



高等学校精品规划教材

案例式教学

# C/C++程序设计学习辅导

李婷 李云峰 编著

A large graphic of the text 'C/C++' in a blue, serif font. The background is a light blue gradient with faint binary code (0s and 1s) and glowing lines, suggesting a digital or programming theme.

C/C++



中国水利水电出版社  
www.waterpub.com.cn

21 世纪高等学校精品规划教材

# C/C++程序设计学习辅导

李 婷 李云峰 编著

## 内 容 提 要

本书是李云峰、李婷编著《C/C++程序设计》的配套辅助教材，分为两篇。第一篇为预备知识，分为2章；第二篇为知识技能，其内容与主教材相对应，分为10章。

“预备知识”篇包括第1章“计算机基础知识”，为教学提供必要参考；第2章“C/C++编程环境与调试”，为实训提供指导。

“知识技能”篇包括C/C++程序设计概述、数据类型与运算、结构化程序设计、利用函数编程、利用数组编程、利用指针编程、利用构造类型编程、文件操作、C++程序设计、综合应用程序设计等内容。每章由“编程指导”、“习题解析”、“实训指导”、“知识拓展”4部分内容组成。

本书既是《C/C++程序设计》的辅导教材，也可以单独作为C/C++语言学习辅导用书，是一套难得的课程教学、等级考试、自主学习的好书。

主教材习题中编程题的解题参考程序和综合应用程序设计的参考程序可以从中国水利水电出版社网站以及万水书苑免费下载，网址为：<http://www.waterpub.com.cn/softdown/>或<http://www.wsbookshow.com>。

## 图书在版编目(CIP)数据

C/C++程序设计学习辅导 / 李婷, 李云峰编著. --  
北京: 中国水利水电出版社, 2012.8  
21世纪高等学校精品规划教材  
ISBN 978-7-5170-0023-5

I. ①C… II. ①李… ②李… III. ①  
C语言—程序设计—高等学校—教学参考资料 IV.  
①TP312

中国版本图书馆CIP数据核字(2012)第173787号

策划编辑: 雷顺加 责任编辑: 李炎 加工编辑: 李刚 封面设计: 李佳

书 名	21世纪高等学校精品规划教材 C/C++程序设计学习辅导
作 者	李婷 李云峰 编著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: <a href="http://www.waterpub.com.cn">www.waterpub.com.cn</a> E-mail: <a href="mailto:mchannel@263.net">mchannel@263.net</a> (万水) <a href="mailto:sales@waterpub.com.cn">sales@waterpub.com.cn</a>
经 售	电话: (010) 68367658 (发行部)、82562819 (万水) 北京科水图书销售中心(零售) 电话: (010) 88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	三河市铭浩彩色印装有限公司
规 格	184mm×260mm 16开本 19印张 480千字
版 次	2012年8月第1版 2012年8月第1次印刷
印 数	0001—3000册
定 价	35.00元

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

版权所有·侵权必究

# 前 言

C/C++语言是一种非常出色的程序设计语言，也是国内外广泛使用的计算机语言。C/C++语言的最大特点是功能强、设计思路灵活多样，编制出来的程序代码短小精练，已经被广泛应用于计算机应用程序开发和计算机课程专业教学等领域。国内外许多高校都将C/C++语言列为大学生学习编程的首选语言，在我国被列为全国计算机等级考试二级的主考语种之一。

本书作者多年从事高校程序设计语言课程的教学，亲身感受到学生在学习过程中遇到的各种困难所在，了解学生迫切需要C/C++语言编程学习辅导用书及参加计算机等级考试的备考复习资料，为此，编写了《C/C++程序设计学习辅导》。本书是对主教材内容的扩充和拓展，它不仅包括了学习C/C++语言中的“困难”辅导，而且涵盖了《全国计算机等级考试二级C语言程序设计考试大纲》的有关内容，为读者学习本课程和参加等级考试提供了方便。

为了便于教学组织和实施，本书分为两篇。

第一篇为“预备知识”，分为两章，第1章为“计算机基础知识”，以适应参加等级考试的需要。第2章为“C/C++编程环境与调试”，以适应课程实训的需要。

第二篇为“知识技能”，它与主教材内容同步，分为10章。每章分为4部分：

第1部分为“知识要点”。包括“考点内容提要”和“常见编程错误”。其中“考点内容提要”系本章知识重点，内容包括该章的基本概念和语法结构知识。“常见编程错误”是列举因为对基本概念和语法结构知识掌握不牢或忽视某些知识点而导致的编程错误。换句话说，编程错误是对“考点内容提要”内容掌握不牢的必然结果。因此，通过“考点内容提要”学习，巩固本章的知识要点；通过对“常见编程错误”解析，进一步加深对基本概念的理解和对语法规则的掌握。

