普通高等教育“十一五”规划教材
21世纪大学计算机基础分级教学丛书

计算机程序设计基础

C语言程序设计（第二版）

马德骏　张建宏　汤练兵　主编

科学出版社
www.sciencep.com

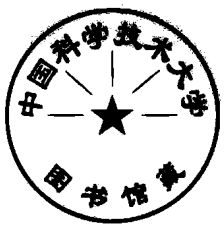
普通高等教育“十一五”规划教材

21世纪大学计算机基础分级教学丛书

计算机程序设计基础
——C语言程序设计

（第二版）

马德骏　张建宏　汤练兵　主编



科学出版社

北京

内 容 简 介

本书为高等院校非计算机专业初级计算机语言教材，主要面向初学程序设计者，介绍了C语言的基本知识、基本算法和基本程序设计方法。本书共有10章，内容分别为：C语言程序设计基础知识选择结构程序设计、循环结构程序设计、数组、函数、指针、结构体、共用体和枚举、文件。考虑到系统平台的发展和程序设计方法的发展，在部分章节中适当兼顾介绍VC++的面向过程部分的程序设计方法，使读者在了解一般的C语言程序设计知识的同时，初步了解面向过程和面向对象开发方式上的差异，为读者今后向面向对象程序设计语言VC++平滑过渡打下基础。

本教材通俗易懂，便于自学，主要针对计算机语言的初学者，适用于各类院校非计算机专业本、专科学生，也可供高等职业技术学院、网络学院、成教学院学生，计算机等级考试者，以及培训班学员、C语言自学者学习使用。

图书在版编目(CIP)数据

计算机程序设计基础：C语言程序设计/马德骏，张建宏，汤练兵主编. —2版. —北京：科学出版社，2009

(21世纪大学计算机基础分级教学丛书)

普通高等教育"十一五"规划教材

ISBN 978-7-03-026204-2

Ⅰ.计… Ⅱ.①马…②张…③汤 Ⅲ.C语言－程序设计－高等学校－教材 Ⅳ.TP312

中国版本图书馆CIP数据核字(2009)第228170号

责任编辑：程 欣/责任校对：梅 莹

责任印制：彭 超/封面设计：苏 波

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

武汉市新华印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2006年8月第 一 版 开本：787×1092 1/16

2009年12月第 二 版

2009年12月第一次印刷 印张：15 1/4

印数：13 001—19 000 字数：350 000

定价：25.80元

(如有印装质量问题，我社负责调换)

前　言

计算机应用水平是当今衡量一个人知识水平和能力的重要标准之一。程序设计课程是计算机基本技能教育和能力培养的一个重要组成部分，它能培养学生利用计算机解决实际问题的思维方式和能力，也能为学生在后继课程以及工作中，应用计算机解决实际问题打下良好的基础。C语言已成为大多数学校理工专业所选的程序设计课程的教学语言，也是在各种计算机证书考试中为大多数人所选择的程序设计语言。

通过长期的C语言教学和对第一版教材使用过程中经验的总结以及教学规律和教学内容的更新需求，编者对第一版的内容进行了部分调整，其目的就是要通过内容调整，适应教学规律，突出重点，强调教学连贯性，有利于加深对课堂知识的理解拓宽、提高能力。《C语言程序设计教程》(第二版)主要是将有关指针的内容从开始就渗透到各个章节，使学生能逐步地深入理解和掌握指针的相关内容。

本书是专为初学程序设计者编写的教材，共有10章，介绍了C语言的基本知识、基本算法和基本程序设计方法，它的内容分别为：C语言程序设计基础知识选择结构程序设计、循环结构程序设计、数组、函数、指针、结构体、共用体和枚举、文件。本教材在力求通俗易懂的同时兼顾对难点以及基本概念的解疑释惑，相关章节中安排有可选内容(*)，供读者参考，这样便于教师在教学活动中根据具体情况灵活取舍。由于考虑到系统平台的发展和程序设计方法的发展，在部分章节中适当兼顾介绍VC++的面向过程部分的程序设计方法，目的是使读者在了解一般的C语言程序设计知识的同时，初步了解面向过程和面向对象开发方式上的差异，为读者今后向面向对象程序设计语言VC++平滑过渡打下基础。在VC++集成环境中可以弥补TC环境下编辑功能的不足，突破TC环境下的汉字系统平台限制。

