

全国计算机等级考试教育丛书

CX-2317 C语言程序设计

(二级)

李大友 主编

C语言程序设计 (二级)

李盘林 陈宪福 王旭 编著



版社

科学出版社
龍門書局

全国计算机等级考试教育丛书

C 语言程序设计

(二级)

李大友 主编

李盘林 陈宪福 王旭 编著

科学出版社

龙 门 书 局

1998

JS/40/23
内 容 简 介

本书是作者根据国家教委考试中心颁布的《全国计算机等级考试大纲》(1995)C 语言程序设计考试要求并结合多年的教学经验编写而成的,在内容编排上尽量体现出易学的特点,在文字叙述上条理清晰、简洁,便于读者阅读。

本书分十二章,按照循序渐进的原则,逐步地介绍 C 语言中的基本概念和语法规则;同时,通过典型的例题分析,着重强调了利用 C 语言进行程序设计的方法。

书末附有各章的习题解答,以及 1994 至 1997 历年来全国计算机等级考试 C 语言程序设计试题和答案,供读者学习之用,以利于全面、系统地对 C 语言程序设计的理解和掌握。

本书可以作为全国计算机等级考试用书和自学用书,也可以作为大专院校非计算机专业的教材或教学参考书。

欲购本书或欲得到有关本书技术支持的读者,请与北京海淀 8721 信箱书刊部联系,电话:010-62562329、010-62541992,或传真至 010-62561057。

全国计算机等级考试教育丛书
C 语言程序设计(二级)

李大友 主编

李益林 陈宪福 王 旭 编著

责任编辑:秦人华

科 学 出 版 社 出 版
北 京 科 学 局

北京东黄城根北街 16 号

邮政编码:100717

北京媛明印刷厂 印刷

新华书店北京发行所发行 各地新华书店经售

*

1998 年 3 月 第 一 版 开本:787×1092 1/16

1998 年 3 月 第一次印刷 印张:16 5/8

印数:1—5000

字数:376 000

ISBN 7-03-005861-5/TP·765

定价:20.00 元

第一届全国计算机等级考试 委员会成员名单

主任委员：

杨芙清

副主任委员：(以姓氏笔画为序)

朱三元 杨学为 罗晓沛 谭浩强

委员：(以姓氏笔画为序)

王义和 王申康 古天祥 齐治昌 仲萃豪 刘淦澄
刘瑞挺 李大友 李克洪 吴文虎 沈均毅 杨洪
杨明福 林卓然 施伯乐 钟津立 侯炳辉 俞瑞钊
张福炎 袁开榜 席先觉 唐兆亮 徐沪生 钱维民
潘桂明 鞠九滨 瞿坦

序 言

《全国计算机等级考试教育丛书》是按国家教委考试中心颁布的《全国计算机等级考试大纲(1995)》并兼顾考试发展趋势的要求编写的。

这套教育丛书不仅考虑了大纲的要求(DOS平台),而且兼顾当今计算机发展现状,对Windows平台下的应用,也做了全面考虑。

本套丛书首批共计三本:《C语言程序设计(二级)》、《数据库语言及其应用》和《二级等级考试试题及详解》。

《C语言程序设计(二级)》以结构化程序设计作为基础,全面介绍了C语言程序设计的基本方法,并且给出了大量应用实例,以及94~97历年来全国计算机等级考试C语言程序设计试题和答案。

《数据库语言及其应用》以DOS平台下的FoxBASE+和Windows平台下的FoxPro为依据,介绍了数据库语言的基本概念、基本操作技术和基本程序设计方法,各章末均附有习题和例题。

为了使读者更好地掌握丛书的内容,《二级等级考试试题及详解》收集了近年来全国各地的等级考试试题上千题,并对试题进行了分类,做了详解,从而使其不仅可作为在试题类型、深度、广度方面的参考用书,而且可以作为很好的学习教材。