第2部分为“习题解析”。包括问答题、选择题、填空题。其中选择题与填空题选自近年来全国计算机等级考试二级C语言的笔试真题。这不仅有利于全面掌握本章的知识内容，也有利于参加国家等级考试。

第3部分为“实训指导”。包括程序验证、程序填空、程序修改和程序设计四个方面的内容。其中，程序填空、程序修改和程序设计均选自近年来全国计算机等级考试二级C语言机试真题。

第4部分是“知识拓展”。它是对主教材教学内容的补充和拓展，如相关理论知识的拓展、相关算法的拓展、相关技术的拓展。掌握这部分内容对参加全国计算机等级考试是非常重要的。

这样，主教材及其习题与等级考试的笔试完全吻合；辅助教材及其实训与等级考试的机试完全吻合，构成了一个完整的知识、技能体系。

为了检验教与学的效果，本书设计了检查各章教学效果的教学评价表，教学评价贯穿于课程始终，方便学生自我检测和评价掌握知识的程度。

科学的教学评价体系是实现课程教学目标的重要保障。教学评价包含了两个方面：一是学生的自我评价，以检验学生学习的三种能力（理解能力、技术能力和实践能力）。二是对教师教学能力和态度的评价，以检验教师教学的四个方面（教学方法、教学手段、教学经验

和教学态度)。因此,教学评价既是学生对掌握各章理论知识和实践动手能力的自我检测,也是对教师教学方法和教学水平的检测。通过对教与学的评价,不断改进“学”与“教”的方法,强化“学”与“教”的效果。

此外,为了便于读者学习和参加等级考试,本书中提供了2个附录:附录A列出了 Turbo C++ 3.0 环境下编译错误信息汇总;附录B给出了全国计算机等级考试二级 C 语言考试大纲。

本书与主教材一起,构成了一个完整的知识、技能体系,具有内容丰富、系统性好、适应性强的特点。本套教材采用“案例式、解析法”的教学方法,它是作者多年来教学方法研究的结晶。

“案例式、解析法”的思想是从实际案例(问题描述)入手,剖析求解问题的关键点(算法和解题思路),给出问题的求解方法和实现过程,然后结合问题讲解需要的知识点,通过“程序实例”列举典型问题的求解方法和实现过程,通过“问题点拨”和“算法评价”,达到触类旁通的教学效果。

本书由李婷博士(副教授)和李云峰教授编写。曹守富老师为本书校稿、程序调试、课程网站建设做了大量工作;丁红梅、周国栋、刘艳、刘冠群、谭阳、方颂、陆燕等老师参加了课程教学资源建设。

在本书编写过程中,参阅了大量近年来出版的国内外同类教材以及全国计算机等级考试二级 C 语言考试题库资源,并从中吸取了许多有益的营养,特别是湖南理工学院杨克昌教授为本书提供了极为珍贵的文献资料,在此,谨向这些著作者一并表示衷心感谢!

