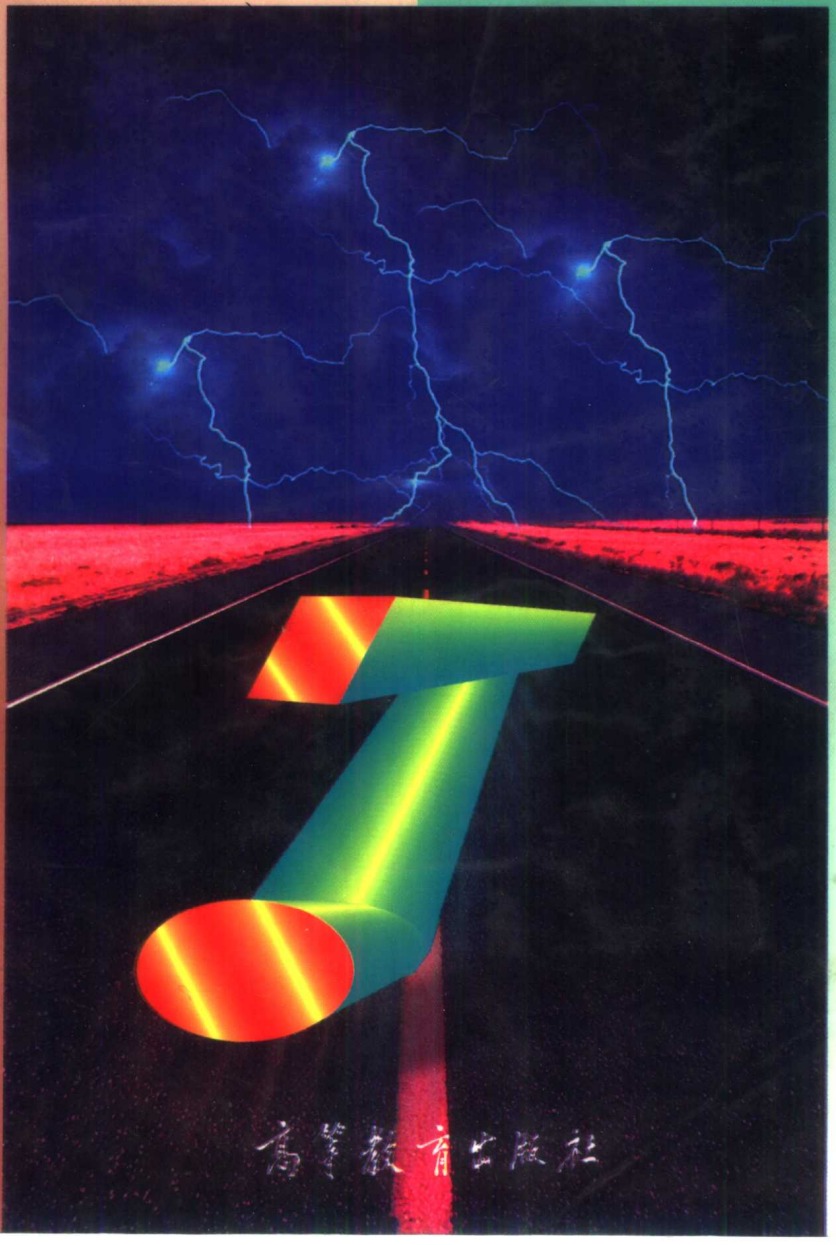
高等学校计算机基础教材系列

C程序设计及应用

李盘林　孟宪福　编著



高等教育出版社

高等学校计算机基础教材系列

C 程序设计及应用

李盘林　孟宪福　编著

高 等 教 育 出 版 社

（京）112号

内 容 提 要

C 语言是一种应用非常广泛的结构化高级程序设计语言，它既适合于编写应用软件，也适合于编写系统软件。本书由 12 章组成，按照循序渐进的原则，详细地介绍了 C 语言的基本概念和语法规则。在此基础上，通过精选的典型例题分析，使读者能够很快掌握用 C 语言进行程序设计的技巧和方法。特别是，为了使读者能够尽快利用 C 语言编写大型实用程序，在书中的最后一章详细分析了一个图形编辑程序的设计与实现过程，并给出了完整的源程序。

本书是作者根据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，在文字叙述上力求条理清晰、简洁，以便于读者阅读。

本书可作为大专院校有关专业程序设计课程的教材或教学参考书，也可作为自学用书。

图书在版编目(CIP)数据

C 程序设计及应用/李盘林，孟宪福编著. -北京：高等教育出版社，1998
ISBN 7-04-006409-X

Ⅰ.C… Ⅱ.①李… ②孟… Ⅲ.C 语言-程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（97）第 24945 号

高等教育出版社出版
北京沙滩后街 55 号
邮政编码：100009 电话：64054588 传真：64014048
新华书店总店北京发行所发行
高等教育出版社印刷厂印装
*
开本 787×1092 1/16 印张 15.5 字数 380 000
1998 年 5 月第 1 版 1998 年 8 月第 1 次印刷
印数 0 001-5 083
定价：15.00 元

