

普通高等教育“十二五”创新型规划教材

# C语言 程序设计 与 实验指导

主编 冯志杰 侯清兰

 北京理工大学出版社  
BEIJING INSTITUTE OF TECHNOLOGY PRESS

## 内 容 简 介

本书较为全面地介绍了C语言程序设计的基本语法、程序设计的基本思想及传统的结构化程序设计的一般方法;介绍了全国计算机二级考试公共基础知识;介绍了C语言程序设计实验。全书共分15章,第1~10章为C语言程序设计篇,第11~14章为全国计算机二级考试公共基础知识篇,第15章为C语言程序设计实验篇。

本书结构清晰、内容翔实、深入浅出、注重实用、易学易用,可作为高等院校工科专业的教材使用,也可供从事计算机技术的工程技术人员学习参考。

版权专有 侵权必究

---

### 图书在版编目(CIP)数据

C语言程序设计与实验指导/冯志杰,侯清兰主编. —北京:北京理工大学出版社,2012.8

ISBN 978-7-5640-6642-0

I. ①C… II. ①冯…②侯… III. ①C语言-程序设计-高等学校-教材  
IV. ①TP312

中国版本图书馆CIP数据核字(2012)第192688号

---

出版发行/北京理工大学出版社

社 址/北京市海淀区中关村南大街5号

邮 编/100081

电 话/(010)68914775(办公室) 68944990(批销中心) 68911084(读者服务部)

网 址/http://www.bitpress.com.cn

经 销/全国各地新华书店

印 刷/北京慧美印刷有限公司

开 本/787毫米×1092毫米 1/16

印 张/18.25

字 数/420千字

版 次/2012年8月第1版 2012年8月第1次印刷

责任编辑/高芳

印 数/1~4000册

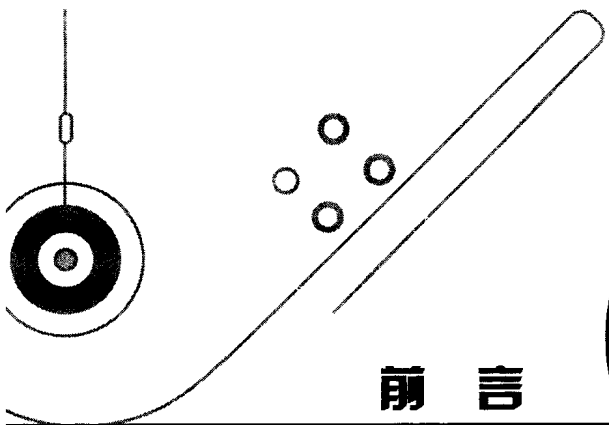
责任校对/周瑞红

定 价/39.00元

责任印制/王美丽

---

图书出现印装质量问题,本社负责调换



# C 语言程序设计 与实验指导

## 前 言

《C 语言程序设计与实验指导》是计算机及相关专业的专业基础课程之一，C 语言简洁高效、结构丰富，有良好的结构化语言，可移植性强，生成代码质量高，既可以用来编写系统软件，也可以用来编写应用软件，是工科学生学习计算机语言的首选课程，也是学习其他高级语言的基础。该课程在我校已开设多年，积累了较丰富的教学经验。2010 年该课程被评为青岛滨海学院校级精品课程，2012 年该课程参加山东省省级精品课程评选。

通过本课程的学习，掌握 C 语言的基本语法、程序设计的基本思想及传统的结构化程序设计的一般方法。以 C 语言为基础，培养严谨的程序设计思想、灵活的思维方式及较强的动手能力。并以此为基础，逐渐掌握复杂软件的设计和开发手段，为后续专业课程的学习打下扎实的理论和实践基础。

