



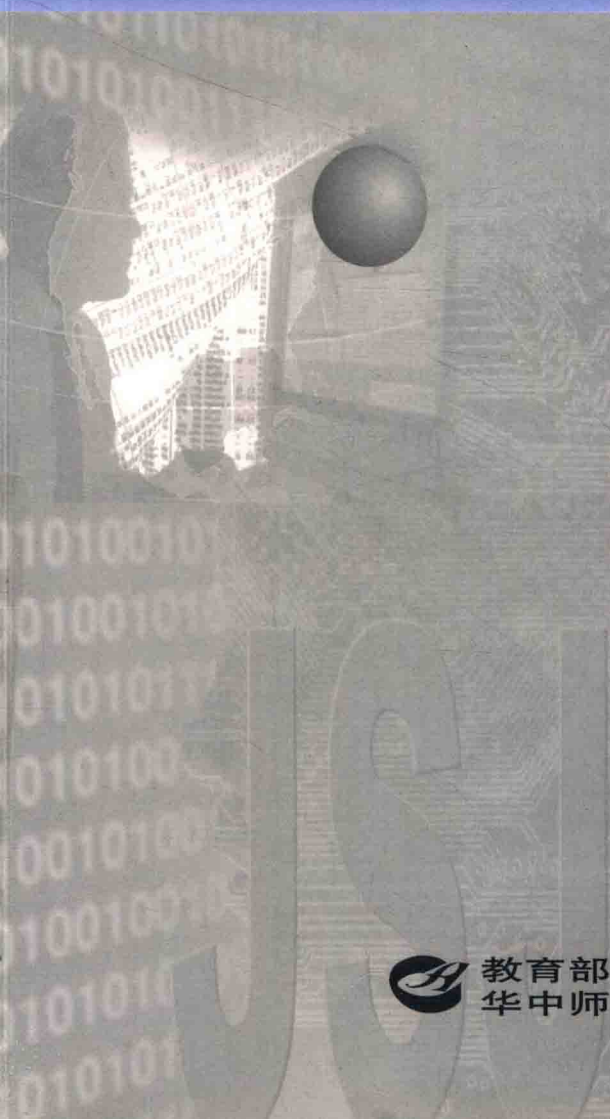
计算机科学与技术系列丛书

高级程序设计语言C

(上) (第二版)

GAOJI CHENGXU SHEJI YUYAN C (SHANG)

■ 李晓燕 贺 杰 主编



教育部直属师范大学
华中师范大学出版社

计算机科学与技术系列丛书

高级程序设计语言 C(上)

(第二版)

主 编: 李晓燕 贺 杰

副主编: 黄苏雨 汤 雷

编 者: (以姓氏笔画为序)

王晶晶	毛丽娟	刘智珺
汤 雷	杨云超	李晓燕
贺 杰	胡艳蓉	黄苏雨
彭 媛	熊 艳	

华中师范大学出版社

内 容 简 介

高级程序设计语言 C(上),全面地介绍了 C 语言的基本概念和各种语句,以及程序设计的基本方法和技能。全书共分 12 章:第 1 章引论,第 2 章数据及其类型,第 3 章运算符和表达式,第 4 章基本语句,第 5 章条件控制语句,第 6 章循环控制语句,第 7 章数组,第 8 章函数,第 9 章结构与联合,第 10 章指针,第 11 章文件,第 12 章编译预处理。本书可作为普通高校(民办高校)、二级学院和高职高专计算机专业和非计算机专业的本科和专科教材。

新出图证(鄂)字 10 号

图书在版编目(CIP)数据

高级程序设计语言 C(上)(第二版)/李晓燕,贺 杰主编. —2 版.