为提高本套丛书的质量,我们聘请了高等院校教学经验丰富的专家教授编写。这套丛书结构合理、层次清晰、语言简练、概念准确、深入浅出、便于自学。

我们深信,这套丛书定会发挥出应有的作用。由于时间仓促,有不妥之处,还望得到各方面的批评指正。

主编 李大友

1997年3月3日

前 言

C 语言是一种短小精悍的计算机高级程序设计语言。它是根据结构化程序设计原则设计并实现的。C 语言具有丰富的数据类型。它为结构化程序设计提供了各种数据结构和控制结构。它能够实现汇编语言中的大部分功能。同时,用 C 语言编写的程序具有特别好的可移植性。

尽管当初 C 语言是为编写 UNIX 操作系统而设计的,但它并不依赖于 UNIX 操作系统。目前,C 语言能在多种操作系统环境下运行,并且已经在广泛的领域里得到了应用,是目前国际上应用最为普及的高级程序设计语言之一。

多年来,作者一直从事 C 语言的教学工作,同时也利用 C 语言来开发实际的课题。本书就是作者根据国家教委考试中心提出的《C 语言程序设计考试要求》并结合多年的教学经验和应用 C 语言的体会写成的。本书的主要特点可归纳如下:

- (1) 按照循序渐进的原则,逐步引出 C 语言中的基本概念。
- (2) 在文字叙述上力求条理清晰、简洁,以利于读者阅读。
- (3) 在讲解 C 语言中的基本概念时,除了阐述理论之外,还通过典型的例题,着重强调了基本概念在程序设计中的应用,以利于读者理解和掌握。
- (4) 本书的重点是放在 C 语言的使用上,书中没有深奥的理论和算法。在例题中出现的每一个算法,都给出了比较详细的解释。因此,特别适合于初学者和自学者使用。
- (5) 本书的每一章中都包括“应用举例”一节,其中的例题包含了本章讲解的主要内容,有些例题还具有一定的难度。通过阅读和分析这些例题,能使读者对本章内容的应用有一个全面的了解。
- (6) 每章的最后都附有一定量的习题,其中包括程序分析题和编程题,这些习题对于读者巩固已学习的内容是大有益处的。在附录 E 中给出了各章习题的详细解答,可供读者参考。
- (7) 在附录 F 中给出了 94 至 97 历年来全国计算机等级考试 C 语言程序设计试题及答案,供读者学习之用,以便读者系统、全面地理解和掌握 C 语言程序设计。

作者认为,要学好 C 语言,除了掌握 C 语言的基本理论之外,还必须加强实践环节。本书中的所有例题都在微机上(使用 Microsoft C)调试通过。希望读者边学习边上机实践,这样不仅可以加快学习进度,也能提高学习效率。

本书在编写过程中,一直得到李大友教授的支持和帮助,也得到了张华女士在文稿编辑中的大力协助,在此表示深深的谢意。

由于作者水平有限,经验不足,本书中一定有不少缺点和错误,敬请计算机工作者和广大读者批评指正。

编 者

一九九七年六月

• v •

目 录

第一章 绪言	(1)
1.1 C 语言的特点	(1)
1.2 C 语言程序的开发过程	(2)
1.3 简单的 C 语言程序	(3)
习题.....	(5)
第二章 数据类型、运算符及其表达式	(6)
2.1 常量和变量	(6)
2.2 基本数据类型及其常量	(7)
2.3 算术运算符、赋值运算符及其表达式	(10)
2.4 关系运算符、逻辑运算符及其表达式	(12)
2.5 逗号运算符、条件运算符及其表达式	(13)
2.6 变量的初始化.....	(15)
2.7 不同类型数据之间的转换.....	(15)
2.8 类型定义.....	(16)
2.9 应用举例.....	(17)
习题	(17)
第三章 基本语句	(19)
3.1 语句和复合语句.....	(19)
3.2 数据的输入.....	(19)
3.3 数据的输出.....	(21)
3.4 其它基本语句.....	(23)
3.5 应用举例.....	(24)
习题	(24)
第四章 选择结构程序设计	(26)
4.1 if 条件选择语句	(27)
4.2 switch 多分支选择语句.....	(29)
4.3 应用举例.....	(32)
习题	(34)
第五章 循环结构程序设计	(36)
5.1 while 循环语句	(36)
5.2 do-while 循环语句	(37)
5.3 for 循环语句	(38)
5.4 break 语句和 continue 语句	(41)
5.5 空操作语句和 goto 语句	(43)