本书凝聚了作者多年教学、科研以及软件开发的经验和体会,尽管我们希望做到更好,但因作者水平和成书时间所限,书中难免有许多不足之处,敬请专家和读者批评指正。

作者  
2012年6月

# 目 录

前言

## 第一篇 预备知识

第1章 计算机基础知识	1	第2章 C/C++编程环境与调试	21
§1.1 用计算机求解 $f(x)=ax^2+bx+c$	1	§2.1 Turbo C++ 3.0 编程环境	21
1.1.1 基本任务	1	2.1.1 Turbo C++ 3.0 的安装	21
1.1.2 涉及的问题	2	2.1.2 输入和编辑源程序	23
§1.2 数制及其转换	3	2.1.3 编译、连接和运行	25
1.2.1 常用进位计数制	3	2.1.4 建立文件工作区域	28
1.2.2 数制之间的转换	4	§2.2 Visual C++ 6.0 编程环境	31
1.2.3 数值数据的四则运算	7	2.2.1 Visual C++ 6.0 的安装	31
1.2.4 二进制数的逻辑运算	9	2.2.2 输入和编辑源程序	32
1.2.5 数值数据的编码表示	11	2.2.3 编译、连接和运行	34
§1.3 计算机硬件系统	15	2.2.4 建立文件工作区域	39
1.3.1 硬件系统的结构组成	15	§2.3 程序错误类型与查找方法	45
1.3.2 计算机主机	15	2.3.1 程序错误类型	46
§1.4 计算机软件系统	17	2.3.2 查找错误的基本方法	50
1.4.1 什么是软件	17	§2.4 程序调试方法	52
1.4.2 软件的功能	18	2.4.1 在 Turbo C++ 3.0 中调试程序	52
1.4.3 软件的分类	18	2.4.2 在 Visual C++ 6.0 中调试程序	56

## 第二篇 知识技能

第1章 C/C++程序设计概述	60	1.4.1 算法的设计要求	69
§1.1 知识要点	60	1.4.2 算法的评价指标	70
1.1.1 考点内容提要	60	教学评价	72
1.1.2 常见编程错误	61	第2章 数据类型与运算	74
§1.2 习题解析	63	§2.1 知识要点	74
1.2.1 问答题	63	2.1.1 考点内容提要	74
1.2.2 选择题	64	2.1.2 常见编程错误	75
1.2.3 填空题	66	§2.2 习题解析	79
§1.3 实训指导	67	2.2.1 问答题	79
1.3.1 实训规划	67	2.2.2 选择题	80
1.3.2 实训内容	68	2.2.3 填空题	84
§1.4 知识拓展	69	§2.3 实训指导	85



2.3.1 实训规划 .....	85	5.1.1 考点内容提要 .....	139
2.3.2 实训内容 .....	85	5.1.2 常见编程错误 .....	140
§2.4 知识拓展 .....	89	§5.2 习题解析 .....	143
2.4.1 数据结构的基本概念 .....	89	5.2.1 问答题 .....	143
2.4.2 数据结构研究的问题 .....	90	5.2.2 选择题 .....	144
教学评价 .....	92	5.2.3 填空题 .....	146
<b>第3章 结构化程序设计</b> .....	<b>93</b>	§5.3 实训指导 .....	149
§3.1 知识要点 .....	93	5.3.1 实训规划 .....	149
3.1.1 考点内容提要 .....	93	5.3.2 实训内容 .....	149
3.1.2 常见编程错误 .....	94	§5.4 知识拓展 .....	154
§3.2 习题解析 .....	96	5.4.1 查找算法 .....	154
3.2.1 问答题 .....	96	5.4.2 排序算法 .....	156
3.2.2 选择题 .....	98	5.4.3 集合 .....	161
3.2.3 填空题 .....	101	教学评价 .....	163
§3.3 实训指导 .....	103	<b>第6章 利用指针编程</b> .....	<b>165</b>
3.3.1 实训规划 .....	103	§6.1 知识要点 .....	165
3.3.2 实训内容 .....	104	6.1.1 考点内容提要 .....	165
§3.4 知识拓展 .....	108	6.1.2 常见编程错误 .....	167
3.4.1 穷举算法 .....	108	§6.2 习题解析 .....	168
3.4.2 迭代算法 .....	110	6.2.1 问答题 .....	168
3.4.3 递推算法 .....	115	6.2.2 选择题 .....	169
教学评价 .....	118	6.2.3 填空题 .....	172
<b>第4章 利用函数编程</b> .....	<b>119</b>	§6.3 实训指导 .....	175
§4.1 知识要点 .....	119	6.3.1 实训规划 .....	175
4.1.1 考点内容提要 .....	119	6.3.2 实训内容 .....	175
4.1.2 常见编程错误 .....	121	§6.4 知识拓展 .....	182
§4.2 习题解析 .....	123	6.4.1 贪心算法 .....	182
4.2.1 问答题 .....	123	6.4.2 动态规划 .....	186
4.2.2 选择题 .....	124	教学评价 .....	191
4.2.3 填空题 .....	127	<b>第7章 利用构造类型编程</b> .....	<b>193</b>
§4.3 实训指导 .....	129	§7.1 知识要点 .....	193
4.3.1 实训规划 .....	129	7.1.1 考点内容提要 .....	193
4.3.2 实训内容 .....	129	7.1.2 常见编程错误 .....	196
§4.4 知识拓展 .....	134	§7.2 习题解析 .....	199
4.4.1 递归算法 .....	134	7.2.1 问答题 .....	199
4.4.2 分治算法 .....	136	7.2.2 选择题 .....	201
教学评价 .....	138	7.2.3 填空题 .....	203
<b>第5章 利用数组编程</b> .....	<b>139</b>	§7.3 实训指导 .....	205
§5.1 知识要点 .....	139	7.3.1 实训规划 .....	205