—武汉:华中师范大学出版社,2008. 8

(计算机科学与技术系列丛书)

ISBN 978-7-5622-3412-8

I. 高… II. ①李… ②贺… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 129453 号

高级程序设计语言 C(上) (第二版)

主 编:李晓燕 贺 杰

责任编辑:陈 勇

责任校对:张 忠

封面设计:罗明波

选题策划:第二编辑室

电话:027—67867362

出版发行:华中师范大学出版社©

地 址:武汉市武昌珞瑜路 152 号

邮编:430079

电话:027—67863426 67863040 67867076(发行部)

027—67861321(邮购)

传真:027—67863291

网址:<http://www.ccupress.com>

电子信箱:hscbs@public.wh.hb.cn

印刷:湖北新华印务股份有限公司

督印:章光琼

字数:290 千字

开本:787mm×1092mm 1/16

印张:12

版次:2008 年 8 月第 2 版

印次:2008 年 8 月第 1 次印刷

印数:1—3 100

定价:21.80 元

欢迎上网查询、购书

敬告读者:欢迎举报盗版,请打举报电话 027—67861321。

再版前言

第一版《高级程序设计语言 C》(上)于 2005 年出版,至今已三年整。经过三年的教学实践,该教材比较受教师和学生的欢迎。现在我们根据这三年来的教学实际和学生实际,对第一版教材的内容进行了必要的修正和补充,出版了第二版教材,使之更切合学生的实际需要。

第二版教材的作者作了部分变动,第 1 章引论、第 2 章数据及其类型由胡艳蓉编写,第 3 章运算符与表达式、第 4 章基本语句及附录由李晓燕编写,第 5 章条件控制语句、第 6 章循环控制语句由王晶晶和彭媛编写,第 7 章数组由刘智珺编写,第 8 章函数由黄苏雨编写,第 9 章结构与联合由熊艳编写,第 10 章指针由汤雷编写,第 11 章文件、第 12 章编译预处理由贺杰编写。

全书由李晓燕教授统稿。

第二版教材的出版,得到了作者所在高校和华中师范大学出版社有关领导及编辑的大力支持与帮助,在此深表谢意。

由于作者水平有限,书中可能出现的错误和不妥之处,敬请批评指正。

编 者

2008.6

第一版前言

《高级程序设计语言 C》分上、下两册。《高级程序设计语言 C》(上)全面地介绍了该语言的基本概念和各种语句,以及运用这些概念和语句进行程序设计的基本方法和技能。《高级程序设计语言 C(下)——实训教程》则包括了大量的实例和练习。

高级程序设计语言 C(上)全书共有 12 章:第 1 章引论和第 2 章数据及其类型由胡艳蓉编写,第 3 章运算符与表达式和第 4 章基本语句以及附录由李晓燕编写,第 5 章条件控制语句和第 6 章循环控制语句由彭媛编写,第 7 章数组由刘智珺编写,第 8 章函数由毛丽娟编写,第 9 章结构与联合由熊艳编写,第 10 章指针由汤雷编写,第 11 章文件和第 12 章编译预处理由杨云超编写。

全书由主编李晓燕教授统稿审定。

参加本书编写的作者是在高校从事计算机教学的教师和教授,有的在高校从教四十年,有十分丰富的教学经验。编者在编写教材的过程中,注意以学生为本,从学生的实际出发,贯彻了循序渐进、深入浅出的原则。

高级程序设计语言 C,是一种使用广泛、最受欢迎的计算机语言,该语言拥有许多众所周知的优点。本教材将会成为读者学习和掌握这门计算机语言的良好益友。

在教材编写和出版的过程中,编者得到了所在高校有关领导和华中师范大学出版社的大力支持和帮助,在此一并致谢。

由于水平有限,书中可能出现的缺点和错误,敬请批评指正。

编 者

2005 年 5 月

目 录

第 1 章 引论	(1)
1.1 计算机语言发展简况	(1)
1.2 C 语言的 发展和特点	(1)
1.3 C 语言的程序结构	(3)
1.4 上机步骤	(5)
第 2 章 C 数据及其类型	(7)
2.1 常量和变量	(7)
2.2 数据类型	(9)
2.3 变量赋值	(15)
第 3 章 运算符和表达式	(17)
3.1 算术运算符和算术表达式	(17)
3.2 关系运算符和关系表达式	(19)
3.3 逻辑运算符和逻辑表达式	(20)
3.4 赋值运算符和赋值表达式	(21)
3.5 其他运算符	(22)
3.6 位运算符	(23)
第 4 章 基本语句	(27)
4.1 C 语言中的语句	(27)
4.2 数据输入	(29)
4.3 数据输出	(32)
第 5 章 选择语句	(36)
5.1 if 条件语句	(36)
5.2 switch 多分支选择语句	(43)
5.3 程序举例	(46)
第 6 章 循环结构	(49)
6.1 while 语句	(49)
6.2 do-while 语句	(51)
6.3 for 语句	(53)
6.4 循环的嵌套	(56)
6.5 break、continue、goto 语句	(59)
6.6 程序举例	(62)
第 7 章 数组	(65)
7.1 一维数组	(65)
7.2 二维数组	(69)

