高等学校“十三五”规划教材

C语言程序设计

（第二版）

主编　崔永君　王芬琴　孙娟红

西安电子科技大学出版社
http://www.xduph.com

高等学校“十三五”规划教材

C语言程序设计

（第二版）

主编　崔永君　王芬琴　孙娟红

西安电子科技大学出版社

内容简介

本书是在2011年第一版的基础上修订而成的。

本书共11章，根据初学者的认知特点，循序渐进、紧贴教学、深入浅出地讲述了C语言的基本概念、数据类型、结构化程序设计的三种结构(顺序结构、选择结构、循环结构)、数组、函数、指针、结构体与共用体、预编译、位运算以及文件等相关知识。通过大量有着明确知识点的例题与习题，使读者理解和掌握程序设计，更好地驾驭计算机这个“程序的机器”。

本书可作为高等院校计算机及理工类各专业、成人教育学院C语言程序设计课程的教材，也可作为高等学校学生和广大计算机爱好者学习掌握C语言的自学教材。

图书在版编目(CIP)数据

C语言程序设计 / 崔永君，王芬琴，孙娟红主编. —西安：西安电子科技大学出版社，2019.2
ISBN 978-7-5606-5231-3

Ⅰ. ① C… Ⅱ. ① 崔… ② 王… ③ 孙… Ⅲ. ① C语言—程序设计 Ⅳ. ① TP312.8

中国版本图书馆CIP数据核字(2019)第023728号

策划编辑 杨丕勇
责任编辑 秦媛媛 阎 彬
出版发行 西安电子科技大学出版社(西安市太白南路2号)
电 话 (029)88242885 88201467 邮 编 710071
网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com
经 销 新华书店
印刷单位 北京虎彩文化传播有限公司
版 次 2019年2月第1版 2019年2月第1次印刷
开 本 787毫米×1092毫米 1/16 印 张 19.5
字 数 462千字
印 数 10 501～11 500册
定 价 45.00元

ISBN 978-7-5606-5231-3 / TP

XDUP 5533002-6

*** 如有印装问题可调换 ***

❖❖❖ 前　　言 ❖❖❖

计算机程序设计基础是高等学校计算机基础课程中的核心课程、本书总结了作者多年的教学和软件开发经验，以 C 程序设计语言为基础，注重程序设计与软件开发的基本概念、方法和思路，旨在培养读者的基本编程能力、逻辑思维和抽象思维能力。学习程序设计对于大学生来说不仅是职业技能的培养过程，也是创造性思维的开发过程。

“C 语言程序设计”是计算机及相关专业的一门程序设计启蒙课程，也是许多计算机后续课程的基础。Joel Spolsky，昔日耶鲁大学计算机系学生，今日 Fog Creek 软件公司的 CEO 指出：

“虽然在实际使用中 C 语言已经越来越罕见，但是它仍然是当前程序员的共同语言。C 语言让程序员互相沟通，更重要的是，它比你在大学中学到的‘现代语言’(比如 ML 语言、Java 语言、Python 语言或者其他正在教授的流行垃圾语言)都更接近机器语言。”“不管你懂得多少延续、闭包、异常处理，只要你不能解释为什么 while(s++ = t++)的作用是复制字符串，那你就是在盲目无知的情况下编程，就像一个医生不懂最基本的解剖学就在开处方。”

本书以C程序设计零起点读者为主要对象，注重教材的可读性和可用性，由浅入深，强化知识点、算法、编程方法与技巧，很多例题后面给出了思考题，帮助读者了解什么是对的以及哪些是容易出错的，从而能够举一反三。本书还将程序测试、程序调试与排错、结构化与模块化程序设计方法等软件工程知识融入其中，并且习题以历年等级考试真题为主，题型丰富，具有代表性。

本书是在 2011 年第一版的基础上修订而成的。

本书共 11 章。第 1 章从程序设计语言的发展着手，通过例题，由浅入深地介绍了C程序设计的基本概貌；第 2 章通过有针对性的例题介绍 C 语言的基本数据类型、运算符与表达式；第 3、4、5 章详细讲解了面向过程的程序设计的三种基本结构：顺序、选择、循环；第 6 章介绍了一维数组、二维数组、字符数组和字符串的概念，并结合专业应用介绍了矩阵运算、数据表操作及杨辉三角的求解问题；第 7 章从模块化