5.6	应用举例	(43)
	习题	(45)
第六章	数组	(48)
6.1	一维数组	(48)
6.2	二维数组	(51)
6.3	字符数组和字符组	(54)
6.4	应用举例	(60)
	习题	(62)
第七章	函数和变量	(64)
7.1	函数的概念	(64)
7.2	函数的定义和调用	(65)
7.3	函数的返回值及其类型	(66)
7.4	函数的参数及其传递方式	(69)
7.5	函数的嵌套调和递归调用	(73)
7.6	变量的作用域及其存储类型	(76)
7.7	内部函数的外部函数	(80)
7.8	应用举例	(81)
	习题	(84)
第八章	结构和联合	(86)
8.1	结构类型变量的定义	(86)
8.2	结构类型变量的引用	(88)
8.3	结构的初始化	(89)
8.4	结构和函数	(90)
8.5	结构和数组	(91)
8.6	结构的嵌套	(93)
8.7	联合	(95)
8.8	枚举	(97)
8.9	应用举例	(98)
	习题	(100)
第九章	指针	(101)
9.1	指针的基本概念	(101)
9.2	指针变量的定义和引用	(101)
9.3	指针和结构	(104)
9.4	指针和数组	(110)
9.5	指针和函数	(117)
9.6	应用举例	(126)
	习题	(132)
第十章	位运算	(134)
10.1	二进制位运算	(134)

10.2	位段	(139)
10.3	应用举例	(141)
	习题	(141)
第十一章	编译预处理	(143)
11.1	宏定义	(143)
11.2	文件包括	(145)
11.3	条件编译	(147)
11.4	应用举例	(150)
	习题	(151)
第十二章	文件	(153)
12.1	文件概述	(153)
12.2	文件类型指针和文件号	(154)
12.3	缓冲文件系统	(154)
12.4	非缓冲文件系统	(161)
	习题	(165)
附录 A	标准 ASCII 字符集	(167)
附录 B	运算符的优先级认真结合性	(171)
附录 C	Turbo C 集成开发环境简介	(172)
附录 D	C 语言的巴科斯范式(BNF)描述	(175)
附录 E	各章习题解答	(180)
附录 F	94 至 97 历年来全国计算机等级考试 C 语言程序设计试题及答案	(208)

第一章 绪 言

C 语言是目前世界上最为流行的计算机高级程序设计语言之一,它设计精巧,功能齐全,既适合于编写应用软件,又特别适合于编写系统软件。

C 语言是在 1972 年至 1973 年间由 D. M. Ritchie 在贝尔实验室为描述和实现 UNIX 操作系统而设计和实现的。在此以前象 UNIX 操作系统这样的系统软件,一般都是利用汇编语言这种低级语言来编写的,自 C 语言开发成功以来,使得利用高级语言来编写系统软件成为可能。UNIX 操作系统源代码的 90% 以上是由 C 语言编写的,UNIX 操作系统的一些主要特点,如便于理解、易于修改及具有良好的可移植性等,在一定程度上都受益于 C 语言,所以,UNIX 操作系统的成功与 C 语言是密不可分的。

最初的 C 语言是依赖于 UNIX 操作系统环境的,目前,随着 C 语言不断的发展及应用的普及,C 语言已经能够在多种操作系统如 UNIX、DOS 等环境下运行,而且实用的 C 语言编译系统种类也越来越多,适用于 IBM-PC 机运行的就有 Microsoft C (MSC)、Turbo C (TC) 和 Lattice C (LC) 等。

1.1 C 语言的特点

C 语言能够成为目前应用最广泛的高级程序设计语言之一,完全是由其语言特点决定的。C 语言的特点可大致归纳如下:

1. C 语言短小精悍,基本组成部分紧凑、简洁

C 语言一共只有 32 个标准的关键字、45 个标准的运算符以及 9 种控制语句,不但语言的组成精练、简洁,而且使用方便、灵活。

2. C 语言运算符丰富,表达能力强

C 语言具有“高级语言”和“低级语言”的双重特点,其运算符包含的内容广泛,所生成的表达式简练、灵活,有利于提高编译效率和目标代码的质量。

3. C 语言数据结构丰富,结构化好

C 语言提供了编写结构化程序所需要的各种数据结构和控制结构,这些丰富的数据结构和控制结构以及以函数调用为主的程序设计风格,保证了利用 C 语言所编写的程序能够具有良好的结构化。

4. C 语言提供了某些接近汇编语言的功能,有利于编写系统软件

C 语言提供的一些运算和操作,能够实现汇编语言的一些功能,如它可以直接访问物理地址,并能进行二进制位运算等,这为编写系统软件提供了方便条件。

5. C 语言程序所生成的目标代码质量高

C 语言程序所生成的目标代码的效率仅比用汇编语言描述同一个问题低 20% 左右,因此,用 C 语言编写的程序执行效率高。

6. C 语言程序可移植性好

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。

当然,C 语言本身也有其弱点:一是运算符的优先级较多,不容易记忆;二是由于 C 语言的语法限制不太严格,这在增强了程序设计的灵活性的同时,在一定程度上也降低了某些安全性,这对程序设计人员提出了更高的要求。

总之,由于 C 语言的上述特点,使得 C 语言越来越受到程序设计人员的重视,并且已经在广泛的领域里得到了应用。

1.2 C 语言程序的开发过程

开发一个 C 语言程序的基本过程如图 1.1 所示。

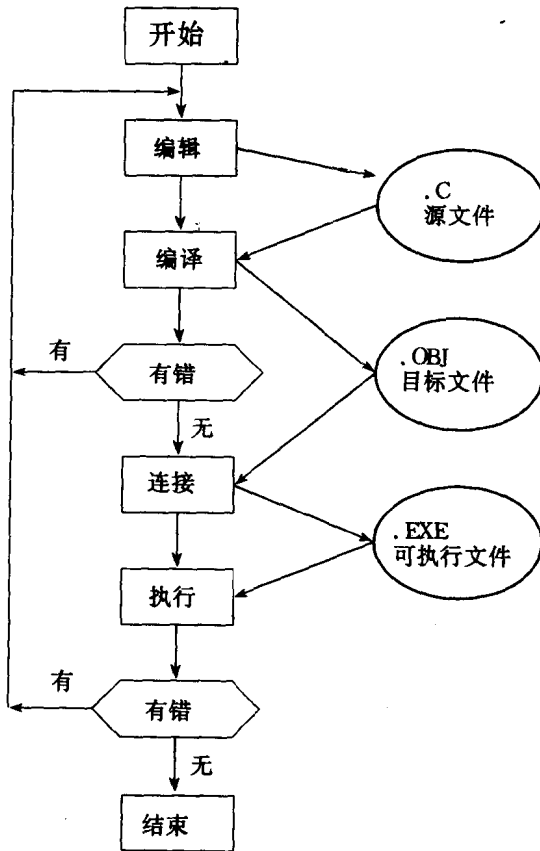


图 1.1 C 语言程序开发过程

1. 编辑

选择适当的编辑程序,将 C 语言源程序通过键盘输入到计算机中,并以文件的形式存入到磁盘中。在 DOS 系统下,经过编辑后得到的源程序文件都是以 .C 为其文件扩展名。

2. 编译

通过编辑程序将源程序输入到计算机后,需要经过 C 语言编译器将其生成目标程序。在对源程序的编译过程中,可能会发现程序中的一些语法错误,这时就需要重新利用编辑程序来修改源程序,然后再重新编译。在 DOS 系统下,经过编译后得到的目标文件都以 .OBJ 为其文件扩展名。

3. 连接

经过编译后生成的目标文件是不能直接执行的,它需要经过连接之后才能生成可执行的代码。在 DOS 系统下,连接后所得到的可执行文件都以 .EXE 为其文件扩展名。

4. 执行

经过编译、连接之后,源程序文件就可以生成可执行的文件,这时就可以执行了。在 DOS 系统下,只要键入可执行的文件名,并按“回车”键后,就可执行文件了。

现在有许多厂家都推出一种集成环境来处理 C 语言程序,如 Turbo C,在这种集成环境下,对程序的编辑、编译和连接等操作,都可以在一个窗口下进行,使用起来非常方便。