凡购买高等教育出版社的图书，如有缺页、倒页、脱页等质量问题者，请与当地图书销售部门联系调换

前　言

C 语言是目前应用最为广泛的高级程序设计语言之一，它短小精悍，功能齐全，结构化，能够运行于多种操作系统环境，既适合于编写应用软件，又适合于编写系统软件。

作者多年来一直从事 C 语言的教学工作，同时也利用 C 语言开发大型的实际课题。本书就是作者根据多年的教学经验和应用 C 语言的体会写成的，它既注重于 C 语言的理论体系，又特别强调 C 语言的应用。本书的主要特点可归纳如下：

1. 在内容编排上，按照循序渐进的原则，逐步介绍 C 语言中的基本概念和理论。在章节的安排上，尽可能考虑初学者的接受能力，使整个学习过程按照从简单到复杂的顺序进行。

2. 在讲授 C 语言基本理论的基础上，着重强调 C 语言的应用。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。每一章中都包含“应用举例”一节，其中的例题涉及到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，能使读者对本章讲授的内容及其应用有一个全面的了解。

3. 为了使读者能够尽快掌握利用 C 语言来编写大型实用程序的方法，在本书的第十二章中，详细介绍了利用 C 语言编写的规模较大的图形编辑程序 Panda。该程序几乎包含了 C 语言的各个方面，其中包括菜单设计、图形处理（图形输入和图形编辑）以及文件管理等。仔细地阅读和分析此程序，无疑会在短期内提高利用 C 语言来编写大程序的能力。

4. 每章的最后都附有大量的习题，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大有益处。

5. 在语言的描述上，尽量使用规范化的术语。同时，在文字叙述上力求条理清晰、简单明了，以利于读者阅读。

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调试通过，读者可以边学习边上机。刚开始时可以调试本书中的例题，待学习一段时间之后，就可以调试自己编写的程序了。只有这样，才能加快学习进度，提高学习效率。

在本书编写过程中，得到了同济大学计算机系吴永明教授的支持和帮助，在此表示谢意。

由于作者水平有限，经验不足，书中一定会有不少缺点和错误，敬请有关老师、计算机工作者和广大读者批评指正。

编　者

1997 年 6 月

目　录

第一章 绪 言

随着计算机硬件技术和软件水平的不断提高，作为人机交流主要工具的计算机程序设计语言，也经历了从简单到复杂、从低级到高级的发展过程。在诸多的计算机高级语言当中，C 语言是目前国内外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、使用灵活，既适合于编写应用软件，又适合于编写系统软件。

C 语言最初是为描述和实现 UNIX 操作系统而设计和实现的。在此以前像 UNIX 操作系统这样的系统软件，一般都是用汇编语言这种低级语言编写的，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 %以上是用 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，随着 C 语言的不断发展以及应用的普及，目前，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 语言编译系统种类繁多，适用于 IBM PC 机的就有 Microsoft C (MSC)、Turbo C (TC)、Borland C (BC)和 Lattice C (LC)等。

1.1 C 语言简况

1.1.1 C 语言发展简史

在 20 世纪 60 年代，随着计算机科学体系的形成与完善，高级程序设计语言的研究得到了长足的发展。但是，在当时出现的高级语言中，缺乏用于编写像操作系统、编译程序等系统软件的工具，系统程序的设计主要还是依赖于汇编语言。为了改变这种状况，1969 年，Martin Richards 设计并实现了 BCPL(Basic Combined Programming Language)语言，后来，这一语言被移植到了多种计算机上，并得到了广泛的应用。此后不久，Ken Thompson 在 BCPL 语言的基础上设计并实现了 B 语言，并用 B 语言在 PDP-7 机上实现了第一个 UNIX 操作系统。接着，在 1972 年至 1973 年间，D. M. Ritchie 在 B 语言的基础上，又重新设计了一种语言，并在 PDP-11 机上实现，同时用这种语言重写了 UNIX 操作系统。由于这一语言是在 BCPL 语言和 B 语言的基础上开发出来的，因此被称为 C 语言。由于 BCPL 语言和 B 语言是无类型的语言，而 C 语言却能支持多种数据类型，因此 C 语言与 BCPL 语言和 B 语言是不同的，它更能反映当代计算机的体系结构，因而得到了广泛的应用。

1.1.2 C 语言的特点

C 语言能成为目前应用最为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 语言能够处理多种运算符，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 语言所编写的程序具有良好的结构化。同时，在 C 语言程序设计中，允许将一个复杂的程序分割为多个模块，由多人同时编写，当分别调试完成后，再通过连接程序连接到一起，形成一个完整的程序。

(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 语言具有“高级语言”和“低级语言”的双重特点，它提供的一些运算和操作，能够实现汇编语言的一些功能，如 C 语言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) 由 C 语言程序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很高。

(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和结合性比较复杂，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 语言程序的开发步骤