对于C语言程序设计的学习，除了理论教学外，习题和实验环节是必不可少的，在与本教材配套的《C语言程序设计实验与习题》(第二版)一书中给出了大量精心设计的习题及其参考答案和实验内容以配合对本教材各个知识点的学习和掌握。

本书主要针对计算机语言的初学者，适用于各类院校非计算机专业本、专科学生，也可供高等职业技术学院、网络学院、成教学院学生，计算机等级考试者，以及培训班学员、C语言自学者学习使用。依不同专业和层次，教师可灵活掌握深广度。

本书由马德骏、张建宏、汤练兵主编。第一、二章由张建宏编写，第三、九章由汤练兵编写，第四、五、六、七章由马德骏编写，第八章由陈志铭编写，第十章由杨朝阳编写，附录部分由李捷编写整理。参加本书编写和程序调试工作的还有郑敬、段翠萍、孙骏、李宁等。马成前、王舜燕、汤英等参与了本教材的大纲编写工作。

由于编者水平和时间的限制，书中难免存在不少缺点和不足，敬请读者和同行专家不吝赐教。

编　者

2009年11月

目　录

第1章 概 论

1.1 C语言基本知识

1.1.1 C语言的发展历史及特点

C语言是目前广泛流行的一种计算机高级语言，由美国贝尔实验室的D Ritchie于1972年编写并在PDP11计算机上实现。实际上C语言是在一系列高级语言的基础上发展而来的，其根源可以追溯到20世纪60年代剑桥大学的CPL语言、其后的BCPL语言、70年代贝尔实验室的B语言。C语言问世后，其自身也得到不断的发展，标准C、ANSI C、87 ANSI C的语言标准相继出现。其中87 ANSI C在1990年成为ISO(国际标准化组织)的ISO C的标准。C语言也移植到各种不同的机器上，现在C语言已成为世界上最为广泛流行的计算机高级语言之一。

C语言的语言精练简洁，表达方式丰富灵活，代码质量高，可移植性好。其主要特点概括起来有以下几点：

(1) C语言程序结构紧凑、语言简洁，一共只有32个关键字、9种控制语句，用C语言编写的程序可读性强，编译效率高。

(2) C语言的数据类型丰富，有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等，用来表示各种复杂的数据结构，以适应不同的编程需要。

(3) C语言的运算符丰富，多达44种(见附录II)。将它们与丰富的数据类型相结合，使得C语言的表达方式非常灵活，编译效率非常高，这一点是其他高级语言所不能及的。

(4) C语言是一种结构化程序设计语言。因为C语言程序由函数(函数作为模块化设计的基本单位)构成，同时一个C语言程序可由多个C语言源程序文件组成，这些源文件可独立编制、编译，然后再将它们连接成可执行的目标程序。因此，C语言适宜用来编制大型软件。

(5) C语言是处于汇编语言和高级语言之间的程序设计语言，即中级语言。它既具有高级语言的基本特征，又具有汇编语言的功能。即它将高级语言所具有的面向用户、易编程和易维护、可读性强与汇编语言的面向硬件、面向系统、能进行位(bit)操作、直接访问物理地址的功能有效地结合起来，使它既能用来编制一般的应用软件，又能用来编制系统软件。

(6) C语言可移植性好。虽然C语言具有汇编语言的一些功能，涉及硬件的操作，但是由于它的这些操作都是通过函数来调用操作系统的功能。因此，能够很方便地在不同硬件的计算机之间进行移植。

(7) C语言的语法限制不太严格。这一特点对初学者来说是一个“难点”，而对一个有经验的程序员来说，这为他们提供了一个施展其“技巧”的空间。

由于C语言具有上述主要特点，近年来它得到了越来越广泛的应用。初学者对于上

述特点可能不太理解，随着学习的不断深入，会逐步加深印象、充分理解，并且可以利用这些特点编制出“漂亮”的程序。

1.1.2 C语言的标识符与关键字

在C语言程序中，标识符和关键字都属于程序的基本语法单位。正确地使用标识符和关键字对于程序的编制是至关重要的。

1. 标识符

标识符一般是指用户或系统定义的符号名、变量名、数组名、类型名、函数名、文件名等。C语言规定标识符由字母、数字、下划线组成，必须以字母或下划线开头。大写字母小写字母被认为是不同的字符，在标识符中不能含有其他字符，也不能跨行书写。

