化组织目前都承认的 C 语言标准版本。本书是按照面向应用、重视实践、便于自学的原则编写的, 书中选取了大量的实例、练习题以及应用题, 集中了软件工程的规则并强调了用结构化程序设计方法编写程序。全书共分为两部分, 正文部分分为十章: 概述, 数据类型、运算符、表达式及基本的 I/O 函数, 流程控制语句, 数组, 函数, 指针, 结构体与联合体, 编译预处理文件, 面向对象的程序设计 C++。附录部分介绍了 Turbo C 的基本概念、环境、使用以及部分函数。

本书由谷建华、王备战、杨志义、张盛兵、胡飞和谭毅六人合作编写。谷建华、王备战负责全书的统稿和定稿。我们对本书所选用的参考文献的著作者表示真挚的感谢。限于作者水平, 书中难免有不足之处, 欢迎广大读者批评指正。

编 者

2000 年 10 月

目 录

第一章 概述	1
§ 1.1 C 语言的起源	1
§ 1.2 C 语言的特点	2
§ 1.3 C 语言的基本字符和标识符	3
§ 1.4 简单的 C 程序介绍	4
§ 1.5 C 程序的结构特点和书写风格	6
习题一	7
第二章 数据类型、运算符、表达式及基本输入输出函数	8
§ 2.1 数据类型的概念	8
§ 2.2 常量	10
§ 2.3 变量	13
§ 2.4 数据类型转换	14
§ 2.5 运算符和表达式	16
§ 2.6 位运算符	23
§ 2.7 C 语言基本输入/输出函数	26
习题二	32
第三章 流程控制语句	35
§ 3.1 分支控制语句	35
§ 3.2 循环控制语句	43
§ 3.3 break 语句和 continue 语句	47
§ 3.4 循环的嵌套	48
习题三	50
第四章 数组	52
§ 4.1 一维数组	52
§ 4.2 多维数组的定义与引用	55
§ 4.3 字符数组	59

习题四	63
第五章 函数	64
§ 5.1 函数及其定义	64
§ 5.2 函数的参数和函数值	68
§ 5.3 函数的调用	71
§ 5.4 局部变量和全局变量	76
§ 5.5 动态存储变量和静态存储变量	80
§ 5.6 内部函数与外部函数	84
习题五	85
第六章 指针	87
§ 6.1 指针的概念及指针变量	87
§ 6.2 指向数组的指针	92
§ 6.3 指向字符串的指针	99
§ 6.4 指向函数的指针	101
§ 6.5 指针数组与指向指针的指针	104
§ 6.6 有关指针类型与指针运算小结	106
习题六	109
第七章 结构体和共用体	111
§ 7.1 结构体	111
§ 7.2 线性链表及其处理	124
§ 7.3 位段	131
§ 7.4 共用体	133
§ 7.5 枚举类型	135
§ 7.6 用 typedef 定义类型	136
习题七	137
第八章 编译预处理	138
§ 8.1 C 语言的预处理命令	138
§ 8.2 #include 命令	138
§ 8.3 #define	140
§ 8.4 条件编译	142
习题八	145
第九章 文件	146
§ 9.1 文件的基本概念	146
§ 9.2 文件类型指针	147

§ 9.3 文件的打开与关闭	148
§ 9.4 文件的读和写	150
§ 9.5 文件读写定位	158
§ 9.6 文件的检测函数	161
习题九	162
第十章 面向对象的程序设计——C++	163
§ 10.1 背景知识	163
§ 10.2 面向对象的一些基本概念与特性	163
§ 10.3 C++介绍	166
§ 10.4 对象类	170
习题十	183
附录	185
附录一 turbo C 基本概念	185
一、启动 Turbo C	185
二、基本操作	185
附录二 编写运行 C 语言程序	187
一、进入编辑状态	187
二、状态行	187
三、相关热键	187
四、编辑程序	188
五、运行程序	190
六、简单说明	191
附录三 错误诊断	191
附录四 程序调试	193
一、观察变量值	194
二、程序运行控制	195
附录五 常用函数	197
参考文献	202

第一章 概 述

本章主要介绍 C 语言的起源、应用、概貌以及构成。通过几个简单的例子说明了 C 语言的特点。

§ 1.1 C 语言的起源

C 语言出现的历史背景与操作系统息息相关。以前的操作系统等系统软件主要是用汇编语言编写的,但汇编语言依赖于具体的计算机硬件,程序的可读性和可移植性都比较差。而一般的高级语言又难以实现汇编语言的某些功能,所以不宜用来编写操作系统。于是,人们寻求一种具有高级语言的优点又具有低级语言的特点的语言,C 语言就是在这种情况下出现。