7.3.2 实训内容	205	9.3.1 实训规划	248
§7.4 知识拓展	210	9.3.2 实训内容	248
7.4.1 线性结构	210	§9.4 知识拓展	252
7.4.2 栈结构	212	9.4.1 C语言存在的不足	252
7.4.3 队列结构	213	9.4.2 C++与C语言的关系	253
7.4.4 树结构	215	9.4.3 面向对象程序设计的优点	254
7.4.5 图结构	217	9.4.4 面向对象程序设计的特征	254
教学评价	219	教学评价	256
<b>第8章 文件操作</b>	<b>220</b>	<b>第10章 综合应用程序设计</b>	<b>258</b>
§8.1 知识要点	220	§10.1 知识要点	258
8.1.1 考点内容提要	220	10.1.1 本章学习提示	258
8.1.2 常见编程错误	221	10.1.2 考点内容提要	258
§8.2 习题解析	222	§10.2 软件测试基础	259
8.2.1 问答题	222	10.2.1 软件测试概念	259
8.2.2 选择题	224	10.2.2 黑盒测试和白盒测试	261
8.2.3 填空题	225	§10.3 习题解析	264
§8.3 实训指导	226	10.3.1 问答题	264
8.3.1 实训规划	226	10.3.2 选择题	265
8.3.2 实训内容	226	10.3.3 填空题	267
§8.4 知识拓展	229	§10.4 知识拓展	268
8.4.1 回溯算法	229	10.4.1 数据库的基本概念	268
8.4.2 运算模拟	231	10.4.2 数据管理技术的发展	269
教学评价	235	10.4.3 数据库的数据模型	271
<b>第9章 C++程序设计</b>	<b>237</b>	10.4.4 软件工程的基本概念	276
§9.1 知识要点	237	10.4.5 软件生存周期	278
9.1.1 考点内容提要	237	教学评价	280
9.1.2 常见编程错误	240	<b>附录A Turbo C++ 3.0 环境下编译错误</b>	
§9.2 习题解析	241	信息汇总	282
9.2.1 问答题	241	<b>附录B 全国计算机等级考试二级C语言</b>	
9.2.2 选择题	243	考试大纲	291
9.2.3 填空题	246	<b>参考文献</b>	<b>295</b>
§9.3 实训指导	248		



# 第一篇 预备知识

## 第 1 章 计算机基础知识

**【问题原由】**在各种语言等级考试中都包含了计算机相关基础知识内容。那么，C/C++语言等级考试涉及哪些基础知识呢？这就是本章所要讨论的问题。

**【知识要点】**为了便于读者学习 C/C++，也为了适应参加国家 C/C++等级考试的需要，本章以求解 $f(x) = ax^2 + bx + c$ 为例，讨论利用计算机求解该方程所涉及的相关基本问题，包括计算机中的数据表示、计算机硬件、计算机软件等。

**【能力要求】**掌握计算机中的数据表示方法、计算机硬件的基本概念、计算机软件的基本概念等，为学习 C/C++程序设计奠定基础。

### § 1.1 用计算机求解 $f(x) = ax^2 + bx + c$

利用编程解决实际问题时，通常会涉及两个方面的内容：一是程序设计问题，包括程序设计方法、算法与数据结构、程序设计语言、程序实现（编辑与编译）；另一个是计算机实现问题，包括数据的表示与编码、计算机硬件、计算机软件等。

程序设计是本课程的核心内容，因此在主教材中已有全面介绍，这里仅介绍计算机的实现问题。为了便于初学者理解，我们以 $f(x) = ax^2 + bx + c$ 作为案例，说明计算机必须具备哪些条件才能完成解题任务，以此引出数据的表示和编码、计算机硬件系统和计算机软件系统所涉及的基本概念和问题。