本教材根据教育部考试中心制定的《全国计算机等级考试大纲》对二级 C 语言的考试范围要求，组织有多年教授 C 语言教学经验的老师编写而成。本教材主要分三部分：第一部分是 C 语言程序设计，包括 C 语言的各种数据类型、运算符、各种表达式、语句结构、函数、指针、结构体和共同体、文件等内容。第二部分为全国计算机二级考试公共基础知识，包括数据结构与算法、程序设计基础、软件工程、数据库原理等内容。第三部分为 C 语言程序设计实验，包括 16 个实验。本教材内容精炼、体系合理、逻辑性强、例题和习题丰富、讲解通俗易懂，理论与实践相结合，既可以作为高等院校工科专业的教材，也可以作为自学教材使用。

根据计算机科学技术的发展和教学实践的需求，课程组不断对该书进行修订，作为内部教材已试用两轮，现正式出版。课程组希望尽己所能力求最佳，但毕竟水平和精力有限，最终的书稿中肯定存在一些不妥甚至错误之处，敬请广大读者批评指正。

编者

# 目录

## 第1章 C语言程序设计基础 ..... 001

- 1.1 简单C语言程序的构成和格式 ..... 001
- 1.2 C语言的特点 ..... 002
- 1.3 练习题 ..... 003

## 第2章 数据类型与运算 ..... 005

- 2.1 常量与变量 ..... 005
  - 2.1.1 常量与符号常量 ..... 005
  - 2.1.2 变量 ..... 006
- 2.2 整型数据 ..... 006
  - 2.2.1 整型常量 ..... 006
  - 2.2.2 整型变量 ..... 008
  - 2.2.3 整型数据的分类 ..... 008
- 2.3 实型数据 ..... 009
  - 2.3.1 实型常量 ..... 009
  - 2.3.2 实型变量 ..... 009
- 2.4 字符型数据 ..... 009
  - 2.4.1 字符常量 ..... 009
  - 2.4.2 字符变量 ..... 011
- 2.5 各种数据类型间的混合运算 ..... 011
- 2.6 算数运算符和算数表达式 ..... 012
  - 2.6.1 基本算术运算符和算术表达式 ..... 012
  - 2.6.2 算术运算符的优先级和结合性 ..... 013
  - 2.6.3 强制类型转换运算符 ..... 013
  - 2.6.4 自加、自减运算符 ..... 013
- 2.7 赋值运算符和赋值表达式 ..... 014
  - 2.7.1 赋值运算符和赋值表达式 ..... 014
  - 2.7.2 复合的赋值运算符 ..... 014
  - 2.7.3 赋值运算中的类型转换 ..... 015
- 2.8 关系运算和逻辑运算 ..... 015
  - 2.8.1 关系运算符和关系表达式 ..... 015
  - 2.8.2 逻辑运算符和逻辑表达式 ..... 016
- 2.9 逗号运算符和逗号表达式 ..... 017
- 2.10 位运算 ..... 017
- 2.11 练习题 ..... 018

## 第3章 顺序结构 ..... 021

- 3.1 C语句概述 ..... 021
- 3.2 格式数据输入输出 ..... 022

3.2.1	printf 函数 .....	022
3.2.2	scanf 函数 .....	025
3.3	字符数据输入输出 .....	027
3.3.1	putchar 函数 .....	027
3.3.2	getchar 函数 .....	028
3.4	程序举例 .....	028
3.5	练习题 .....	030
<b>第 4 章</b>	<b>选择结构 .....</b>	<b>034</b>
● 4.1	if 语句 .....	034
4.1.1	if 语句的三种基本形式 .....	034
4.1.2	嵌套的 if 语句 .....	037
4.2	switch 语句 .....	040
4.3	条件运算符和条件表达式 .....	042
4.4	练习题 .....	043
<b>第 5 章</b>	<b>循环结构 .....</b>	<b>048</b>
● 5.1	用 while 语句构成循环 .....	048
5.2	用 do-while 语句构成循环 .....	050
5.3	用 for 语句构成循环 .....	052
5.4	循环的嵌套 .....	055
5.5	break 语句和 continue 语句 .....	057
5.5.1	break 语句 .....	057
5.5.2	continue 语句 .....	058
5.6	练习题 .....	060
<b>第 6 章</b>	<b>数组与字符串 .....</b>	<b>065</b>
● 6.1	一维数组 .....	065
6.2	二维数组 .....	067
6.3	字符数组 .....	070
6.3.1	字符数组的定义和初始化 .....	070
6.3.2	字符数组输入和输出 .....	070
6.4	字符串处理函数 .....	072
6.5	练习题 .....	074
<b>第 7 章</b>	<b>函 数 .....</b>	<b>081</b>
● 7.1	函数的定义和返回值 .....	081
7.2	函数的调用 .....	082
7.2.1	函数调用的形式 .....	082
7.2.2	对被调用函数的说明 .....	083
7.2.3	函数间变量作参数的传递 .....	084