7.3	多维数组	(74)
7.4	字符数组与字符串	(75)
7.5	程序举例	(80)
第8章	函数	(86)
8.1	函数概述	(86)
8.2	函数的定义和调用	(87)
8.3	变量的作用域和存储域	(96)
8.4	函数参数的传递方式	(101)
8.5	函数的嵌套调用和递归调用	(107)
第9章	结构与联合	(110)
9.1	结构	(110)
9.2	联合	(119)
第10章	指针	(123)
10.1	指针的概念	(123)
10.2	数组与指针	(125)
10.3	字符指针	(129)
10.4	结构指针	(130)
10.5	函数指针	(133)
10.6	指针函数	(135)
10.7	指针数组和指向指针的指针	(136)
第11章	文件	(138)
11.1	文件概述	(138)
11.2	文件系统	(138)
11.3	文件指针	(139)
11.4	缓冲文件系统	(140)
11.5	文本文件的顺序读写	(143)
11.6	二进制文件的顺序读写	(149)
11.7	文件定位函数	(151)
11.8	文件状态检查函数	(153)
第12章	编译预处理	(154)
12.1	宏定义	(154)
12.2	文件包含	(160)
12.3	条件编译	(161)
附录1	ASCII 码表	(164)
附录2	运算符优先级和结合性	(169)
附录3	常用 C 标准函数	(170)
参考文献		(184)

第1章 引 论

本教材主要介绍和讲解高级程序设计语言 C,介绍 C 语言的基本概念和各种语句,以及运用这些语句进行程序设计的基本方法和技巧。

本章将介绍计算机语言的发展简况和高级程序设计语言 C 的特点。

1.1 计算机语言发展简况

计算机语言是用来描述数据计算过程和处理过程的一种符号系统,是进行程序设计的一种工具,它经历了以下几个发展阶段:

1. 机器语言

机器语言是计算机发展初期使用的语言。所谓机器语言就是由机器指令组成的指令系统。机器指令是以二进制代码形式给出的,因此用机器语言编写程序相当复杂和繁琐,程序难读、难调试,并且缺乏通用性,不同计算机系统有各自的指令系统。

2. 汇编语言

汇编语言是一种面向机器的符号系统,它使用便于记忆的符号来代替机器指令。相对于机器语言,汇编语言前进了一步,它便于理解和记忆,但汇编语言大部分语句还是和机器指令一一对应的,语句功能不强,因此用汇编语言编写程序仍然很繁琐,花费时间,依然无通用性,所以汇编语言与机器语言同属于计算机低级语言。

3. 高级语言

计算机高级语言是一种比较接近自然语言和数学语言的计算机语言。高级语言可分为面向过程语言和面向对象语言。

面向过程语言有:FORTRAN、BASIC、COBOL、PASCAL、C、Ada 等。

面向对象语言有:C++、Java 等。

正是因为计算机高级语言接近自然语言,所以运用计算机高级语言编写程序就比运用计算机低级语言编写程序方便多了。

1.2 C 语言的发展和特点

C 语言是目前世界上最流行、使用最广泛的计算机高级语言之一。使用 C 语言明显优于其他计算机语言,许多大型应用软件和系统软件都是用 C 语言编写的。

1.2.1 C 语言的发展

C 语言的起源可以追溯到 1960 年出现的 ALGOL 60。1963 年,英国剑桥大学在 ALGOL 60 语言的基础上推出了 CPL(Combined Programming Language)语言,但是 CPL 语言难以实现。1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,推出了