#### 1.1.1 基本任务

设有 $f(x) = ax^2 + bx + c$ ，求 $x_1$ 和 $x_2$ ，可将计算机求解的工作过程描述为4个步骤。

第一步：先将问题求解编成程序，并将原始数据（方程系数和常数项的值） $a$ 、 $b$ 和 $c$ 通过输入设备输入到存储设备中予以保存。

第二步：在计算机控制器的控制作用下，按照程序执行步骤自动地完成如下操作：

① 从存储设备中取出原始数据 $a$ 、 $b$ 和 $c$ ，送到运算部件进行运算，求出中间结果值 $(\sqrt{b^2 - 4ac})/2a$ ，我们将中间结果值用 $g$ 表示。

② 将运算部件的中间结果值 $g$ 送到寄存部件予以临时寄存。

③ 从存储器中取出 $b$ ，从寄存部件取出中间结果值 $g$ ，到运算部件进行运算： $-b + g$ ， $b + g$ 。即 $x_1 = -b + g$ ， $x_2 = b + g$ 。

④ 将运算部件中的最终结果 $x_1$ 和 $x_2$ 送回到存储设备。

第三步：显示或打印存储设备中的最终结果 $x_1$ 和 $x_2$ 。

第四步：停机。

### 1.1.2 涉及的问题

从上述解题过程可知，用计算机解决实际问题涉及以下 4 个方面的内容。

#### 1. 数据的表示

组成现代计算机的电子器件只能识别电位的有、无，通常用 1 和 0 来表示这两种独特状态。因此，计算机中所有的数据信息都只能用由 1 和 0 组成的二进制代码来表示，并且所有的数据信息（数据、符号和文字）都是以二进制代码形式进行存储、处理和传送的。然而，人类通常习惯使用十进制数来描述数据的大小，用文字来描述语言，用符号来描述图形。那么，如何解决“人—机”之间的这种“兼容性”问题呢？实现用计算机解题的第一步就是要解决计算机中数据的表示和转换问题。

#### 2. 数据的编码

由于计算机只能识别 1 和 0 组成的二进制代码，而在编制求解  $f(x) = ax^2 + bx + c$  程序时，无论是程序中提供的系数  $a$ 、 $b$  和  $c$ ，变量“ $x$ ”，运算符“+”和中间结果  $g$ ，还是程序的命令语句，都是字符数据，因此，必须将数值数据和字符数据均使用二进制编码来表示、运算和存储。只有这样，才能使计算机结构简化、运算简单、存储简便、表述简捷。

#### 3. 计算机硬件

要实现用计算机解题的第三步，计算机必须具备以下设备和部件：

- 输入程序和原始数据的输入设备；
- 存放程序和原始数据的存储设备；
- 对数据进行数据处理的运算部件；
- 自动地完成各种操作的控制部件；
- 显示或打印最终结果的输出设备。

我们把构成计算机的所有部件称为硬件（Hardware），并将这些硬件的整体结合称为硬件系统（Hardware System）。

#### 4. 计算机软件

仅有计算机硬件是无法完成给定任务的。因为计算机硬件只能识别电位的高低，没有人—机之间的语言交流工具，用户无法与硬件进行联系，即无法指挥机器做任何事情。要使计算机真正能发挥作用，必须有指挥硬件系统工作的一系列命令，这些命令的有机结合被称为程序。我们把计算机使用的各种程序称为软件（Software），并将所有程序的集合称为软件系统（Software System）。在计算机系统中必须要有如下软件的支持：

- 能实现人—机之间的交流并能对其进行管理的系统程序；
- 能使用户写入原始数据、文件和实现文字处理的编辑程序；
- 把用不同程序语言设计的程序翻译成机器能识别的代码的翻译程序，例如求解  $f(x) = ax^2 + bx + c$ ，必须通过程序设计、编辑、翻译等过程，才能求出  $x_1$  和  $x_2$ 。

计算机软件是为用户操作、管理和维护计算机而编制的各种程序的总和，计算机的启动以及计算机所进行的各项工作都是在软件的支持下完成的。现代计算机，如果没有软件的支持，硬件将变得毫无意义。因此，在计算机系统中软件和硬件具有同样的重要地位。