7.2.4 函数的嵌套调用和递归调用 .....	086
7.3 函数间数组做参数的传递 .....	087
7.3.1 数组元素作函数实参 .....	087
7.3.2 数组名作函数实参 .....	087
7.4 局部变量和全局变量 .....	090
7.4.1 局部变量 .....	090
7.4.2 全局变量 .....	091
7.5 变量的存储类别 .....	092
7.5.1 局部变量的存储类别 .....	092
7.5.2 全局变量的存储类别 .....	093
7.6 编译预处理 .....	094
7.6.1 宏定义和调用 .....	094
7.6.2 文件包含 .....	095
7.7 练习题 .....	096
<b>第8章 指 针</b> .....	<b>104</b>
8.1 指针和指针变量的概念 .....	104
8.2 用指针访问变量 .....	104
8.2.1 指针变量的定义、赋值 .....	104
8.2.2 指针变量的引用 .....	105
8.3 数组与指针 .....	106
8.3.1 一维数组与指针 .....	106
8.3.2 二维数组与指针 .....	108
8.4 字符串与指针 .....	108
8.5 指针作函数参数 .....	110
8.5.1 指针变量作函数参数 .....	110
8.5.2 数组名作函数参数 .....	111
8.5.3 字符指针作函数参数 .....	112
8.6 返回指针值的函数 .....	114
8.7 指针数组和指向指针的指针 .....	114
8.7.1 指针数组 .....	114
8.7.2 指向指针的指针 .....	115
8.8 关于函数的进一步讨论 .....	116
8.8.1 main( )函数的参数 .....	116
8.8.2 指向函数指针变量的定义 .....	116
8.9 练习题 .....	117
<b>第9章 结构体与共用体</b> .....	<b>123</b>
9.1 用 typedef 定义新类型 .....	123
9.2 结构体类型 .....	124
9.2.1 结构体类型的定义 .....	124
9.2.2 结构体变量定义、成员引用和初始化 .....	125

9.2.3	结构体数组的定义、初始化和引用举例 .....	128
9.2.4	结构体指针变量 .....	130
9.2.5	结构体在函数内的传递 .....	131
9.2.6	用结构体构成链表 .....	134
9.3	共用体类型 .....	137
9.3.1	共用体变量的定义 .....	137
9.3.2	共用体变量的成员引用 .....	137
9.3.3	共用体类型数据的特点 .....	137
9.4	练习题 .....	138
<b>第10章 文件</b> .....		<b>142</b>
10.1	C语言文件的概念 .....	142
10.2	文件类型指针 .....	142
10.3	文件的打开和关闭 .....	143
10.3.1	文件的打开(fopen 函数) .....	143
10.3.2	文件的关闭(fclose 函数) .....	144
10.4	文件的读写 .....	144
10.4.1	fgetc 函数和 fputc 函数 .....	145
10.4.2	fgets 函数和 fputs 函数 .....	146
10.4.3	fread 函数和 fwrite 函数 .....	147
10.4.4	fscanf 和 fprintf 函数 .....	148
10.5	文件的定位与检测 .....	149
10.5.1	文件的定位 .....	149
10.5.2	文件的检测函数 .....	151
10.6	练习题 .....	151
<b>第11章 数据结构与算法</b> .....		<b>154</b>
11.1	算法与数据结构的基本概念 .....	154
11.1.1	算法的描述与分析 .....	154
11.1.2	数据结构的基本概念 .....	156
11.2	线性表 .....	156
11.2.1	线性表的顺序存储结构 .....	157
11.2.2	线性表的链式存储结构 .....	158
11.3	栈和队列 .....	158
11.3.1	栈及其基本运算 .....	158
11.3.2	队列及其基本运算 .....	159
11.4	二叉树 .....	161
11.4.1	二叉树的基本概念 .....	161
11.4.2	二叉树的性质 .....	163
11.4.3	二叉树的遍历 .....	164
11.5	查找与排序 .....	165
11.5.1	查找技术 .....	165
11.5.2	排序技术 .....	166