在本书附录 II 中简单介绍了 Turbo C 集成开发环境的使用方法,感兴趣的读者可以参考使用。

1.3 简单的 C 语言程序

这一节将介绍几个简单的 C 语言程序,并对其基本语法成分进行相应的说明,以便使读者对 C 语言程序有一个概括的了解。

例 1.1 编写一个 C 语言程序,显示字符串“Hello!”。

```
#include "stdio. h"
main()
{
    printf("Hello! \n");
}
```

上述程序是一个简单而完整的 C 语言程序,经过编辑、编译和连接后,其执行结果是在屏幕的当前光标位置处显示如下字符串:

Hello!

下面将对上述程序进行一下说明:

(1) 一个 C 语言程序可以由多个函数组成,但任何一个完整的 C 语言程序,都必须包含一个且只能包含一个名为 main 的函数,程序总是从 main 函数开始执行的。

(2) 由左右花括号括起来的部分是函数体,函数体中的语句将实现程序的预定功能。在本例中,main 函数的函数体中只有 printf 一个语句,它的功能是进行格式化输出(显示),即将字符串“Hello! \n”显示在终端屏幕上。其中,字符串中的字符“\”和“n”合起来,表示一个“换行”字符,在“换行”字符后输出的任何字符,将被显示在屏幕的下一行上。

(3) C 语言中的每个基本语句,都是以“;”结束的。

(4) C 语言程序的书写格式比较自由,没有固定的格式要求,在一行内,既可以写一个语句,也可以写多个语句。为了提高程序的可读性,往往根据语句的从属关系,以缩进书写的形式

来体现出语句的层次性。

(5) #include 语句是编译预处理语句,其作用是将由双引号或尖括号括起来的文件中的内容,读入到该语句的位置处。在使用 C 语言输入、输出库函数时,一般需要使用 #include 语句将"stdio. h"文件包含到源文件中。有关#include 语句的作用及其使用方法,将在第十一章“编译预处理”中作详细介绍。

例 1.2 从键盘输入两个整数,并将这两个整数之和显示出来。

```
#include "stdio. h"
main()          /* 计算两个整数之和 */
{
    int x,y,z;
    scanf("%d%d",&x,&y); /* 读入两个整数,存入变量 x 和 y 中 */
    z=x+y;
    printf("The sum of %d and %d is %d",x,y,z);
}
```

上述程序经过编辑、编译和连接后,执行情况如下:

当程序执行到 scanf 语句时,将等待用户输入两个整型数据后再继续执行,假如用户输入 10 和 20(此时,x 的值为 10,y 的值为 20),则程序将显示如下信息:

The sum of 10 and 20 is 30

下面,对上述程序进行一下说明:

(1) 程序中由/* 和 */括起来的内容是程序的注释部分,它是为增加程序的可读性而设置的。注释部分对程序的编译过程和执行结果没有任何影响。

(2) C 语言中的所有变量都必须定义为某种数据类型,同时必须遵循“先定义、后使用”的原则,如语句:

```
int x,y,z;
```

定义了 x,y,z 是三个整型变量,以后就可以使用这三个变量来存放整型数据。

(3) 一个 C 语言程序可以由多个函数组成,通过函数之间的相互调用来实现相应的功能。程序中所使用的函数,既可以是系统提供的库函数,也可以是用户根据需要自己定义的函数。如上述 main 函数中调用的 scanf 函数和 printf 函数,就是系统提供的库函数,这些函数不需要用户自己定义,在需要时,只要按照指定的格式进行调用即可。C 语言编译系统提供的库函数非常丰富,特别是与硬件打交道的部分,很多工作只要调用库函数就可以实现。

(4) 程序中调用的 scanf 函数的作用是进行格式化输入,其中由圆括号括起的部分是函数参数部分,不同的函数需要不同的参数,scanf 函数中的参数主要包括两部分的内容:一是“格式控制”部分,它用于对输入数据的格式进行说明;二是“地址表”部分(本书中出现的表的概念,如地址表、输出表等,是指用逗号分隔的有限个元素序列),它使用的是存放输入数据的变量的地址。