程序设计的基本思想以及程序设计的易读性和可维护性出发，介绍了函数的基本概念，并介绍了多文件中函数和变量的处理；第 8 章从内存管理的角度对指针进行了较深入的分析；第 9 章介绍了结构体、共用体与预编译的基本知识，并介绍了数据结构中线性链表的基本知识；第 10 章介绍了位运算；第 11 章介绍了文件的基本概念和常用操作。

参加本书编写工作的有兰州交通大学博文学院孙娟红(第 2～6 章)、王芬琴(第 1、9 章)，兰州交通大学崔永君(第 7、8 章)、张永花(第 10、11 章，附录 B 和附录 C)。崔永君、王芬琴负责全书的策划、组织和定稿。

鉴于作者水平有限，书中难免会有疏漏之处，真诚地欢迎各位专家和读者批评指正，以帮助我们进一步完善教材。

编　者

2018 年 12 月

❖❖❖ 目　录 ❖❖❖

第 1 章　C 语言概述

1.1　程序设计语言的发展

自 1946 年世界上第一台电子计算机问世以来，计算机科学及其应用的发展十分迅猛，计算机被广泛地应用于人类生产、生活的各个领域，推动了社会的进步与发展。特别是随着因特网(Internet)的普及，传统的信息收集、传输及交换方式正被革命性地改变，人们已经难以摆脱对计算机的依赖，计算机将人类带入了一个新的时代——信息时代。

计算机是由硬件系统和软件系统两大部分构成的，硬件是物质基础，而软件可以说是计算机的灵魂。没有软件，计算机就是一台“裸机”，什么也不能干；有了软件，计算机才能灵动起来，成为一台真正的“电脑”。所有的软件，都是用计算机程序设计语言编写的。

计算机程序设计语言是人与计算机进行交流的有力工具。随着计算机技术的发展，计算机程序设计语言也不断发展。依据对硬件的依赖程度，计算机程序设计语言经历了从机器语言到汇编语言，再到高级语言的发展过程。

1．机器语言

机器语言是第一代计算机语言。计算机发明之初，人们只能用计算机的语言去命令计算机工作，也就是只能写出一串串由“0”和“1”组成的指令序列(程序)交由计算机执行，这种语言就是机器语言。例如，要完成将寄存器 BX 的内容送到寄存器 AX 中，其机器指令为 1000100111011000。

使用机器语言是十分痛苦的，特别是在程序有错需要修改时更是如此。而且，由于每台计算机的指令系统往往各不相同，所以，在一台计算机上执行的程序想要在另一台计算机上执行，必须另编程序，这就造成了大量的重复性工作。但由于计算机可以直接识别和运行用机器语言编写的程序，即无需翻译，而且机器语言是针对特定型号计算机的语言，故机器语言所编程序的运算效率是所有程序语言中最高的。

2．汇编语言

为了减轻使用机器语言编程的困难，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来代替一个特定指令的二进制串。例如，同样要完成将寄存器 BX 的内容送到寄存器 AX 中，用汇编语言编程为 MOV BX, AX(MOV 是数据传送指令)，这样一

来，人们容易读懂并理解程序在干什么，纠错及维护也变得方便了，这种程序设计语言就是汇编语言。然而计算机是不认识这些符号的，这就需要一个专门的程序，负责将这些符号翻译成二进制的机器语言，这种翻译程序被称为汇编程序。

汇编语言同样十分依赖于机器硬件，移植性不好，但运算效率仍十分高。针对计算机特定硬件而编制的汇编语言程序能准确地发挥计算机硬件的功能和特长，程序精练且质量高，所以至今仍是一种常用且强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的痛苦经历中人们意识到，应该设计一种这样的语言：接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有的机器上通用。经过努力，1954 年，第一个完全脱离机器硬件的高级语言——FORTRAN 问世了。

20 世纪 60 年代中后期，软件越来越多，规模越来越大，而软件的生产基本上是各自为政，缺乏科学而规范的系统规划与测试以及评估标准，其恶果是耗费巨资建立起来的大批软件系统由于含有错误而无法使用，甚至带来巨大损失，软件给人的感觉是越来越不可靠，几乎没有不出错的软件。这一切极大地震动了计算机界，史称“软件危机”。人们认识到：大型程序的编制不同于写小程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性，也便于验证正确性。为此，1969 年提出了结构化程序设计方法。1970 年，第一个结构化程序设计语言——Pascal 语言出现，标志着结构化程序设计时期的开始。