11.6 练习题 .....	169
----------------	-----

## 第 12 章 程序设计基础 .....

12.1 结构化程序设计 .....	171
12.1.1 结构化程序设计的原则和方法 .....	171
12.1.2 结构化程序的基本结构与特点 .....	172
12.2 面向对象程序设计 .....	173
12.2.1 关于面向对象方法 .....	173
12.2.2 面向对象方法的基本概念 .....	174
12.3 练习题 .....	176

## 第 13 章 软件工程 .....

13.1 软件工程基本概念 .....	177
13.1.1 软件的定义与软件生命周期 .....	177
13.1.2 软件开发工具与软件开发环境 .....	181
13.2 结构化分析方法和设计方法 .....	182
13.2.1 结构化分析方法 .....	182
13.2.2 结构化设计方法 .....	184
13.3 软件测试 .....	188
13.3.1 软件测试的目的和准则 .....	188
13.3.2 软件测试技术与方法 .....	189
13.3.3 软件测试的过程 .....	193
13.4 程序的调试 .....	194
13.5 练习题 .....	196

## 第 14 章 数据库 .....

14.1 数据库的基本概念 .....	198
14.1.1 数据库的基本概念 .....	198
14.1.2 数据库系统的基本特点 .....	200
14.2 数据模型 .....	201
14.2.1 数据模型的基本概念 .....	201
14.2.2 数据模型的组成要素 .....	202
14.2.3 E-R 模型 .....	202
14.2.4 层次模型 .....	204
14.2.5 网状模型 .....	205
14.2.6 关系模型 .....	205
14.3 关系代数 .....	208
14.3.1 传统的集合运算 .....	209
14.3.2 专门的关系运算 .....	210
14.4 数据库设计方法及步骤 .....	213
14.4.1 数据库设计方法 .....	214
14.4.2 数据库设计步骤 .....	215



14.5 练习题 .....	216
----------------	-----

## 第 15 章 C 语言上机指导 .....

15.1 实验一:熟悉 C 语言的运行环境 .....	219
实验目的 .....	219
实验要求 .....	219
15.2 实验二:数据类型和运算符的运用 .....	221
实验目的 .....	221
实验要求 .....	222
15.3 实验三:格式输入输出 .....	223
实验目的 .....	223
实验要求 .....	224
15.4 实验四:选择语句的应用 .....	225
实验目的 .....	225
实验要求 .....	225
15.5 实验五: while 和 do... while 语句的应用 .....	227
实验目的 .....	227
实验要求 .....	227
15.6 实验六:for 语句和嵌套循环语句的应用 .....	229
实验目的 .....	229
实验要求 .....	230
15.7 实验七:一维数组的应用 .....	232
实验目的 .....	232
实验要求 .....	232
15.8 实验八:二维数组的应用 .....	234
实验目的 .....	234
实验要求 .....	235
15.9 实验九:字符数组的应用 .....	238
实验目的 .....	238
实验要求 .....	238
15.10 实验十:函数的定义和调用 .....	241
实验目的 .....	241
实验要求 .....	241
15.11 实验十一:数组作为函数参数 .....	244
实验目的 .....	244
实验要求 .....	244
15.12 实验十二:指针变量的定义、数组和指针 .....	249
实验目的 .....	249
实验要求 .....	249
15.13 实验十三:指针作函数参数 .....	252
实验目的 .....	252
实验要求 .....	252
15.14 实验十四:结构体的应用 .....	257