例如，合法的标识符如下：

a_1　name　_x123　worker　World_1　_Wolf

非法的标识符如下：

1abc　name, 1　$x_{\sqcup}123n$　&a12　To-me　date..1

ANSI C标准没有规定标识符的长度(字符个数)，但不同的C的编译系统有各自的规定：MS C规定标识符长度不能超过8个字符，长度大于8个字符的取前8个字符作为标识符；Turbo C规定标识符长度不超过32个字符，超过32个字符的部分无效。

用户在定义标识符的时候应注意：

(1) 不要与系统预定义的标识符或关键字相同。如：用户不要定义printf作为用户标识符，因为系统已经将其定义为格式输出函数的标识符，以免引起冲突。

(2) 尽可能直观地定义标识符。也就是说，从所定义的标识符的字面上就能够了解其含义，以方便以后的使用。

2. 关键字

C语言中的关键字又称保留字，它是由C的编译程序预定义的、具有特定含义的单词，用户不可将其定义成自己的标识符使用。C的关键字有以下32个：

auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	int	long
register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void
volatile	while				

需要指出的是，这些关键字都是小写字母，C语言对大小写敏感，即不能将关键字中的任何字符改为大写，否则，C编译不将其作为关键字处理。

1.1.3 C语言的基本结构

(1) C语言程序由一个或多个函数组成，其中有且只有一个主函数，名为main( )。

(2) 函数由函数说明和函数体组成。其中函数名命名应符合标识符的规定，函数体

应以“{”开始，并以“}”结束。

(3) 函数体由C语言的“语句”组成，每条语句必须以“;”结尾。

(4) C语言程序的语句书写格式自由，一行内可以写一条或多条语句，也可将一条语句写在多行上，但关键字、标识符、字符串作为一个整体不能分写在两行上。

(5) 在C语言程序中，可以在任何位置成对地使用“/*”和“*/”对程序进行注释，注释的作用是利用文字信息对程序进行说明或注解，以便人们在阅读程序时理解程序的功能，注释部分在程序的运行时不执行。

例1.1 在屏幕上显示“Hello, C!”这串字符。

```
main()                          /*主函数*/
{                               /*函数体开始*/
  printf("Hello, C!");          /*在屏幕上显示字符串*/
}                               /*函数体结束*/
```

说明：程序中第一行main是系统预定义的标识符，它是主函数名。第2行和第4行是表示函数体的开始和结束。**注意**：“{”和“}”必须成对使用。第3行是函数体，在本例中，函数体只有一条库函数的调用语句，以“;”结束，它调用了C函数库中的输出函数printf，同样printf是系统预定义的标识符，括号中双引号内是要输出的字符串。

例1.2 求和运算。

```
main()                          /*主函数*/
{                               /*主函数体开始*/
  int a, b, c;                  /*定义3个整型变量名*/
  printf("Hello, C!");          /*屏幕输出字符串*/
  scanf("%d%d", &a, &b);        /*从键盘输入两个数给a和b这两个变量*/
  c=add(a, b);                  /*调用用户定义的函数add,并将函数值置于变量c中*/
  printf("%d\n", c);            /*屏幕输出变量c的值*/
}                               /*主函数体结束*/
add(a, b)                       /*用户自定义标识符add作为函数名,有两个参数a和b*/
{                               /*用户自定义函数的函数体*/
  int y;                        /*定义一个整型变量名y*/
  y=a+b;                        /*将a+b的值置于y中*/
  return y;                     /*将y的值返回*/
}                               /*用户自定义函数的函数体结束*/
```

本例中，除了主函数main以外，增加了用户自定义函数add，它的功能是进行加法运算，并返回运算结果。当运行程序时，计算机按照语句的顺序逐条执行每条语句，当执行到c=add(a, b);语句时，转而执行用户自定义函数add，并将函数值置于变量c中。值得注意的是函数add的功能由其函数体的语句序列所决定，如果我们将其改成其他语句序列，则其功能也将发生变化。初学者现在不能理解某些具体语句的含义，这不重要，重要的是要注意到“不同的函数具有不同的功能，用户可以通过函数体中的语句序列来实现这些功能，将这些函数按照C语言的规则组合起来就成为了C语言程序”。

加深对C语言程序基本结构的理解，对今后的学习会有很大帮助。另外，关于如何运行一个C程序，可参看配套教材《计算机程序设计基础——C语言程序设计实验与习

题》有关章节，这里不再叙述。建议初学者除了注意对书本知识的学习以外，要多观察、多思考、多做自己的“作品”(不在于复杂程度，只要是自己的)、多做上机练习，坚持一段时间，培养起自己的兴趣，有了兴趣就意味着成功的开始。