C 语言是 70 年代初由美国贝尔实验室的 Dennis M. Ritchie 发明并首先在配备 Unix 操作系统的 DEC PDP-11 计算机上实现的。C 语言是一种比较古老的语言 BCPL (Combined Programming Language) 发展过程中的产物。BCPL 是由 Martin Richards 开发的,它影响了由美国贝尔实验室的 Ken Thompson 发明的 B 语言,而 B 语言又导致了 C 语言在 70 年代的产生。

最初的 C 语言是为了描述和实现 Unix 操作系统而设计的。1973 年,K. Thompson 和 D. M. Ritchie 两个人合作把 Unix 的 90% 以上的代码用 C 语言进行了改写,即 Unix 第五版 (原来的 Unix 操作系统是由美国贝尔实验室的 K. Thompson 和 D. M. Ritchie 用汇编语言开发成功的)。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 语言是为开发 Unix 操作系统而研制的,它随着 Unix 的出名而闻名。C 语言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。随着 1975 年 Unix 第六版的推出和面向对象程序设计的出现,C 语言的突出优点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 第七版中的 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。

随着微型计算机的发展,出现了一大批 C 语言系统。由于当时不存在统一的标准,因而不同的系统存在着差异和不相容。为了改变这种局面,1983 年美国国家标准协会 (ANSI) 根据 C 语言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。目前大多数 C++ 和 C 语言编译器都实现了这个标准。本书的叙述基本上以 87 ANSI C 为基础。

§ 1.2 C 语言的特点

C 语言之所以具有强大的生命力,能被推广使用,成为世界上应用最广泛的几种计算机语言之一,其主要原因源于其以下特点。

一、C 语言是中级语言

C 语言通常被称为中级计算机语言。这并不意味着它的功能差,难以使用,或者比 BASIC, Pascal 那样的高级语言原始,也不意味着它会像汇编语言那样给使用者带来类似的麻烦。C 语言之所以被称为中级语言是因为它把高级语言的成分同汇编语言的功能结合起来了,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,因此它比其它高级语言更接近硬件系统。作为中级语言, C 语言支持对位、字节和地址这些有关计算机功能的基本成分进行操作,这使得它非常适于经常进行上述操作的系统程序设计。事实上 C 语言被程序员广泛使用的一个原因是它可以代替汇编语言。

二、C 语言是结构化程序设计语言

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。用 C 语言编写的程序易读、易维护。

C 语言的主要结构成分是函数。在 C 语言中,函数作为程序构件使得 C 语言便于进行模块化程序设计。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个相对独立的模块,模块间通过函数来实现相互连接。

三、运算灵活、数据类型丰富

C 语言具有使用灵活的运算符,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。

C 语言具有比较丰富的数据类型,它不仅具有字节型、整型(基本整型、短整型、长整型、无符号整型等)、单精度实型、双精度实型和空类型等基本数据类型;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构造(也称复合)数据类型,这使 C 语言具有很强的数据处理能力。

四、高效性

C 语言程序可以通过 #define, #include 等预编译命令,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软件开发效率。

五、可移植性

C 语言程序具有较高的可移植性,可移植性表现为可将为某种计算机写的软件改编到另一种机器上实现,可移植性节省了时间和财力。C 语言不包含依赖于具体硬件的输入/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,便于在不同的计算机系统间移植。

六、C 语言简洁、紧凑,使用方便灵活

C 语言总共 32 个关键字(其中 27 个来源于 Kernighan 和 Ritchie 的公认标准,另外 5 个

是 ANSI 标准化委员会增加的),这些关键字构成了 C 语言的命令集。比 IBM PC 用的 BASIC 的 159 个关键字要少得多。C 语言有 9 种控制语句,程序的书写形式自由。

§ 1.3 C 语言的基本字符和标识符

构成 C 语言的基本字符和标识符如下。

一、字符集

- 1. 英文字母(大、小写字母共 52 个): 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) " (双引号) ! # ,

二、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 32 个关键字,如表 1.1 所示。

表 1.1 ANSI C 标准定义的 32 个关键字

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

表 1.1 所列举的 32 个关键字与标准 C 的文法相结合形成了程序设计语言 C。C 语言区分字母的大写与小写,else 是关键字,ELSE 则不是, C 的关键字都是小写的。在 C 程序中,关键字不能用于其它目的,即不允许将关键字作为变量名或函数名使用。