实验目的 .....	257
实验要求 .....	257
<b>15.15 实验十五:用指针处理链表 .....</b>	<b>261</b>
实验目的 .....	261
实验要求 .....	261
<b>15.16 实验十六:文件的应用 .....</b>	<b>267</b>
实验目的 .....	267
实验要求 .....	267
附录 1 C 语言中的关键字 .....	271
附录 2 C 语言中的运算符及优先级 .....	272
附录 3 常用字符与 ASC II 码对照表 .....	274
附录 4 库函数 .....	275

## C 语言程序设计基础

### 学习目标

学习 C 语言的构成和格式,了解 C 语言的优缺点。

### 学习要求

- 掌握:C 语言的构成和格式。
- 了解:C 语言的优缺点。



### 1.1 简单 C 语言程序的构成和格式

为了解 C 语言程序的构成和编写格式,下面先看三个简单的 C 程序例子。

例 1.1 在屏幕上显示“This is a C Program”。

```
#include <stdio. h >           /* stdio. h 是标准输入/输出头文件 */
void main( )                   /* main 是 C 语言的主函数 */
{                               /* 用花括号“{ }”括起来的部分为函数体 */
    printf("This is a C Program"); /* printf 是格式输出函数 */
}
```

执行以上程序后的输出结果为:

This is a C Program

例 1.2 求矩形的面积。

程序如下:

```
#include <stdio. h >
void main( )
{   float a, b, area;
    a = 1.2;           /* 将矩形的两条边长分别赋给 a 和 b */
    b = 3.6;
    area = a * b;     /* 计算矩形的面积并存储到变量 area 中 */
    printf(" a = %f, b = %f, area = %f\n", a, b, area); /* 输出矩形的边长和面积 */
}
```

执行以上程序后的输出结果为:

a = 1. 200000, b = 3. 600000, area = 4. 320000

以上两个程序都是完整的 C 程序,在代码编写完成之后,生成的源程序文件后缀名为.c。需要经过编译、连接、执行三个步骤才可看到程序结果。编译是将高级程序设计语言编写的源程序翻译成二进制形式的“目标程序”,后缀名为.obj。连接是将该目标程序与系统的函数库以及其他

目标程序连接起来,形成可执行的程序,后缀名为 .exe。最后执行就是执行上述两个步骤生成的可执行程序,将结果输出显示在屏幕上。

程序中的 `#include <stdio.h>` 通常称为命令行,是一条编译预处理命令,它的作用是通知 C 语言编译系统在 C 程序进行正式编译之前应该做一些预处理工作。

`stdio.h` 是系统提供的头文件,该文件中包含着有关输入输出函数的说明信息。

`main` 是主函数名。C 语言规定必须用 `main` 作为主函数名,是程序的“入口”,`main()` 是程序执行的第一条语句。注意:在 C 程序中,主函数必须有且只能有一个,但可以包含任意多个不同名的函数。主函数的位置可以放在整个 C 程序中的任何位置,但 C 程序的执行始终都是从 `main` 函数开始执行。一个函数名后面必须跟一对圆括号。

`{ }` 括起来的部分称为函数体。函数体内部包括说明部分和执行部分。

`int` 和 `float` 是 C 程序的数据类型。这两个关键字的作用是向计算机系统提出请求,定义整型变量和浮点型变量,同时申请在内存中占用相应的内存空间。

`printf()` 是格式输出函数,它是系统的库函数(又称为标准函数)。这些库函数由系统开发商事先编写好并存放在系统文件中,该函数的作用是在屏幕光标的位置上输出数据。