BCPL 语言。1970 年,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,又作了进一步的简化,设计出了很简单且接近硬件的 B 语言,并用 B 语言写了第一个 UNIX 操作系统,在 DEC PDP-7 型计算机上实现。1971 年,在 DEC PDP-11 上实现了 B 语言。1972 年,美国的 Dennis M. Ritchie 在 B 语言的基础上设计出了 C 语言,并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。

为了推广 UNIX 操作系统,1977 年,Dennis M. Ritchie 发表了不依赖于具体机器系统的可移植的 C 语言编译程序。1978 年,Brian W. Kernighian 和 Dennis M. Ritchie 出版了《The C Programming Language》一书,从而使 C 语言成为目前世界上流行最广泛的高级程序设计语言之一。

随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种状况,1983 年,美国国家标准化协会(ANSI)为 C 语言制定了第一个 ANSI 标准,称为 ANSI C。1987 年,ANSI 又公布了新的 C 语言标准,称为 87 ANSI C。这个标准于 1989 年被国际标准化(ISO)组织采用,称为 ANSI/ISO Standard C(即 C89)。目前国际上流行的 C 编译系统都是以这个标准为基础的。

1995 年又为 C 语言增加了一些新的函数,使之具有 C++ 的一些特征,使 C89 成为 C++ 的子集。1999 年推出的 C99 在保留 C 语言特征的基础上,增加了一系列面向对象的新特征,C 语言也从面向过程的语言发展成为面向对象的语言。

C 语言已经成为全球程序员所使用的公共编程语言,并由此产生了当前程序设计的两个主流语言: C++ 和 Java——它们都是建立在 C 语言的语法和基本结构基础上的,而且现在世界上的许多软件都是在 C 语言及其衍生的各种语言的基础上开发而形成的。所以 C 语言是学习 C++ 和 Java 语言的基础语言,每一名大学生都应该努力学会用 C 语言编写程序。

1.2.2 C 语言的特点

C 语言之所以能成为目前世界上最流行的高级程序设计语言,主要是因为它具有以下诸多的特点。

1. 语言简洁

C 语言一共只有 32 个关键字和 9 种控制语句。现将 C 语言与 PASCAL 语言做以下比较(如表 1-1 所示)。

表 1-1 C 语言与 PASCAL 语言的对比

C 语言	PASCAL 语言	含 义
{ }	BEGIN...END	复合语句
if (e) S;	IF(e) THEN S	条件语句
int i;	VAR i: INTEGER;	定义 i 为整型变量
int a[10];	VAR a: ARRAY[1...10] OF INTEGER;	定义 a 为整型一维数组
int *p;	VAR p: ↑ INTEGER;	定义 p 为指向整型变量的指针变量
i += 2;	i := i+2	赋值语句,使 i+2→i
i++, ++i;	i := i+1	i 自增 1, i+1→i

2. 数据类型丰富

C语言包含了多种数据类型,如整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型,由这些数据类型还可以构造更复杂的数据类型和数据结构。

3. 表达能力强

C语言拥有丰富的运算符,共34个,其中括号、逗号、赋值等都作为运算符处理,使得表达式类型多种多样。

4. 结构化

C语言能提供多种数据结构(如链表、树、堆栈、图等)和控制结构(如if()—else语句、for()—语句、while()—语句、do-while()语句等),C语言是以函数形式提供给用户的,这些函数可方便地调用,并具有多种循环、条件语句来控制程序流向,从而使程序完全结构化。

5. 可移植性好

在C语言所提供的语句中,没有直接依赖于硬件的语句。与硬件有关的操作,如数据的输入、输出等都是通过调用系统提供的库函数来实现的,而这些库函数本身并不是C语言的组成部分。因此,用C语言编写的程序能够很容易地从一种计算机环境移植到另一种计算机环境中。

1.3 C语言的程序结构

1.3.1 简单的C语言程序

【例1.1】 编写一个C语言程序显示字符串“Good morning!”。

```
#include <stdio.h>           /* 第1行 */
main()
{
    printf("Good morning! \n"); /* 第4行 */
}
```

执行上述程序后在屏幕上将显示:

Good morning!