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,这类标识符称为 预定义标识符。对于这类标识符,虽然 C 语言准许程序设计做其他使用(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,建议还是不要把这类标识符再定义为用户标识符(用户定义标识符)使用。预定义标识符包括预编译程序命令(define, undef, include, ifdef, ifndef, endif, line 等)和 C 编译系统提供的库函数名。

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。这类标识符的命名规则如下。

(1) 标识符的构成是由字母、数字和 下划线组成,但第一个的字符必须是字母或下划线。我们建议变量名用小写字母,常数名用大写字母。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。良好

的标识符命名习惯是一个优秀程序员的基本素质之一。

(2) 大小写英文字母含义不同, SUM, Sum, sum 是完全不同的标识符。

(3) 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。

为了使程序清晰、易读,我们在定义标识符时应注意如下几点:① 名字要有明确含义,应尽量选用具有一定含义的英文单词或汉语拼音进行命名,努力做到“见名知意”、“同名同意”、“同意同名”。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 而不要用 a 等。如果所选用的英文单词太长,可采用公认的缩写方式。② 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。③ 对于由多个单词组成的标识符,建议用下划线将各单词隔开,以增强可读性。例如 average_score。④ 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,“c”表示字符型等以增加程序的可读性。

§ 1.4 简单的 C 程序介绍

为了说明 C 程序组成的特点,我们先看几个简单的 C 程序实例,然后从中分析 C 程序的特点。

例 1.1 编写输出“Hello World!”的程序。

```
main()  
{  
    printf("Hello World! \n");  
}
```

这是一个最简单的 C 程序,它把字符串“Hello World!”输出后换行。该程序由一个函数 main() (称为主函数) 构成。每个 C 程序都必须有一个主函数,花括号 { } 所括的内容是 main 的函数体。

printf 是由系统提供的标准库函数,是格式化的输出函数,它完成输出功能。C 语言的输出是由函数来完成的。“Hello World!”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,称为转义字符。有关转义字符,在后面将具体介绍。printf() 后的分号是语句结束符,C 的每一个语句都以“;”结束。

例 1.2 从键盘输入 5 个数,计算这 5 个数的平均值。

```
main()  
{  
    float a,b,c,d,e,average; /* 说明 a,b,c,d,e,average 为单精度实数 */  
    printf("Please input the five numbers: \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); /* 输出 5 个数的平均值 */  
}
```

该程序运行时,首先提示你输入 5 个数 a,b,c,d 和 e,然后计算并输出平均值。在此程序中,/* */ 是注释语句,/* 和 */ 中包含要注释的内容,该语句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。恰到好处的注释也是一名优秀程序员的基本素质之一。“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 定义为单精度实型数。在 C 语言程序中的所有变量,都要先定义后使用。

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,要求从键盘输入实型变量 a, b, c, d, e 的值。

“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 从键盘输入两个数,编写程序输出两个数中的较大者。

```
#include <stdio.h>

main( ) /* 主函数 */
{
    int a, b, i_max; /* 定义变量 */
    printf("please input two integer numbers a, b: \n");
    scanf("%d, %d", &a, &b); /* 在此输入 a, b 的值 */
    i_max = max(a, b); /* 调用函数 */
    printf("\n max _ num is : %d", i_max); /* 输出结果 */
}

int max(x, y)
int x, y;
{
    int z;
    if (x > y) z = x;
    else z = y;
    return(z); /* 向主程序返回 z 的值 */
}
```

此程序由两个函数组成,除了主函数 main() 之外,还有一个计算 x 和 y 的较大值的用户自定义函数 max()。“int max(x, y)”说明函数的返回值类型为 int 类型,函数名字为 max,函数的形式参数为 x, y。

“int x, y;”说明各参数的类型。

return(z) 将求解的结果返回主函数。

该程序的执行是从 main() 函数开始,当主函数执行到 i_max = max(a, b) 语句时,控制被传递给 max() 函数,当执行 return(z) 语句时,则结束 max() 函数,控制又被传递给 main() 函数,并把 max() 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。

本例用到了函数调用、实参和形参等概念,我们暂且不予深究,后面章节中将会深入学习,这里引用的目的是让我们对 C 语言程序的组成有一个初步的了解。

§ 1.5 C 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是编写一个好程序的前提。

一、C 程序的结构

从上节所介绍的简单的 C 程序中,我们可以看出 C 程序的结构具有如下特点。

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而且只能有一个主函数,主函数名必须为 main,其余函数的名字由程序设计者自定。因此,函数是 C 程序的基本单位。程序的执行总是从主函数开始,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,也可以是用户根据需要而编制的函数,程序的全部工作都是由函数来完成的。所以,我们说 C 语言是函数式语言。C 的这种特点容易实现程序的模块化。