程序中调用的 printf 函数的作用是进行格式化输出,其参数也包括两部分内容:一是“格式控制”部分,用于对输出数据的格式进行说明;二是“输出表列”部分,它使用的是存放输出数据的变量本身。

有关数据的输入、输出以及函数的调用形式,将在以后作详细的介绍。

习 题

- 1.1 请简述 C 语言的主要特点及其用途。
- 1.2 请参照本章例题,编写一个 C 语言程序,用于显示以下信息:
Happy New Year!
- 1.3 请参照本章例题,分析下面的 C 语言程序,并给出运行结果。

```
#include "stdio. h"
main( )
{
    int a,b,c;
    a=100;
    b=20;
    c=a-b;
    printf("sum=%d",c);
}
```

- 1.4 请参照本章例题,编写一个 C 语言程序,用于显示以下信息:

```
#####
Wonderful!
#####
```

- 1.5 请参照本章例题,分析下面的 C 语言程序,并给出模拟运行结果。

```
#include "stdio. h"
main( )
{
    int x,y,z;
    scanf("%d%d",&x,&y);
    z=x/y;
    printf("Result=%d",z);
}
```

第二章 数据类型、运算符及其表达式

或显式地与一种数据类型相联系。每种数据类型都表明了它的可能取值范围及能在其上所进行的运算。

C 语言中的数据类型,大体上可划分为基本的数据类型和导出的数据类型两种,基本数据类型主要包括整型、字符型和单双精度浮点型等;导出数据类型是在基本数据类型的基础上产生的,其中包括数组、结构等。

本章主要讨论 C 语言中变量名及其常量的构成形式,说明 C 语言中的几种基本数据类型、算术运算符及构成算术表达式的一些基本规则。

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。常量也分为不同的类型,这是由常量本身隐含决定的,将在下面详细介绍。为了增加程序的可读性,可以用一个名字(字符序列)来代表一个常量,此时的常量被称为符号常量。有关符号常量的使用,将在第十一章中详细介绍。

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。变量被区分为不同的类型,不同类型的变量在内存中占用不同的存储单元,以便用来存放相应变量的值。

在计算机语言中,标识符的概念经常被用到。所谓标识符,是指用来标识程序中用到变量名、函数名、类型名、数组名、文件名以及符号常量名等的有效字符序列。

需要说明的是,保留字(又称关键字)也是 C 语言中的一种标识符,它用来命名 C 语言程序中的语句、数据类型和变量属性等。每个保留字都有固定的含义,不能另作其它用途。

在 C 语言中,标识符的命名规则是:由字母(大、小写皆可)、数字及下划线组成,且第一个字符必须是字母或下划线。

由上述标识符的命名规则可知,下面的变量名是合法的:

```
year, Day, ATOK, x1, _CWS, _change_to
```

而下面的变量名是不合法的:

```
#123, .COM, $100, 1996Y, 1_2_3
```

在 C 语言中,大写字母和小写字母是有区别的,即作为不同的字母来看待。如变量 RAN、Ran 和 ran 分别表示三个不同的变量,这一点同其它高级语言是有区别的,应引起注意。

组成变量名(标识符)的有效字符数随 C 语言的编译系统而定。有的编译系统允许使用长达 31 个字符的变量名,而有的编译系统只取变量名的前 8 个字符作为有效字符,后面的字符

无效,不被识别,这样,只要变量名的前 8 个字符相同,就被认为是同一个变量。因此,在进行程序设计之前,应首先了解所使用的编译系统中对变量名长度的规定,以免造成变量使用上的混乱。

2.2 基本数据类型及其常量

在 C 语言中,最基本的数据类型只有四种,它们分别由如下标识符进行定义:

int	整型
char	字符型
float	单精度浮点型
double	双精度浮点型

C 语言规定,对程序中用到的所有变量,都必须先定义后使用。在定义变量时,不能把 C 语言中具有固定含义的保留字(如 int、char 等)作为变量名,同时,同一个函数内所定义的变量不能同名。