1.2 算法及其表示

1.2.1 算法的概念和特点

使用计算机解决实际问题的时候需要编写程序。程序是可连续执行并完成特定功能的一系列指令的集合，主要包含两个部分，即数据结构和算法。数据结构就是在程序中要对处理的对象(数据)进行描述，指定数据的类型和组织形式；算法就是对操作步骤的描述。一个高效实用的程序与算法的设计、数据结构的描述、程序设计方法的选择、语言工具以及环境条件关系密切。

使用计算机解决实际问题的过程通常包括以下几个步骤：

(1) 分析问题，找出解决问题的模型。

(2) 根据模型设计出适合计算机特点的处理方法(即算法)。

(3) 选择适合的计算机语言进行编程，以实现算法。

(4) 上机编辑、调试、运行所编制的程序，得到结果。

(5) 对结果进行分析，整理出文字材料(即文档)。

在实际操作过程中，上述步骤可能会出现反复。需要指出的是，算法的好坏直接影响到程序的编制、运行效率以及程序的可读性和可维护性。

算法就是为解决某一问题，对数据进行操作和处理所采取的一系列步骤。算法可分为两大类，一类为数值数据运算并求解的算法；另一类为非数值数据运算及其处理的算法。前者用来描述一个求数值解的过程，如求定积分、微分方程求解、高阶方程求解等；后者用来描述非数值求解的过程，如情报检索、事务处理、数据管理等。