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。函数说明部分指明函数返回值的类型、属性、函数名、参数和参数说明等,如例 1.3 中的 max 函数的说明部分为:

```
int max(x,y)
```

```
int x,y;
```

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可执行语句。每个语句都以分号“;”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)、外部变量的说明等。关于这一点我们会在后面进行详细的介绍。

二、C 程序的书写风格

本着清晰第一、效率第二的原则,为了写出一个层次清晰、可读性强的程序,在书写 C 程序时应注意以下几点。

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,为了使程序层次清晰,应当把程序写成锯齿形,即在语句之前加上适当数量的空格字符,使处于同一层次的语句从同一列开始。

2. 一行一句 尽管 C 程序一行可写多句,但我们提倡一行一句的风格。

3. 变量名的使用应当“见其名而知其意” 在 C 语言程序中使用变量名时,应尽量用“约定成俗”的变量名或是其英文或是拼音的缩写,这样程序容易理解。在命名变量时我们还应当注意大小写字母的应用与区别。

4. 适当地使用注释 注释分为序言性注释和功能性注释。序言性注释一般写在函数定义之外,用以对整个函数加以说明,帮助读者理解整个函数。功能性注释用以说明某一块程序或某几个语句,以帮助程序阅读者理解这些程序块或语句。注释在编译时被忽略掉,只有其它的语句才被分析编译,故注释不会增加执行程序的长度和降低执行效率。

习题一

1.1 请简述 C 语言的主要特点。

1.2 C 语言的主要用途是什么?

1.3 简述 C 程序的构成。

1.4 参照本章例题,编写一个 C 程序,输出如下结果:

```
* * * * *
*               Happy New Year
* * * * *
```

1.5 C 语言的保留字有哪些?有何用途?

1.6 C 语言用户自定义标识符有何规定、有何用途?

1.7 判断下面标识符中哪些是 C 语言中合法的用户定义标识符?

A) b30 B) #abc C) typedef D) _max

E) a-b F) signed G) emp_number H) student * no

1.8 简述 C 程序的执行过程。

第二章 数据类型、运算符、表达式 及基本输入输出函数

本章主要介绍 C 语言的基本数据类型、运算符(算术运算符、关系运算符和逻辑运算符)、表达式和 C 语言的基本输入/输出函数。我们会发现 C 语言具有较丰富的数据类型、齐全的运算符和高效的表达式以及不同于其它语言的输入/输出的处理方式。表达式由 C 语言的原子要素数据和运算符构成,数据可以是变量或常量。由于数据是一切表达式的核心,因此本章将从 C 语言的基本数据类型开始介绍。

§ 2.1 数据类型的概念

著名的计算机科学家沃斯(Nikikláus Wirth)提出一个公式:

$$\text{程序} = \text{数据结构} + \text{算法}$$

这个公式说明了一个程序应包括两个方面的内容:数据的描述和操作的步骤(即算法)。

C 语言提供的数据结构是以数据类型的形式表现的。C 语言有 5 种基本数据类型:字符、整数、单精度浮点、双精度浮点和无值,分别用 char, int, float, double 和 void 来表示。C 语言中的其它数据类型都是从这些基本类型演变而来的,这些数据的长度和范围因处理器的类型和 C 语言编译程序的实现而异,ANSI C 强调的是每种数据类型的最小范围而不是字节长度(见表 2.1)。

表 2.1 ANSI C 标准中的数据类型定义

类 型	长度/bit	范 围
char(字符型)	8	-128~127
unsigned char(无符号字符型)	8	0~255
signed char(有符号字符型)	8	-128~127
int(整型)	16	-32768~32767
unsigned int(无符号整型)	16	0~65535
signed int(有符号整型)	16	同 int
short int(短整型)	16	同 int
unsigned short int(无符号短整型)	16	0~65535
signed short int(有符号短整形)	16	同 int
long int(长整型)	32	-2147483648~2147483647

续表

类 型	长度/bit	范 围
signed long int(有符号长整型)	32	同 long int
unsigned long int(无符号长整型)	32	0~4294967295
float(浮点型)	32	约 6 位有效数字
double(双精度型)	64	约 10 位有效数字
long double(长双精度型)	128	约 10 位有效数字

数据类型是程序设计中的一个重要概念,我们从以下几方面阐述数据类型的基本含义。

1. 数据类型规定了一个以值为其元素的集合,即规定了该类型中数据的定义域。例如,数值类型的数据,其值域是计算机所能表示的数值范围内的所有数据;逻辑类型的数据取值范围是“真”(TRUE)或“假”(FALSE);字符类型的数据取值域是某一字符集中的所有元素等。