第1行中头文件stdio.h包含了输出函数printf的函数声明。若要使用输入、输出函数,则必须包含头文件stdio.h。

第2行至第5行定义了一个主函数(main函数)。第4行是一个输出语句,其功能是在屏幕上显示双引号内的字符串“Good morning!”。

【例1.2】 随机输入两个整数,求两者的最小值,并输出这个最小值。

```
#include <stdio.h>           /* 第1行 */
main()
{
    int x,y,z;                /* 第4行 */
    printf("enter two numbers:"); /* 第5行 */
    scanf("%d_%d", &x, &y);    /* 第6行 */
}
```

```
if(x<y)                                /* 第 7 行 */
    z=x;
else
    z=y;
printf("The min is %d\n", z);
}
```

运行结果:

```
enter two numbers: 32 37
```

```
The min is 32
```

第 4 行定义了 3 个变量,均为整数类型。在 C 语言中所有的变量在引用前都必须进行定义,说明其类型,即变量必须先定义再使用。

第 5 行是输出语句,它的作用是向终端输出数据。

第 6 行是输入语句,“%d %d”是格式符,要求从键盘上输入两个整数。

第 7 行到第 10 行是 C 语言中的 if-else 语句。

第 7 行中的“ $x < y$ ”是 if 语句的判断条件,当条件满足时(即 x 小于 y),执行第 8 行“ $z = x$ ”;否则(即 x 不小于 y)执行第 10 行“ $z = y$ ”。

1.3.2 C 语言程序的特点

通过上一节的几个例子可以看出 C 语言程序有以下几个特点:

1. C 语言程序由函数构成

一个 C 语言程序由多个函数组成,其中必须包含一个且只能包含一个 main 函数,程序总是从 main 函数开始执行。

2. 函数由两个部分构成

(1) 函数首部,即函数的第一行,包括函数名、函数类型、函数参数及参数类型。

(2) 函数体,即函数首部下面的大括号{}内的部分。如果一个函数内有多个大括号,则最外层的一对{}为函数体的范围。函数体包括声明部分及执行部分。在声明部分,可以定义所用到的变量,对所调用的函数进行声明。执行部分由若干个语句组成。在特殊情况下也可以没有声明部分,甚至可以既无声明部分也无执行部分。

3. 书写格式自由

C 语言程序的书写格式比较自由,一行可以写几个语句,一个语句也可以分写在多行上。如:

```
int a,b,c=3,d=4;
```

也可以写成:

```
int a,b,
c=3,d=4;
```

4. 语句和变量声明必须以“;”结束

每个语句和数据定义的最后必须有一个分号,分号是 C 语言的必要组成部分。如:

```
z=x+y;
```

分号必不可少,即使是程序中最后一个语句也应包含分号。

5. 程序中使用注释

可以用/*...*/对程序中的任何一部分作注释。一个好的、有使用价值的源程序都应加上必要的注释,以增加程序的可读性。

6. 变量必须先定义后使用

C语言程序中,在使用每一个变量前都必须先定义其相应的数据类型。若变量在使用前没有定义,则程序在进行编译连接时,系统会提示“程序中的变量未被定义”的字样。

1.3.3 C语言书写规则

1. 一般情况下,一个说明或一个语句占一行。

2. 用{ }括起来的部分,通常表示程序的某一层结构。{ }一般与该结构语句的第一个字母对齐,并单独占一行。

3. 低一层次的语句可比高一层次的语句或说明缩进若干格后书写。这样看起来会更加清晰,增加了程序的易读性。

在编写程序时应力求遵循这些规则,以养成良好的编程习惯。

1.3.4 C语言字符集

字符是组成语言的最基本的元素。C语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

1. 字母

小写字母a~z共26个,大写字母A~Z共26个。

2. 数字

0~9共10个。

3. 空白符

空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时,只起间隔作用,编译程序对它们忽略。因此在程序中使用空白符与否,对程序的编译不产生影响,但在程序中适当的地方使用空白符可以增加程序的清晰性和可读性。

4. 标点符号

‘ “ ; :

5. 特殊符号

\ _ \$ #

1.4 上机步骤

要运行一个C程序,达到预期的目的,需要经过以下几个步骤:

1. 程序的编辑

在上机操作之前先编写好C源程序,然后借用一种编辑软件(编辑器)将编写好的C程序输入计算机,并以文本文件的形式保存到计算机磁盘上。编辑的结果是建立一个C源程序文件,该文件的后缀名为“.C”(一般由编辑器自动加上)。

C 程序习惯上使用小写英文字母,常量和和其他用途的符号可用大写字母。C 语言中大、小写字母是有区别的,关键字必须小写。

2. 程序的编译

编译是指将编辑好的源文件翻译成二进制目标代码的过程。编译过程是由编译程序(编译器)完成的。不同操作系统下的各种编译器的使用命令不完全相同,使用时应注意计算机环境。编译时,编译器首先要对源程序中的每一个语句检查语法错误,当发现错误时,就在屏幕上显示错误的位置和错误类型的信息。此时,要再次调用编辑器进行查错修改。然后,再进行编译,直至排除所有语法和语义错误。正确的源程序文件经过编译后在磁盘上生成目标文件(后缀名为 .obj 的文件)。

3. 程序的连接

编译后产生的目标文件是可重定位的程序模块,不能直接运行。连接就是把目标文件和其他必需的目标程序模块(这些目标程序模块是分别编译而生成的目标文件)以及系统提供的标准库函数连接在一起,生成可以运行的可执行文件的过程。连接过程使用 C 语言提供的连接程序(连接器)完成,生成的可执行文件(后缀名为 .exe 的文件)存到磁盘中。

4. 程序的运行

程序经过连接生成可执行文件后,就可以在操作系统控制下运行。若执行程序后达到预期目的,则 C 程序的开发工作到此完成。否则,要进一步检查修改源程序,重复编辑—编译—连接—运行的过程,直到取得预期结果为止。

整个流程图如图 1-1 所示。

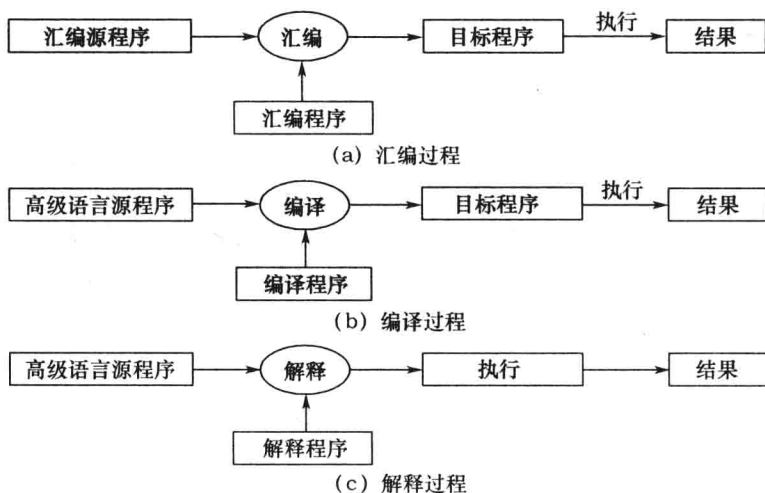
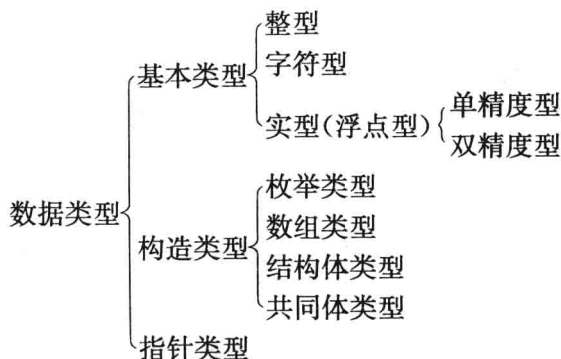


图 1-1 汇编、编译与解释的过程

第2章 C 数据及其类型

C 语言的数据类型非常丰富,包括:



本章主要介绍 C 语言的三种基本数据类型:整型、浮点型和字符型。

2.1 常量和变量

2.1.1 常量

常量又称常数,是指在程序运行过程中其值不能被改变的量。常量按其属性可分为不同的类型。

1. 整型常量:如 21, -5, 9 等。
2. 实型常量:如 3. 2, -0. 14, 3. 1415 等。
3. 字符常量:如 'a', 'A', 在 C 语言中,将字母 a 和 A 用单引号引起即分别代表两个不同的字符常量。