自 20 世纪 80 年代初开始，在软件设计思想上又产生了一次革命，其成果就是产生了面向对象的程序设计方法。在此之前的高级语言几乎都是面向过程的，程序的执行是流水线式的，即在一个模块被执行前，人们不能干别的事，也无法动态地改变程序的执行方向。这和人们日常处理事物的方式是不一致的，对人而言是希望发生一件事就处理一件事，也就是说，不能面向过程，而应是面向具体的应用功能，也就是对象(object)。其方法就是软件的集成化，如同硬件的集成电路一样，生产一些通用的、封装精密的功能模块(称之为软件集成块)，它与具体应用无关，但能相互组合，完成具体的功能，同时又能重复使用。对使用者来说，只关心它的接口(输入量、输出量)及能实现的功能，至于具体是如何实现的，使用者完全不用关心，C++、VB、Delphi 就是典型代表。几十年来，共有几百种高级语言出现，有重要意义的有几十种，影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、VC、VB、Delphi、JAVA 等。

高级语言的下一个发展目标是面向应用，也就是说，只需要告诉程序你要干什么，程序就能自动生成算法并自动进行处理，这就是非过程化的程序设计语言。

1.2 C 语言的发展及其特点

C 语言是 1972 年由美国的 Dennis Ritchie 设计的，并首次在 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL(Basic Combind Programming Language)演变而来。1970 年，贝尔实验室的 Ken Thomposn 根据 BCPL 语言设计出比较先进的并取名为 B 的语言，

最后有了 C 语言的问世。C 语言是国际上广泛流行的计算机高级语言，它既可以用来编写系统软件，也可以用来编写应用软件。

随着微型计算机的日益普及，出现了许多 C 语言版本，但由于没有统一的标准，这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准协会为 C 语言制定了一套 ANSI 标准，通常称之为 ANSI C，成为现行的 C 语言标准。

C 语言之所以发展迅速，成为最受欢迎的语言之一，这源于 C 语言具有很多显著的特点。主要如下：

(1) 简洁、紧凑，使用方便，灵活。

ANSI C 共有 32 个关键字，9 种控制语句，程序书写自如，一般用小写字母表示，压缩了一切不必要的成分。C 语言在表达方式上力求简单易行，如使用一对花括号“{}”来表示复合语句的开始和结束，用赋值运算符(如 +=、-=、*=、/+ 等)表示进行相应运算并且将结果赋值给左值(赋值号左边的变量)，等等。

(2) 运算符和数据结构丰富，表达式类型多样。

C 语言共有 34 种运算符。在 C 语言中，把括号、赋值号、逗号、关系运算、逻辑运算等都作为运算符处理。灵活使用各种运算符，可以实现在其他高级语言中难以实现的运算。

C 语言提供了丰富的数据类型。C 语言的数据类型基本可以分为两大类：一类是简单类型，如整型、实型、字符型等；另一类是在简单类型基础上按层次产生的各种构造类型，如数组类型、结构体类型和共用体类型等，此外还有指针类型。利用这些数据类型，C 语言能够实现各种复杂的数据结构，如线性表、链表、栈、队列、树、图等。

C 语言的表达形式多样，既提高了编译效率和目标代码的质量，又提高了程序可读性。

(3) C 语言是结构化的程序语言。

结构化语言的特点是代码与数据分离，即程序的各个部分除了必要的信息交流外彼此独立，从而使程序之间很容易实现程序段的共享。C 语言提供了顺序结构、选择(分支)结构和循环结构三种基本结构语句，并以函数作为模块，实现程序的模块化设计，符合现代编程风格。

(4) 语法限制不太严格，程序设计自由度大。

C 语言编译系统检查不太严格，例如，在 C 语言中对数组下标越界不进行检查，变量类型灵活使用，整型数据和字符型数据及逻辑型数据可以通用。在语法上放宽限度，在程序设计上灵活自由，相应地，检查错误的任务也就转到了编程者的身上。因此，这就要求编程者在编程时要自我约束，养成良好的编程习惯。

(5) 生成目标代码质量高，程序执行效率高。

C 语言有位(bit)操作的功能，可以直接对硬件进行操作，这使得 C 语言既具有高级语言的所有优点，又具有低级语言的许多功能，成为所谓的“中间语言”。C 语言通过对位、字节和地址进行操作，可以对硬件进行编程，并可实现汇编语言的大部分功能，具有高效率的目标代码。用 C 语言编写的程序生成代码的效率比汇编语言编写的仅低 10%～20%。