2.2.1 整型变量及其常量

1. 整型变量

整型变量用来存放整型数据(即不带小数点的数)。其定义方式如下所示:

```
int i1,i2;
```

其中 i1 和 i2 即被定义为整型变量。

2. 整型常量

在 C 语言中,整型常量可以用三种数制来表示:

(1) 十进制整型常量:如,250,-12 等,其每个数字位可以是 0~9。

(2) 十六进制整型常量:如果整型常量以 0x 或 0X 开头,那么这就是用十六进制形式表示的整型常量。如,十进制数的 128,用十六进制表示为 0x80 或 0X80,其每个数字位可以是 0~9,A~F。

(3) 八进制表示的整型常量:如果整型常量的最高位为 0,那么它就是以八进制形式表示的整型常量。如,十进制数的 128,用八进制表示为 0200。需要注意的是,八进制数中的每个数字位必须是 0~7。

2.2.2 浮点型变量及其常量

1. 浮点型变量

在 C 语言中,把带有小数点的数称为浮点数,也可以称为实型数。

浮点型变量又被称为实型变量,按其能够表示的数的精度,又被分为单精度浮点型变量和双精度浮点型变量。

单精度浮点型变量的定义方式如下所示:

```
float f1,f2;
```

其中 f1 和 f2 即被定义为单精度浮点型变量。

双精度浮点型变量的定义方式如下所示:


```
double d1,d2;
```

其中 d1 和 d2 即被定义为双精度浮点型变量。

单精度浮点型变量和双精度浮点型变量之间的差异,仅仅体现在所能表示的数的精度上,如果单精度浮点型所提供的精度不能满足要求时,则可以考虑使用双精度浮点型。在不同的系统中,float 型数据和 double 型数据所能提供的精度是有差异的,一般来讲,在同一个系统中,double 型变量值的最大有效位数,通常是 float 型的两倍。

2. 浮点型常量

浮点型常量一般不分 float 型和 double 型,任何一个浮点型常量,既可以赋给 float 型变量,也可以赋给 double 型变量,但由于 float 型变量和 double 型变量所能表示的数的精度不同,所以,在赋值时,将根据变量的类型来截取相应的有效位数。

浮点型常量有如下两种表示形式:

(1) 十进制数形式:它是由数字和小数点组成的,如,3.14159,-7.2,9.8 等都是用十进制数的形式表示的浮点数。

(2) 指数法形式:指数法又称为科学记数法,它是为方便计算机对浮点数的处理而提出的。如,十进制数的 180000.0,用指数法可表示为 1.8e5,其中 1.8 被称为尾数,5 被称为指数,字母 e 也可以用 E 来表示。又如,0.00123 可表示为 1.23E-3。

需要注意的是,用指数形式来表示浮点数时,字母 e 或 E 之前(即尾数部分)必须有数字,且 e 后面的指数部分必须是整数,如,e-3,9.8e2.1,e5 等都是不合法的指数表示形式。

2.2.3 字符型变量及其常量

1. 字符型变量

字符型变量用于存放一个单个字符。它的定义方式如下所示:

```
char c1,c2;
```

其中 c1 和 c2 就被定义为字符型变量。

2. 字符型常量

字符型常量是由一对单引号括起来的一个字符,如,'A','*'和'8'等都是合法的字符型常量。

在 C 语言中,还允许使用一些特殊形式的字符型常量,这些字符型常量都是以反斜线字符'\ '开头的字符序列。下面介绍几个常用的以'\ '字符开头的特殊字符:

'\n': 换行字符

'\r': 回车字符

'\b': 退格字符

'\t': 制表字符,又被称为横向跳格字符

'\'' : 单引号字符

'\"' : 双引号字符

如果要显示单引号或双引号字符,则在字符串中需要以上述特殊字符形式给出。

除了上述具有特殊意义的字符外,C 语言还允许在字符'\ '后面紧跟 1~3 位八进制数或在'\x'后面紧跟 1~2 位十六进制数来表示相应系统中所使用的字符的编码值。使用这种表示方法,可以表示字符集中的任一字符,包括某些难以输入和显示的“控制字符”,ASCII 码表中