注意：在计算机应用方面，非数值运算的应用大大多于数值运算的应用。一个合理的算法具有以下特点：

(1) 有穷性。一个算法应该是“有限”个步骤的，而不能是“无限”个步骤的。即经过有限个步骤的处理以后，算法应该结束。

(2) 确定性。算法中的每一个步骤的含义都是确定的、唯一的，不能具有其他的含义或可能被理解成其他的含义。

(3) 有输入。可有零个或多个输入。输入是用来在一个算法的执行过程中，向它提供处理对象(数据)或控制算法执行过程的信息的。

(4) 有输出。可有一个或多个输出。它是算法执行的结果的输出。没有输出的算法是一个无效的算法。

(5) 有效性。算法的每一步骤都可有效地执行。

1.2.2 算法的表示

表示一个算法的方法很多，如自然语言法、传统流程图法、N-S 流程图法、伪代码法等，理论上都可用来表示算法，但是效率上有很大差异。

例如，M=5!，求 M 的值。

自然语言法描述算法如下：

(1) 设定变量 M，M 初值为 1，设定变量 I，初值为 1。

(2) 如果 I 的值小于 6，则执行(3)，否则执行(4)。

(3) 将 M 乘以 I 并置于 M 中存放，将 I 中的值加 1 并置于 I 中存放，再执行(2)。

(4) 将 M 中的值输出。

用传统流程图法描述算法如图 1-1 所示；用 N-S 流程图描述算法如图 1-2 所示。

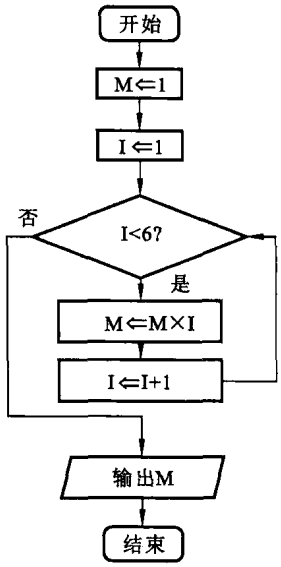


图 1-1 传统流程图

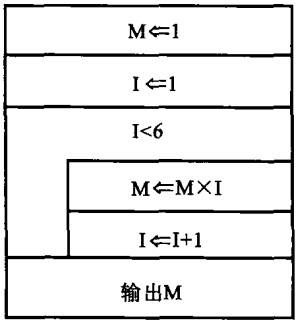


图 1-2 N-S 流程图

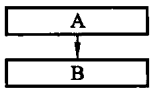
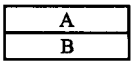
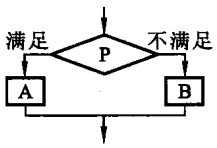
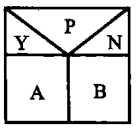
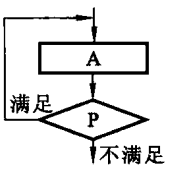
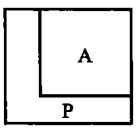
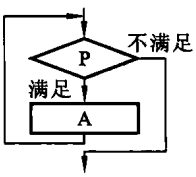
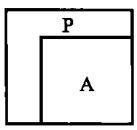
说明：传统流程图使用几何图形、流程线、文字说明(表 1-1)来描述一个算法，它的好处是直观、易懂，便于初学者掌握。但是由于在使用传统流程图描述一个算法时，可以根据需要不受限制地使用流程线，这样设计出来的算法结构性不好，会导致阅读和编程的困难，编制出来的程序结构性不好，不利于以后的使用和维护。

表 1-1 传统流程图基本图形及其含义

图 形	名 称	说 明
	处理框	表示确定的处理或步骤，又称矩形框
	判断框	允许有一个入口，两个或两个以上的可选择的出口
	输入输出框	表示数据的输入或经处理后结果的输出
	起始结束框	表示算法的开始或结束
○	连接点	用于将不同的流程线连接起来
	功能调用框	表示调用一个处理过程
→	流程线	表示算法中处理流程的走向
……[	注释框	用于书写注释或说明信息

人们经过长期的实践，不断地总结经验，提出了结构化程序设计方法，将算法的描述归纳为三种基本结构的顺序组合，即顺序结构、选择结构、循环结构(三种基本结构对照表见表 1-2)。它们的共同特点是：只有一个入口，只有一个出口，每个基本结构中的每一部分都有机会被执行，结构内部不存在死循环。使用三种基本结构描述的算法是结构化的算法，按照结构化算法编写出来的程序具有良好的可读性和可维护性。

表 1-2　三种基本结构对照表

传统流程图	结构名称	N-S 流程图
	顺序结构	
	选择结构	
	直到型循环(后测试循环)	
	当型循环(前测试循环)	

为了适应“结构化”算法的要求，美国学者 I Nassi 和 B Shneidermen 提出了用 N-S 流程图(又称盒图)描述算法的方法。在 N-S 流程图中，取消了流程线，全部算法由一些基本的描述三种基本结构的矩形框图顺序排列组成一个大矩形而成。N-S 流程图特别适合描述一个结构化的算法，用于结构化程序设计。

有关流程图在算法设计中的应用，本书会在以后章节逐步加以介绍。

*1.3　数制与编码

1.3.1　数制

在日常生活中，十进制是人们习惯使用的数制，而在计算机中主要使用二进制，这是由计算机的物理特性所决定的。二进制具有数位少、容易表示、运算简单可靠、便于物理实现、节省设备等优点。所有计算机系统中的信息均以二进制编码形式表示、保存、传输和处理，由此可见二进制在计算机系统中的作用是非常重要的。以下简要介绍计算机系统中常见的几种不同数制系统的概念、表示方法、它们之间的转换以及运算等有关知识。

数制也称为计数制，是指用一组固定的符号和统一的规则来表示数值的方法。按进位的方法进行计数，称为进位计数制。

在进位计数制中有基数、数位、位权三要素。

基数是指在采用进位计数的数值系统中，如果用 R 个基本符号（即数码或数符）表示数值，则称其为基 R 数制（radix-R number system），R 称为该数制的基数（radix）。计数规则为每个数位计满 R 就向高位进一，即逢 R 进一。

数位是指数码在一个数中所在的位置（记为 n，n 为整数）。

位权是指在某种进位计数制中，每个数位上的数码所代表的数值的大小，等于在这个数位上的数码乘以一个固定的数值，这个固定的数值就是这种进位计数制中该数位上的位权，第 n 位数的位权为 R^n。

一般地，任意一个有 n 位整数和 m 位小数的 R 进制数 N 可以表示为

$$N=(k_{n-1}k_{n-2}k_{n-3}\cdots k_1k_0k_{-1}k_{-2}\cdots k_{-m}) \tag{1-1}$$

或

$$N=\sum_{i=n-1}^{-m}k_i\times R^i \tag{1-2}$$

这里，R 为基数（$R\geqslant 2$ 的整数），R^i 为第 i 位数位的位权值。k_i（$i\in[-m,\ n-1]$）为该进位计数制使用的某个数码（$0\leqslant k_i<R$），即 k_i 值只能是 0 到 R−1 这些数字符号中的一个，如在十进制数制中，k_i 的取值只能是 0～9 之间的某一整数。