`scanf()` 是格式输入函数,它也是系统的库函数。该函数是用来输入数据的函数,计算机执行到该函数时会停下来等待键盘上输入的数据。C 语言本身没有输入输出语句,输入输出操作都是由以上两个输入输出库函数来完成的。

每个 C 程序的语句后面必须有分号,分号的作用是表明语句到此结束。如果在编写 C 程序时忘记输入分号的话,那就犯了一个大错误。但需注意预处理命令 `#include <stdio.h>` 后面没有分号。

C 程序书写格式自由,一行内可以写几个语句,一个语句可以分写在多行上。

在编写程序时可以在程序中加入注释,以说明变量的含义、语句的作用和程序段的功能,从而帮助人们阅读和理解程序。因此,一个好的程序应该有详细的注释。在添加注释时,注释内容必须放在符号“`/*`”和“`*/`”之间。添加注释也可用符号“`/**`”。两者的区别是:“`/* ... */`”可以表示跨行的注释说明,而“`/**`”只能说明本行的内容为注释说明。



## 1.2 C 语言的特点

C 语言是一种通用性很强的结构化程序设计语言,它既可以用来编写系统软件,也可以用来编写应用软件。它具有丰富的运算符和数据类型,语言简单灵活,表达能力强。C 语言的主要特点如下。

### 1. 用 C 语言编写的程序非常简洁

C 语言简洁、紧凑,使用方便、灵活。C 语言只有 32 个关键字,9 种控制语句,程序主要由小写字母组成,书写格式自由,压缩了不必要的成分,相对其他计算机语言而言,其源程序较短,因此输入程序时工作量少。

### 2. 运算符非常丰富

C 语言的运算符包含的范围很广泛,共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理,从而使 C 语言的运算符类型极其丰富,表达式类型多样化。灵活使用各种 C 语言的运算符可以实现在其他高级语言中难以实现的运算。

### 3. 数据类型丰富

C 语言的数据类型丰富,具有现代化语言的各种数据类型。C 语言的数据类型有:整型、实型、

字符型、数组型、指针型、结构型、联合型和枚举型等。它们能用来实现各种复杂的数据结构。因此,C语言具有很强的数据处理能力。

#### 4. 具有结构化的控制语句

C语言是一种结构化程序设计语言,它具有结构化控制语句(if else、while、do while、switch、for等语句)。C语言用函数作为程序模块,以实现程序的模块化。因此,C语言十分有利于实现结构化、模块化程序设计。

#### 5. 允许直接访问物理内存

C语言既具有高级语言的特点,又具有低级语言的一些功能。它允许直接访问物理内存,能进行位(bit)运算,可以直接对硬件进行操作。

#### 6. C程序的可移植性好

C语言程序本身不依赖于机器硬件系统,从而便于在硬件结构不同的机种间和各种操作系统中实现程序的移植。

C语言的优点很多,但也有不足之处。C语言语法限制不太严格,程序设计时自由度大,对变量类型的使用比较灵活,允许程序编写者有较大的自由度,放宽了对语法的检查。为此,程序员应当仔细检查程序,以保证其正确性,而不要过分依赖C语言编译程序去查错。



## 1.3 练习题

### 1. 选择题

(1)请指出以下C程序的错误所在。

①#include stdio. h;

```
main(); /* main function */
```

```
float r,s;
```

```
r=5.0;
```

```
s=3.14159*r*r;
```

```
printf("%f\n",s)
```

②main

```
/* main function */
```

```
{ float a,b,c,v; /* a,b,c are sides,v is volume of cube */
```

```
 a=2.0;b=3.0;c=4.0
```

```
 v=a*b*c;
```

```
 printf("%f\n",v)
```

```
}
```

(2)以下叙述中正确的是( )。

(2008年9月二级C真题)

A)C程序的基本组成单位是语句

B)C程序中的每一行只能写一条语句

C)简单C语句必须以分号结束

D)C语句必须在一行内写完

(3)计算机能直接执行的程序是( )。

(2008年9月二级C真题)