值得指出的是,在 C 语言中,也可以用一个标识符代表一个常量,即符号常量。

【例2.1】 计算圆的面积。

```
#define PI 3.1415
#include <stdio.h>
main ()
{
    int R;
    float S;
    printf ("Enter R:");
    scanf ("%d", &R);
    S=PI * R * R;
    printf ("%f", S);
}
```

3.1415 是一个实型常量,在本程序中运用 `#define` 命令行定义了一个标识符常量 `PI`,来代替常量 3.1415。这样,凡是在程序中有 `PI` 出现的地方其值均为 3.1415。再者,在 C 语言中乘号是不能省略的,如本程序中 $S=PI * R * R$,不能写成 $S=PIR^2$ 或写成 $S=PI \times R^2$ 的。C 语言中规定:星号“`*`”代表乘号“ $\times$ ”,为的是避免将乘号“ $\times$ ”和字母 `X` 混淆。 R^2 写成 $R * R$,因为在 C 语言表达式中无法表示右上角标。

2.1.2 变量

变量是指在程序运行过程中,其值可以被改变的量。每一个变量都有一个名字,并根据其类型不同在计算机内存中分配到一定的存储单元。该存储单元用来存放变量的值,如图 2-1 所示。

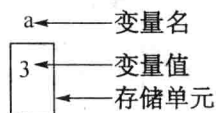


图 2-1 变量存储单元

注意区分变量名和变量值。两者之间既有联系但又有区别,变量名是一个符号地址,在程序运行过程中每一个变量名都会被分配到一个内存地址,变量值则是被存放在该存储单元中的数据。在程序运行过程中从变量中调用其值,其实是通过变量名找到相应的内存地址,在其存储单元之中读取数据。

在程序设计语言中,变量是以名字来标识的,在编程过程中,经常要用到标识符这一概念。标识符是指编程过程中用到的变量名、符号常量名、数组名、函数名、类型名、文件名的有效字符序列。

在 C 语言中规定,标识符只能由字母(包括大小写)、数字和下划线组成,且第一个字符只能是字母或下划线。以下均为合法的标识符和变量名:

`name`, `age`, `year`, `day`, `A1`, `_3_6_9`, `Day`, `li-ming`

在以上标识符中,`Day` 和 `day` 是两个不同的变量名。C 语言中字母区分大小写,这一点是和其他高级语言最大的区别。

理论上,标识符的长度是不受限制的,但实际上,各种程序设计语言都规定了一个有效长度。C 编译系统也有自己的规定,如 IBM PC MSC 的有效标识符长度为 8 个字符,如果程序中出现的变量名长度大于 8 时,只有前面 8 个字符有效。例如,`student_sex` 和 `student_sum` 因前 8 个字符相同,所以这两个变量是难以区分的。事实上,这时把变量改写成 `stu_sex` 和 `stu_sum` 即可。在 Turbo C 系统中允许 32 个字符,因此,在进行程序设计前应首先了解所使用的编译系统对变量名长度的规定,以免造成变量使用上的混乱。

此外,C 语言中的关键字不能用作变量名。关键字是 C 语言中预先定义的符号,它们都有固定的含义,可以直接使用他们来命名 C 语言中的语句。

在 C 语言中,要求对所有的变量做到“先定义,再使用”,即在调用每个变量前都必须明确说明其变量名和类型,例如将 `x` 定义成整型,语句如下:

```
int x;
```

若多个变量在同一条语句中定义,则变量名之间应用逗号隔开,例如将 x, y, z 定义成浮点型,语句如下:

```
float x,y,z;
```

变量在调用前必须先定义,这样可以避免变量名的录入错误,请看下列程序:

【例 2.2】

```
#include <stdio.h>
main ()
{
    int a,b;
    a=3,b=4;
    s=a+b;
    printf("%d",s);
}
```

以上程序在编译时会检查出 s 未被定义,系统输出“变量 s 未被声明”的信息,便于用户发现错误,及时改正。

2.2 数据类型

数据是计算机处理的对象,计算机能够处理哪些数据呢?