（1）十进制数。十进制数制是进位计数制，有 0，1，2，3，…，9 这 10 个数字符号（数码），位权为 10^i（$i\in[-m,\ n-1]$）。

在一个十进制数中，数字（数码）所在位置（数位）不同，则所代表的数值也不同。由式（1-2），任何一个十进制数可以展开成为一个不同数位上的数字（数码）乘以该数位的位权值的和。例如，345 中的 3，处于百位上，权值为 10^2，即用 3×10^2 表示。以此类推，如将 234.56 写成按权展开式，即

$$234.56=2\times10^2+3\times10^1+4\times10^0+5\times10^{-1}+6\times10^{-2}$$

在进行加、减法运算时，十进制遵循“逢十进一，借一当十”的规则。

（2）二进制数。二进制也属于进位计数制。二进制仅有 0 和 1 两个数字符号，位权为 2^i（$i\in[-m,\ n-1]$）。例如，二进制数 $(1110.1)_2$ 按式（1-2）展开表示为

$$1\times2^3+1\times2^2+1\times2^1+0\times2^0+1\times2^{-1}$$

在进行加法、减法运算时，二进制遵循“逢二进一，借一当二”的规则。

例如，对于 $(1110.1)_2+(1100.1)_2$ 和 $(1110.1)_2-(1101.1)_2$，如同十进制加减法运算：首先，小数点对位，而后，逐位相加或相减。**注意：要遵循二进制的规则。**

$$\begin{array}{r} 1110.1 \\ +1100.1 \\ \hline 11011.0 \end{array} \qquad \begin{array}{r} 1110.1 \\ -1101.1 \\ \hline 0001.0 \end{array}$$

至于二进制乘法和除法，与十进制乘除法类似，这里不再叙述。

（3）八进制数。八进制有 0，1，2，3，4，5，6，7 这 8 个数字符号，权值为 8^i（$i\in[-m,n-1]$）。例如，八进制数 $(203.1)_8$ 按式（1-2）展开表示为

$$2\times8^2+0\times8^1+3\times8^0+1\times8^{-1}$$

进行加法、减法运算时，八进制遵循“逢八进一，借一当八”的规则。

(4) 十六进制数。十六进制有0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F这16个数字符号，权值为16^i ($i\in[-m,n-1]$)。例如，十六进制数$(2A3.1)_{16}$按式(1-2)展开表示为

$$2\times16^2+A\times16^1+3\times16^0+1\times16^{-1}$$

注意：在十六进制中，A～F这5个数字符号分别代表的数值是十进制的10～15，且大小写等价。

在进行加法、减法运算时，十六进制遵循“逢十六进一，借一当十六”的规则。

(5) 不同数制间的转换。

1) 二进制、八进制、十六进制数转换为十进制数。转换时，通常按位权法进行，位权值按十进制运算法则计算，并将各位上的数码(按十进制值)乘以该位的权值再进行累加所得到的数就是相应的十进制数。如：

$$(1110.01)_2=1\times2^3+1\times2^2+1\times2^1+0\times2^0+0\times2^{-1}+1\times2^{-2}=(14.25)_{10}$$

$$(125.11)_8=1\times8^2+2\times8^1+5\times8^0+1\times8^{-1}+1\times8^{-2}=(85.140625)_{10}$$

$$(b1.2)_{16}=11\times16^1+1\times16^0+2\times16^{-1}=(177.125)_{10}$$

2) 二进制、八进制、十六进制数之间的转换。

• 二进制数转换成八进制或十六进制数 因为二进制、八进制、十六进制之间有内在的对应关系：$2^3=8$，$2^4=16$。3个二进制位所能表示的最大整数为7，最小非负整数为0；4个二进制位则相应为15和0。由于有这种对应关系的存在，因此在将一个二进制数转换为八进制数时，对其整数部分从小数点开始向左三位一组，进行划分，高位不够三位时，左边补0；对其小数部分从小数点开始向右三位一组，低位不够三位时，右边补0；然后，再将各组二进制数分别转换成八进制数即可。将一个二进制数转换为十六进制数时，也采用上述方法，只是分组时是4个二进制位一组即可。如：