(6) C 语言编写的程序可移植性好。

与汇编语言相比，C 语言编写的程序基本上是不做修改或稍微修改就可以在其他工

作平台上运行，从而使开发的软件独立于具体的计算机体系结构和软件环境，达到“共用”的目标，提高了软件的生产效率，节省了用户的投资。

C 语言还有其他优点，读者可以在学习及实践中体会。当然，C 语言也和其他语言一样，存在不足之处，如某些运算符优先顺序和习惯不完全一致，类型的转换比较随意等。尽管如此，C 语言仍不愧为优秀的程序设计语言之一。

1.3 C 程序的基本结构与书写规则

任何一种程序设计语言都具有特定的语法规则和规定的表达方法。一个程序只有严格按照该语言规定的语法和表达方式编写，才能保证编写的程序在计算机中正确地被执行，同时也便于阅读和理解。

1.3.1 C 程序的基本结构

为了说明 C 语言源程序结构的特点，先看以下几个程序。这几个程序由简到难，体现了 C 语言源程序在组成结构上的特点，从中可以了解一个 C 语言源程序的基本组成部分和书写格式。

【例 1.1】 求两个给定整数之和。

源程序如下：

```
1   #include<stdio.h>                          /*预处理命令，标准输入输出头文件*/
2   int main()                                 /*主函数*/
3   {                                          /*函数开始标志*/
4      int a, b, sum;                          /*定义 3 个整型变量 a、b、sum*/
5      a=2;                                    /*给变量 a 赋值 2*/
6      b=8;                                    /*给变量 b 赋值 8*/
7      sum=a+b;                                /*计算 a 和 b 的和，并赋值给变量 sum*/
8      printf("%d+%d=%d!\n", a, b, sum);       /*调用标准函数 printf()输出计算结果*/
9      return  0;                              /*程序正常结束*/
10  }                                          /*函数结束标志*/
```

该程序的执行结果是在屏幕上显示：

```
a+b=10!
```

该程序总共有 10 行，对于每一行的语义已经在行尾进行了说明。它是一个比较简单的 C 语言程序，它的简单在于整个程序只包含了一个 main()函数。它体现了 C 语言程序最基本、最简单的结构。

下面通过例 1.2 对 C 语言程序的结构和格式进行更全面的认识。

【例 1.2】 求两个数中的较大者。

源程序如下：

```
1   #include<stdio.h>
2   int main()                                 /*主函数*/
```

```
3  {
4    int a,b,c;                          /*定义三个整数变量，以备后面程序使用*/
5    int max(int, int);                  /*函数声明*/
6    printf("Input an integer\n");       /*显示提示信息*/
7    scanf("%d", &a);                    /*从键盘获得一个整数 a*/
8    printf("Input an other integer\n"); /*显示提示信息*/
9    scanf("%d", &b);                    /*从键盘获得一个整数 b*/
10    c=max(a, b);                       /*求 a 与 b 的最大值，并把它赋给变量 c*/
11    printf("max=%d\n", c);             /*显示程序运算结果*/
12    return 0;
13  }
/*在主函数之外，自定义函数 max，其功能是求两个整数中较大者*/
14  int   max(int x, int y)
15  {
16    int z;
17    if(x>y)       /*两个参数进行大小比较，较大者赋值给变量 z*/
18        z=x;
19     else
20        z=y;
21     return z;
22  }
```

该程序的运行结果为

```
Input an integer
8
Input an other integer
7
c=8
```

该程序的功能是从键盘输入两个数 a 和 b，求 a 和 b 的最大值 c，然后输出结果。C 语言规定，源程序中所有用到的变量都必须先说明、后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点，与解释型的 BASIC 语言是不同的。说明部分是 C 语言源程序结构中很重要的组成部分。例 1.2 中使用了三个变量 a、b 和 c，用来表示输入的自变量与计算的最大值。程序第 6～11 行为执行部分，或称为执行语句部分，用以完成程序的功能。第 6 行和第 8 行是输出语句，调用 printf 函数在显示器上输出提示字符串，要求输入自变量 a 与变量 b 的值。第 7 行与第 9 行为输入语句，调用 scanf 函数，接受键盘上输入的数并存入变量 a 与变量 b 中。第 10 行是调用函数 int max(int x，int y)计算 a 与 b 的最大值并把它送到变量 c 中，调用时将实际参数 a 和 b 的值分别传给 max 函数中的形式参数 x 和 y，经过执行 max 函数得到一个返回值(max 函数中变量 z 的值)，把这个返回值赋给变量 c。第 11 行是用 printf 函数输出变量 c 的值(a 与 b 的最大值)。第 12