开发一个 C 语言程序的基本步骤如图 1.1 所示。

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

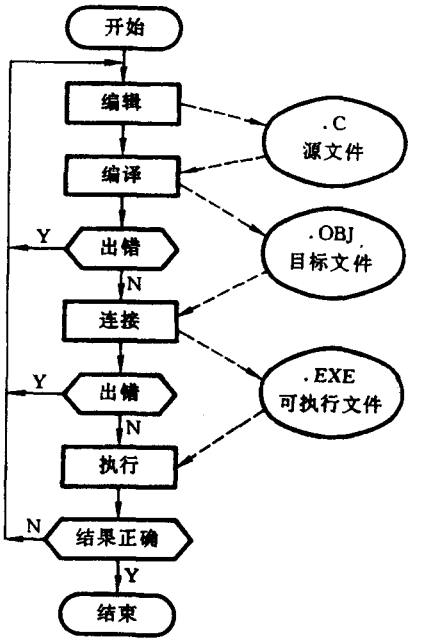


图 1.1　C 语言程序开发步骤

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 语言编译器将其编译成目标程序。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是以.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是以.EXE 为其文件扩展名。

4. 执行

经过编译、连接之后，源程序文件被生成可执行的文件，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。

现在有许多厂家都推出集成环境来处理 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。

在本书的附录III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 C 语言的程序结构

C 语言的程序结构比较简单，很容易掌握，它主要是通过函数之间的相互调用来实现指定的功能。在这一节中，我们通过编写几个简单的 C 语言程序，来阐述 C 语言的程序结构，同时也对 C 语言的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，为以后的学习打下基础。

例 1.1 编写一个 C 语言程序，用于显示字符串“ Hello, World!”。

程序如下：

```
#include "stdio.h"
main( )
{
    printf("Hello, World!\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, World!
```

下面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。

(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, World!\n ”显示在终端屏幕上。其中，字符串“ Hello, World!”中的字

符\n 表示一个“换行”符，在“换行”符后面输出的任何字符，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. 分号是 C 语言语句的终结符。

(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“编译预处理”一章中做详细介绍。

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。

程序如下：

```
#include "stdio.h"
int  ADDxy(a, b)        /*计算两个整数之和*/
   int  a, b;
{   int  c;
    c=a+b;
    return(c);
}
main( )
{
   int  x, y, z;
   scanf("%d%d", &x, &y);   /*读入两个整数，存入变量 x 和 y 中*/
   z=ADDxy(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，执行情况如下所示：

当此程序从 main 函数开始执行，执行到 scanf 语句时，将等待用户从键盘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[①] 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，它是为了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;
```

[①] 鉴于计算机程序的习惯和特点——所有的文字和符号都用正体，因此，为了保持上下文的一致和方便读者阅读，我们在对程序中的标识符进行说明时也用正体。

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。而函数 ADDxy 是自定义的函数，它是用户为了实现加法功能而自己编写的函数。每一个函数都用于完成一特定的功能，ADDxy 函数的功能是求两个整数之和并通过 return 语句将结果返回给 main 函数。正确地使用函数，将有助于编写易于理解的、结构化好的程序。有关函数的详细说明请参阅“函数”一章。

(4) 程序中调用的 scanf 函数的作用是进行格式化输入，其中由圆括号括起来的部分是函数的参数部分，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），它指的是存放输入数据的各个变量的地址。

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”部分，它指的是存放输出数据的各个变量。

有关数据的输入、输出以及函数的调用形式，将在以后做详细的介绍。

习 题

1.1 试简述 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：

Welcome to C!

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("result=%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：

C language!

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、运算符和表达式

常量、变量以及函数等都是程序的基本操作对象，统称为数据。根据数据的取值范围以及能在其上所进行的运算，可把数据分为各种类型，不同类型的数据一般在内存中占用不同的存储空间，同时，数据的类型不同，能够参加的运算也不同。

C 语言中的数据类型非常丰富，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 语言中的一些基本概念，如变量、标识符等，同时详细说明 C 语言中的几种基本数据类型、算术运算符、关系运算符、逻辑运算符以及利用这些运算符来构成相应表达式的一些规则。

2.1　基 本 概 念

2.1.1　标识符

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的变量名、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。

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，下面的标识符名是合法的：

year， Day， ATOK， x1， _CWS， _change_to

而下面的标识符名是不合法的：

#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。如标识符 RAN 、 Ran 和 ran 分别表示三个不同的标识符，这一点同其它高级语言是有区别的，应引起注意。

2.1.2　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，如 100, 3.14 等。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“编译预处理”一章中详细介绍。

2.1.3　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。

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。

2.1.4 关键字

关键字，又被称为保留字或保留关键字，也是 C 语言中的一种标识符，用来命名 C 语言程序中的语句、数据类型和变量属性等。每个关键字都有固定的含义，不能另作其它用途，C 语言中的所有关键字都是用小写字母表示的。

2.2 基本数据类型

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，每个变量只能与一种数据类型相联系。在定义变量时，不能把 C 语言中具有固定含义的关键字（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

整型变量可用来存放整型数据（即不带小数点的数）。其定义方式如下：

```
int i1, i2;
```

其中 i1 和 i2 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 。