$$(1010111.01111)_2=(001\ 010\ 111.011\ 110)_2=(127.36)_8$$

$$(1011111.01111)_2=(0101\ 1111.0111\ 1000)_2=(5F.78)_{16}$$

• 八进制数、十六进制数转换成二进制数 将八进制数转换成二进制数时，只需将八进制数的各个数字分别用3个二进制数位来表示；将十六进制数转换成二进制数时，只需将十六进制数的各个数字分别用4个二进制数位来表示即可。如：

$$(32.12)_8=(011\ 010.001\ 010)_2=(011010.001010)_2$$

$$(9A.0C)_{16}=(1001\ 1010.0000\ 1100)_2=(10111010.00001100)_2$$

说明：在将各个数字转换为三位(或四位)二进制数位表示的过程中，若某个数字变换后不够三位(四位)，则要补0以凑够三位(四位)。例如，将八进制数$(2)_8$转换成二进制数时，应为$(010)_2$，而不能是$(10)_2$。若使用不当，则有可能产生错误。

• 八进制数、十六进制数间的转换 通常，它们之间的转换是利用二进制数作为过渡进行变换的，即先将一种数据转换成二进制数，再将该二进制数转换成另一种数据。如：

$$(A5)_{16}=(1010\ 0101)_2=(010\ 100\ 101)_2=(245)_8$$

3) 十进制数转换成二进制、八进制、十六进制数。

• 十进制数转换成二进制数 将十进制数转换成二进制数时，采用分别对整数部分

和小数部分进行转换的方法。

对于整数部分，采用“除二取余”的方法：将该十进制数整数部分除以 2，所得余数作为二进制数整数部分的最低位，也就是小数点左边第一位（个位）；再将商作为被除数除以 2，所得余数作为小数点左边第二位。以此类推，可以得到一系列的余数，按其先后次序，自小数点起，从右向左排列，就构成了二进制数的整数部分。对于小数部分，采用“乘二取整”的方法：将该十进制数小数部分乘以 2，取积的整数部分，作为二进制数小数点后的第一位；再将积的小数部分乘以 2，取其整数部分，作为二进制数小数点后第二位。以此类推，就可得到二进制数的小数部分。将整数部分和小数部分合并，即得到对应的二进制数。

例如，求$(28.125)_{10}$所对应的二进制数，处理过程如下：

余数

2	28	0……最低位	0.125×2=0.25	……0	最高位
2	14	0	0.25×2=0.5	……0	
2	7	1	0.5×2=1.0	……1	最低位
2	3	1	小数部分：(0.125)10=(0.001)2		
	1	……最高位			

整数部分：(28)10=(11100)2

合并整数部分和小数部分后得到：$(28.125)_{10}=(11100.001)_2$。

• 十进制数转换成八进制、十六进制数 原则上与十进制数转换成二进制数的方法一样，只是除法运算时是除以 8 或 16 取余，乘法运算时是乘以 8 或 16 取整。建议实际转换时，先转换成二进制数，再由二进制数转换成八进制或十六进制数，这样做简单一些。

例如，求$(10.25)_{10}$所对应的八进制数和十六进制数。

$(10.25)_{10}=(1010.01)_2=(001010.010)_2=(12.2)_8=(1010.0100)_2=(A.4)_{16}$

说明：在进行小数部分转换过程中二进制数位分组时，低位不足三位（转换成八进制）或四位（转换成十六进制），应补 0 凑够三位（四位）后再转换。例如，将二进制数$(.01)_2$转换成十六进制数时，要先写成$(.0100)_2$，再转换成$(.4)_{16}$。若不补 0 凑成四位二进制位而直接转换，则可能会产生错误。

1.3.2 编码

计算机中的数据分为两大类：数值型数据和非数值型数据。数值型数据是指可以进行算术运算的数据，如$(256)_{10}$、$(0111.101)_2$等都是数值型数据；非数值型数据通常不参加算术运算，如字符串“你好，世界”、“2002.6.15”等，就是典型的非数值型数据。

(1) 数值型数据的表示。在计算机中表示一个数值型数据，首先需要解决数的长度、符号和小数点位置的表示等问题。通常，在计算机中，使用固定长度的二进制位数来表示数值型数据，如：用 8 个、16 个、32 个二进制位来表示一个数值型数据。在计算机中，数值型数据的小数点位置总是隐含的，以便节省存储空间。隐含的小数点位置可以是固定的，也可以是可变的。前者称为定点数(fixed point number)，后者称为浮点数(floating point number)。