A)源程序

B)目标程序

C)汇编程序

D)可执行程序

(4)以下叙述中正确的是( )。

(2008年3月二级C真题)

A)C程序中的注释只能出现在程序的开始位置和语句的后面。

B)C程序书写格式严格,要求一行内只能写一个语句。

C) C 程序书写格式自由,一个语句可以写在多行上。

D) 用 C 语言编写的程序只能放在一个程序文件中。

(5) 下列叙述中正确的是( )。

(2007 年 9 月二级 C 真题)

A) C 语言程序将从源程序中第一个函数开始执行。

B) 可以在程序中由用户指定任意一个函数作为主函数,程序将从此开始执行。

C) C 语言规定必须用 main 作为主函数名,程序将从此开始执行,在此结束。

D) main 可作为用户标识符,用以命名任意一个函数作为主函数。

(6) 一个 C 语言程序是由( )。

A) 一个主程序和若干子程序组成

B) 函数组成

C) 若干过程组成

D) 若干子程序组成

## 数据类型与运算

### 学习目标

了解 C 语言的基本数据类型,掌握 C 语言的基本运算符和表达式。

### 学习要求

- ▶ 了解:C 语言的基本数据类型。
- ▶ 掌握:C 语言的基本运算符和表达式。

数据类型是指数据在计算机内存中的表现形式。C 语言提供的数据类型及分类有如下几种:

- (1) 基本类型:包括整型、实型(浮点型)、字符型和枚举型 4 种。
- (2) 构造类型:包括数组类型、结构体类型和共用体类型 3 种。
- (3) 指针类型:是一种特殊的数据类型,其值用来表示某个变量在内存中的地址。
- (4) 空类型:空类型 void 用来声明函数的返回值类型为空,不能声明变量。

前三种类型的数据都有常量和变量之分,本章将介绍整型、实型和字符型三种基本的数据类型。

## 2.1 常量与变量

在程序运行过程中,其值不能被改变的量称为常量,其值可以改变的量称为变量。

### 2.1.1 常量与符号常量

C 语言中常用的常量主要有三类:整型常量、实型常量和字符常量。整型常量和实型常量又称为数值型常量,它们有正值和负值之分。如 12、-1、0 都是整型常量,3.1415926、-1.35、0.0 都是实型常量。字符常量是用单引号括起来的一个字符,如'a'、'A'等。这些都是字面上的常量,除此之外,C 语言中可以用一个符号名来代表一个常量,称为符号常量。

例 2.1 符号常量的使用。

```
/* 程序功能:计算圆的面积 */  
#include <stdio.h>  
#define PI 3.1415926  
void main( )  
{ float r,s;  
  r=5.0;  
  s=PI*r*r;  
  printf(" Area is %f", s);
```

程序中用#define 命令行定义 PI 代表常量 3.1415926, 此后凡在该程序中出现 PI 都代表 3.1415926, 可以和常量一样进行运算。

注意: 符号常量的值在其作用域内不能改变, 也不能再被赋值, 这一点要和变量区分开。

使用符号常量有以下好处。

- (1) 含义清楚。定义符号常量时应考虑“见名知意”。
- (2) 在需要改变一个常量时能做到“一改全改”。

## 2.1.2 变量

在程序运行过程中其值可以改变的量称为变量。程序中用到的所有变量都必须有一个名字作为标识。

变量的名字由用户自己定义, 它必须符合标识符的命名规则。C 语言中, 变量名、函数名、数组名等的命名都必须遵守一定的规则, 按此规则命名的符号称为标识符。合法标识符的命名规则是: 标识符可以由字母、数字和下划线组成, 并且第一个字符必须是字母或下划线。以下都是合法的标识符:

a、area、PI、a\_arr、\_ss

以下都是非法的标识符:

123、a-b、a&b、¥s

在 C 语言的标识符中, 大写字母和小写字母被认为是两个不同的字符, 例如 S 和 s 是两个不同的标识符。习惯上, 变量名用小写字母命名, 符号常量用大写字母命名。