2. 数据类型定义了一个运行集。不同类型的数据准许施加不同的运算。例如,对数值型数据可施加算术运算;对逻辑型数据可施加逻辑运算;对字符型数据可施加连接和求子串运算;对指针型数据准许进行加、减运算,而不准许进行乘除运算等。

3. 数据类型定义了数据在计算机内的存储及在书写中的表示方式。例如,一个字符型数据在计算机内占用一个字节。长度为 n 的字符串在计算机内的占用连续的 $n+1$ 个字节,且以 '\0' (空字符) 结尾。在书写中,字符数据用单引号(' ')括起来,字符串数据用双引号(" ")括起来等。

在高级语言中,每一个数据都属于一定的数据类型。C 语言提供如图 2.1 所示的数据类型。数据包含常量和变量,它们都属于上述某种数据类型,本章主要介绍基本数据类型,其他数据类型将在以后的章节中逐步介绍。

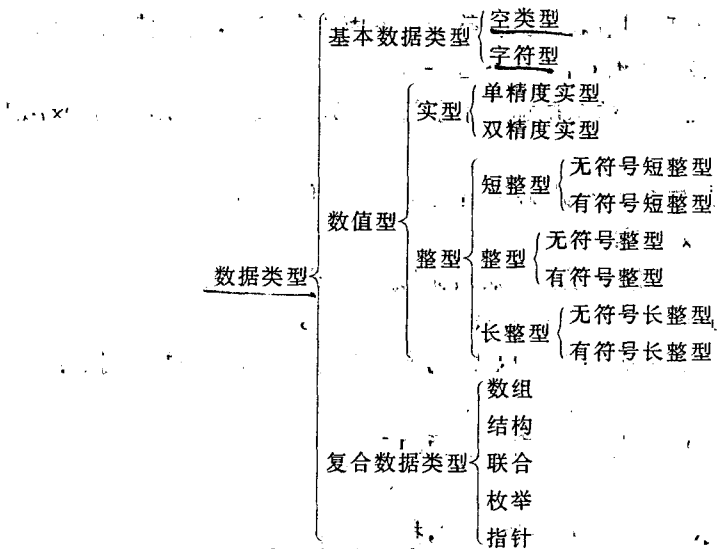


图 2.1 C 语言的数据类型

§ 2.2 常 量

常量是在程序运行过程中其值不发生变化的量。在C语言中,有数值常量、字符常量和字符串常量三种。除此这外,C语言程序中还经常使用另外一种表示形式的常量,称为符号常量。

2.2.1 数值常量

C语言中的数值分整数和实数两种。

一、整数

整型常量的表示形式有十进制数、八进制数和十六进制数三种。其表示形式如表 2.2 所示。

表 2.2 整型常量的三种表示形式

进 制	表 达 方 式	举 例
十进制数	必须以数字 1、2...9 之中的一个数开头	12 168
八进制数	以数字 0 开头	025 076
十六进制数	以 0x 或 0X 开头	0xff 0x25 0x12a

整型常量有正负之分,分别在其前面加上“+”或“-”表示,正数的符号“+”可省略。例如:
345 +215 -125 0125 -0215 0x3bf -0x4de。

整型常量有短整型数、一般整型数和长整型数。对多数计算机系统而言,短整型数一般占用两个字节,一般整型数占用两个字节(即16位二进制位),其取值范围是:

-32768 至 +32767(即 $-2^{15} \sim 2^{15}-1$)

超过该范围的整数用长整型数表示。长整型数占用4个字节,其取值范围为:

-214783648 至 +2147483647(即 $-2^{31} \sim 2^{31}-1$)

长整型数的表示方法是在数的末尾加一个字符 l 或 L,例如:-623l,622l,0x7dfll

二、实数

实数又称为浮点数,只用在十进制数中。它有单精度(float)和双精度(double)之分。其表示形式分为一般形式和指数形式两类。

一般形式的实数由整数部分、小数点和小数部分所组成。例如:

3.1415926 0.1666 -555.33 888.18 等

指数形式的实数由尾数、e 或 E(表示以 10 为底)和指数三部分组成。例如:

0.55e5 3.33E-3 8.88e+18

其中,0.55、3.33 和 8.88 为尾数,e 或 E 后面的 5、-3 和 +18 均是指数。它们分别表示 0.55×10^5 , 3.33×10^{-3} , 8.88×10^{18} 。

注意:用指数形式表示的浮点数必须有尾数,指数部分必须是整数。因此,诸如 e4, .e3, 0.25e4.5 和 e 等都是不合法的数。