行是一个返回语句，这里是把整数 0 传送给运行该程序的操作系统环境，其作用是把程序成功执行完毕这一消息通知给操作系统，标志着程序的终止。

该例中除主函数 main 外，又定义了一个 max 函数。int max(intx, int y)称为函数头，max 为函数名，int 为函数返回值的类型，圆括号中的 x、y 称为函数参数，其前面的 int 规定了参数的类型。

max 函数体中的“int z;”语句定义了一个临时工作变量，它只在这个函数中使用，作用是接受 x、y 中的最大者，最后返回给调用函数。

通过这两个小程序，可以看到 C 语言程序结构并不复杂。一个简单的 C 语言程序的基本结构和格式有以下几点规定：

(1) 每个 C 语言程序可由一个或多个函数组成，函数是组成 C 语言程序的基本单位。所谓函数，是指具有统一定义、声明、调用格式，且能完成特定功能的程序模块或代码。对于每一个 C 语言程序而言，必须有且只有一个 main()函数，俗称主函数。该函数标志着执行 C 程序时的起点和终点，即主函数是 C 语言程序执行的入口和出口。另外，函数可以划分为由编译系统提供的标准函数(如 printf、scanf 等)和用户的自定义函数(如 max)。

(2) 主函数(包括其他函数)都由函数头和函数体(包括变量定义和语句部分)组成，其格式如下：

```
main()
{
    变量说明;
    语句;
}
```

(3) #include<stdio.h>，是一条预处理命令。这里的 include 称为文件包含命令，其意义是把尖括号(<>)或双撇号("")内指定的文件包含到本程序中，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为 .h，因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型，因此，凡是在程序中调用一个库函数时，都必须包含该函数原型所在的头文件。在本例中，使用了 1 个标准库函数：输出函数 printf。其头文件为 stdio.h，所以在主函数前用 include 命令包含了 stdio.h 文件。所以说一个 C 语言源程序可以由一个或多个源文件组成。预处理命令通常放在源文件或源程序的最前面。

(4) 程序中出现的“/*”与“*/”括起来的部分是程序的说明(称为注释)，它不参与程序的运行，它提高了程序的可读性。注释文字可以是任意字符，如汉字、拼音、英文等。注释语句既可放在语句的右侧，也可单独成为一行。

(5) C 程序中的每条语句都要以分号“;”结束。但预处理命令、花括号和一些特殊情况下不能加分号。

(6) 标示符、关键字之间必须加至少一个空格以示间隔。若已有明显的间隔符，也可不再加空格。C 语言的关键字都以小写字母表示。C 语言中区分字母的大写和小写，例如 else 是关键字，ELSE 则不是。在 C 程序中，关键字不能用于其他目的，即不允许将关键字作变量名和函数名来使用。

1.3.2　函数定义、声明和调用

1．函数的定义

函数的定义应在主函数之外，函数定义的基本格式如下：

```
函数类型　函数名(形式参数列表)
{
    数据说明;
    语句;
}
```

其中：

(1) 函数类型　函数名(形式参数列表)称为函数头。如例 1.2 中的第 14 行：

```
int max(int x，int y)
```

① 函数类型：函数最终返回值的类型，如 max()函数最终返回一个整型数据，所以 max()函数的函数类型为 int，函数也可以没有返回值，则其返回值类型为 void(空类型)。

② 函数名：一个函数的标识。在命名变量名时要尽量做到“见名知义”。

③ 形式参数列表：所谓参数，就是函数所操作的对象。一个函数可以没有参数，一个函数也可以有多个参数。如果有参数，则必须指定其类型；如果有多个参数，则应该用逗号进行分隔。函数定义中出现的参数称为形式参数，简称“形参”。

(2) 用{}括起来的部分称为函数体，它是函数功能的真正实现部分。函数体包括函数体内的数据说明和执行函数功能的语句，花括号“{”和“}”分别表示函数体的开始与结束。例 1.2 中从第 15 行到 22 行是 max()函数的函数体。

2．函数的声明

函数在定义之后、调用之前，必须进行函数声明。函数声明应放在主函数或调用函数之内。例如例 1.2 中第 5 行：

```
int max(int, int);
```