提到数据,整数和实数是大家最为熟悉的,它们是计算机能处理的两种最基本的数据,在日常生活中能用整数和实数描述的数据有很多。但也存在许多数据对象,难以用整数和实数进行描述,例如:人的姓名、性别、民族、婚否等。性别的描述是“男”或“女”,可用一个字符来表示;姓名要用一串字符来表示;民族的描述就更多,它可以是汉、满、蒙、回、苗等。更为复杂的数据对象可能由若干称为成分的数据共同描述。在现实生活中,还有许多事件是难以用现在的数据进行描述的,因此,我们必须扩充数据的概念。

数据必须有类型,不同类型的数据,值的存储结果、值的解释以及值的运算方式也不同,C语言中提供了四种基本类型:整型数据、实型数据、字符型、枚举型。

2.2.1 整型数据

1. 整数类型

整数类型简称整型,用标准类型 `int` 来表示。整型数据的值域包括正、负整数和零,由有限多个连续的整数组成,值域大小与机器有关。

在C语言中,根据占用内存的字节数不同,整数类型可分为三种:

- (1) 整型:用 `int` 表示,如 $-33, 21, 0$ 。
- (2) 短整型:用 `short int`(或用 `short`)表示。
- (3) 长整型:用 `long int`(或用 `long`)表示。

数据类型在内存中所占位数与机器有关,对于PC机,各种类型数据所占位数和数值范围如表2-1所示。

表 2-1 各种整型数据所占位数和数的范围

数据类型	所占位数	数的范围
int	16	-32768~32767
short	16	-32768~32767
long	32	-2147483648~2147483647

2. 整型变量

整型变量可存放整型数据(即不带小数点的数)。在 Turbo C 中,一个整型变量的值的范围为-32768~32767。在实际应用中,为了扩充数的描述范围,可以在变量定义前进行限定,如表 2-2 所示。

表 2-2 ANSI 标准定义的整数类型

类 型	所占位数	最小取值范围
[signed] int	16	-32768~32767, 即 $-2^{15} \sim (2^{15}-1)$
unsigned int	16	0~65535, 即 $0 \sim (2^{16}-1)$
[signed] short [int]	16	-32768~32767, 即 $-2^{15} \sim (2^{15}-1)$
unsigned short [int]	16	0~65535, 即 $0 \sim (2^{16}-1)$
long [int]	32	-2147483648~2147483647, 即 $-2^{31} \sim 2^{31}-1$
unsigned long [int]	32	0~4294967295, 即 $0 \sim (2^{32}-1)$

在表 2-2 中[singed]int 表示有符号基本整型,unsigned int 表示无符号基本整型,在整型变量的说明符 int 的前面加上修饰符:signed、unsigned、long 和 short 后,就可以说明一个变量是带符号的、无符号的、长整型的和短整型的。

对于整型变量来说,有如下几点要注意:

- (1) 如果在定义一个整型变量时含有修饰符 signed、unsigned、long 和 short 等,那么 int 可以省略不写。例如:long int y; 与 long y;所说明的变量含义相同。
- (2) int 前面没有修饰符时,默认为是带符号的。即 int 就是 signed int。
- (3) signed int 与 unsigned int 的区别在于对数(二进制)的最高位的解释不同。对于前者,把最高位当作符号位看待;对于后者,最高位仍用于存储数据。

3. 整型常量

整型常量即通常提到的整常数。在 C 语言中,整型常量可以用三种形式表示。

- (1) 八进制整型常量:八进制整型常量以 0 开头,如 0123 表示八进制的 123,其值为 $1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$,等于十进制数 83。八进制数的每个数字位可以是 0~7。
- (2) 十进制整型常量:如 123,132,-21 等,其每个数字位可以是 0~9。
- (3) 十六进制整型常量:十六进制整型常量以 0x 或 0X 开头,如 0x123 表示十六进制数 123,其值为 $1 \times 16^2 + 2 \times 16^1 + 3 \times 16^0$,等于十进制数 291。十六进制数的每个数字位可以是 0~9,A~F。

2.2.2 实型数据

1. 实型类型