一个变量实质上是代表了内存中的一个存储单元。在程序中, 定义了一个变量 a, 实际上是给用 a 命名的变量分配了一个存储单元, 用户对变量 a 进行的操作就是对该存储单元进行的操作; 给变量 a 赋值, 实质上就是把数据存入该变量所代表的存储单元中。

C 语言规定, 程序中所有变量都必须先定义后使用。

像常量一样, 变量也有整型变量、实型变量、字符型变量等不同类型。在定义变量的同时要说明其类型, 系统在编译时就能根据其类型为其分配相应的存储单元。



## 2.2 整型数据

### 2.2.1 整型常量

整型常量即整常数, 在 C 语言中有以下 3 种不同的表示形式。

- (1) 十进制整数: 由数字 1~9 开头, 其余各位由 0~9 组成。如 123、-789、0 等。
- (2) 八进制整数: 由数字 0 开头, 其余各位由 0~7 组成。在书写时要加前缀“0”(零)。如 012, 表示八进制数 12。
- (3) 十六进制整数: 由 0x 或 0X 开头, 其余各位由 0~9 与字母 a~f(0X 开头时输出为 A~F) 组成。在书写时要加前缀“0x”或“0X”。如 0x36, 代表十六进制数 36。

在计算机内部表示数据时是采用二进制, 二进制整数由数字 0 和 1 构成。如 011101 等。

在 C 语言中, 只有十进制数可以是负数, 八进制和十六进制数只能是正数。



二进制、八进制、十进制、十六进制之间可以相互转换。表 2.1 为 0~15 的十进制数与二进制、八进制、十六进制之间的转换关系。

表 2.1 各进制之间的转换关系

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0	0	0	8	1 000	10	8
1	1	1	1	9	1 001	11	9
2	10	2	2	10	1 010	12	A
3	11	3	3	11	1 011	13	B
4	100	4	4	12	1 100	14	C
5	101	5	5	13	1 101	15	D
6	110	6	6	14	1 110	16	E
7	111	7	7	15	1 111	17	F

### 1. 二进制、八进制、十六进制数转化为十进制数

对于任何一个二进制数、八进制数、十六进制数,都可以写出它的按权展开式,再进行计算即可。例如:

$$(1\ 111.11)_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = (15.75)_{10}$$

$$(A10B.8)_{16} = 10 \times 16^3 + 1 \times 16^2 + 0 \times 16^1 + 11 \times 16^0 + 8 \times 16^{-1} = (41\ 227.5)_{10}$$

### 2. 十进制数转换为二进制数

(1)对于整数部分采用除 2 取余法,即逐次除以 2,直至商为 0,得出的余数倒排,即为二进制各位的数码。

(2)小数部分采用乘 2 取整法,即逐次乘以 2,从每次乘积的整数部分得到二进制数各位的数码。

例:将十进制数 6.375 转换为二进制数。

除数	被除数	余数	0.375	
			×	2
	6	0	0.750	乘积无进位, 即 $a_{-1}=0$
2	3	1	0.500	乘积有进位, 即 $a_{-2}=1$
2	1	1	0.000	乘积有进位, 即 $a_{-3}=1$

故:  $(6.375)_{10} = (110.011)_2$

### 3. 二进制数转换为十六进制数

二进制数转换成十六进制数的方法是:将二进制数从小数点开始,对二进制整数部分向左每 4 位分成一组,对二进制小数部分向右每 4 位分成一组,不足四位的分别向高位或低位补 0 凑成四位。每一组有 4 位二进制数,分别转换成十六进制数中的一个数字,全部连接起来即可。

例:  $(1\ 101\ 010.110)_2 = (6A.C)_{16}$

### 4. 二进制数转换为八进制数

二进制数转换成八进制数的方法是:将二进制数从小数点开始,对二进制整数部分向左每 3 位分成一组,对二进制小数部分向右每 3 位分成一组,不足三位的分别向高位或低位补 0 凑成三位。