就是一条函数声明语句。函数声明语句和函数定义中的函数头很像，区别就在于在函数声明语句中函数名后面括号中只需保留各参数类型。如果用户自定义的函数在主函数中没有进行函数声明，则程序编译会报错。

3．函数调用

用户自定义函数可以被主函数调用，也可以被其他函数调用。函数调用的一般格式如下：

```
函数名(实际参数列表);
```

例如，例 1.2 中第 10 行：

```
c=max(a,b);
```

就是一条函数调用语句。此语句的功能是调用自定义函数 max()，求出 a 和 b 中的较大者，并赋值给在主函数中定义的变量 c。函数调用中出现的参数称为实际参数，简称实参。

1.3.3 C 程序的书写规则

从便于阅读、理解及维护的角度出发，在书写程序时应遵循以下规则：

(1) 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次结构。“{”和“}”一般与该结构语句的第一个字母对齐，并单独占一行。

(3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写，以便结构更加清晰，增加程序的可读性。

在编程时应力求遵循上述规则，养成良好的编程习惯。

1.4 计算机运算基础(进位计数制、数值转换)

由于计算机内采用二进制表示数据，而 C 程序中经常使用八进制和十六进制来编制程序，因此程序员必须熟悉二进制、八进制和十六进制的运算及它们之间的转换规则。

1.4.1 数的二进制、十进制、八进制和十六进制表示

常用的数制有二进制、十进制、八进制和十六进制，它们有共性也有差别。

1．数码及进位法则

数码是构造一种数制所用的不同符号。各种进制的数码如下：

二进制：0，1

十进制：0，1，2，3，4，5，6，7，8，9

八进制：0，1，2，3，4，5，6，7

十六进制：0，1，2，3，4，5，6，7，8，9，A(a)，B(b)，C(c)，D(d)，E(e)，F(f)

2．位置计数法

数据中各个数字所处的位置决定着其大小(即权值)，同样的数字在不同的位置上代表的权值是不同的。例如十进制数：

$$22.2 = 2 \times 10^{1} + 2 \times 10^{0} + 2 \times 10^{-1}$$

其中，10^{1}、10^{0}、10^{-1}就是该位置的权值。

1.4.2 数制转换

日常习惯中使用的是十进制数，而计算机中用的却是二进制数，所以需要把十进制数转换成二进制数。但二进制数书写麻烦，因此通常也用八进制和十六进制表示，这样就存在各种数制之间的转换问题。

1．将十进制数转换成二进制数

把十进制的整数和小数转换成二进制数时所用方法是不同的，因此应该分别进行转换。

(1) 用余数法将十进制整数转换成二进制整数。把十进制整数不断地用 2 去除，将所得到的余数 0 或 1 依次记为 K_0，K_1，K_2，…，直到商是 0 为止，将最后一次所得的余数记为 K_n，则 $K_nK_{n-1}\cdots K_1K_0$ 即为该整数的二进制表示。在演算过程中可用竖式形式，也可用线图形式。

【例 1.3】　$(59)_{10}=(\quad)_2=(K_nK_{n-1}\cdots K_1K_0)_2$。

竖式演算如下：

余数

2 | 59　　1　　----- K_0

2 | 29　　1　　----- K_1

2 | 14　　0　　----- K_2

2 | 7　　1　　----- K_3

2 | 3　　1　　----- K_4

2 | 1　　1　　----- K_5

0

因此，$(59)_{10}=(111011)_2$。

(2) 用进位法将十进制小数转换成二进制小数。把十进制小数不断地用 2 去乘，将所得乘积的整数部分 0 或 1 依次记为 K_1，K_2，K_3，…。一般情况下，十进制小数并不一定都能用有限位的二进制小数表示，可根据精度要求，转换成一定位数即可。

注意：在把十进制数转换成二进制数时，熟记一些 2 的幂次所对应的十进制数及二进制数能加快转换速度。表 1.1 中列出了一些 2 的幂次所对应的十进制和二进制数。

表 1.1　2 的幂次所对应的十进制数和二进制数

2 的幂次	十进制数	二进制数
2^0	1	1
2^1	2	10
2^2	4	100
2^3	8	1000
2^4	16	10000
2^5	32	100000
2^6	64	1000000
2^7	128	10000000
2^8	256	100000000
2^9	512	1000000000
2^{10}	1024	10000000000
2^{-1}	0.5	0.1
2^{-2}	0.25	0.01
2^{-3}	0.125	0.001
2^{-4}	0.0625	0.0001