由此看出，利用计算机实现求解  $f(x) = ax^2 + bx + c$ ，它涉及数据转换、数据的编码、计算

机硬件系统和计算机软件系统。下面，按照上述问题顺序进行简要介绍。

## § 1.2 数制及其转换

### 1.2.1 常用进位计数制

#### 1. 数的位置表示法

人们对各种进位计数制的常用表示方法实际上是一种位置表示法。所谓位置表示法，就是当用一组数码（或字符）表示数值大小时，每一个数码所代表的数值大小不但决定于数码本身，而且还与它在一个数中所处的相对位置有关。例如十进制数 789，其中 9 表示个位上的数，8 表示十位上的数，7 表示百位上的数，这里的个、十、百，在数学上叫做“权”（Weight）或“位权”。如果把把这个十进制数展开，则可表示为：

$$789=7 \times 10^2+8 \times 10^1+9 \times 10^0$$

表达式中 10 的各次幂是各个数位的“位值”，称为各数位的“权”。因此，每个数码所表示的数值大小就等于该数码本身与该位“权”值的乘积；而一个数的值是其各位上的数码乘以该数位的权值之和。相邻两个数位中高位的权与低位的权之比如果是常数，则称为基数（Radix）或底数，通常简称为基（或基码）。如果每一数位都具有相同的基，则称该数制为固定基数制（fixed radix number system），这是计算机内普遍采用的方案。基数是进位计数制中所采用的数码的个数，若用  $r$  来表示，那么它与系数  $a_{n-1}$ ， $a_{n-2}$ ， $\dots$ ， $a_0$ ， $a_{-1}$ ， $\dots$ ， $a_{-(m-1)}$ ， $a_{-m}$  所表示的数值  $N$  为：

$$\begin{aligned} N &= a_{n-1}r^{n-1}+a_{n-2}r^{n-2}+\dots+a_0r^0+a_{-1}r^{-1}+\dots+a_{-(m-1)}r^{-(m-1)}+a_{-m}r^{-m} \\ &= \sum_{i=-m}^{n-1} a_i r^i \end{aligned}$$

式中， $n$  是整数部分的位数， $m$  是小数部分的位数， $n$  和  $m$  均为正整数。从  $a_0r^0$  起向左是数的整数部分，向右是数的小数部分。 $a_i$  表示各数位上的数字，称为系数，它可以在  $0, 1, 2, \dots, r-1$  共  $r$  种数中任意取值。一个  $n$  位  $r$  进制无符号数表示的范围是  $0 \sim r^n-1$ 。

#### 2. 常用进位计数制

在计算机中常用的进位计数制有二进制、八进制、十六进制以及用于输入输出的十进制。由位置表示法可知，根据基数  $r$  的取值不同，便可得到各种不同进位计数制的表达式，并且可分别用不同的下标表示。

（1）当  $r=10$  时，得到十进制（Decimal System）计数的表达式为：

$$(N)_{10} = \sum_{i=-m}^{n-1} a_i 10^i$$

其特点是基为 10，系数只能在  $0 \sim 9$  这十个数字中取值，每个数位上的权是 10 的某次幂。

【实例 1-1】将十进制数 4567.89 按权展开。

解： $(4567.89)_{10}=4 \times 10^3+5 \times 10^2+6 \times 10^1+7 \times 10^0+8 \times 10^{-1}+9 \times 10^{-2}$

（2）当  $r=2$  时，得到二进制（Binary System）计数的表达式为：

$$(N)_2 = \sum_{i=-m}^{n-1} a_i 2^i$$

其特点是基为 2，系数只能在 0 和 1 这两个数字中取值，每个数位上的权是 2 的某次幂。

**【实例 1-2】**将二进制数 $(11011.101)_2$ 按权展开。

解： $(11011.101)_2=1\times 2^4+1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}$

(3) 当  $r=8$  时，得到八进制 (Octal System) 计数的表达式为：

$$(N)_8 = \sum_{i=-m}^{n-1} a_i 8^i$$

其特点是基为 8，系数只能在 0~7 这 8 个数字中取值，每个数位上的权是 8 的某次幂。

**【实例 1-3】**将八进制数 $(4334.56)_8$ 按权展开。

解： $(4334.56)_8=4\times 8^3+3\times 8^2+3\times 8^1+4\times 8^0+5\times 8^{-1}+6\times 8^{-2}$

(4) 当  $r=16$  时，得到十六进制 (Hexadecimal System) 计数的表达式为：

$$(N)_{16} = \sum_{i=-m}^{n-1} a_i 16^i$$

其特点是基为 16，系数只能在 0~15 这 16 个数字中取值。其中 0~9 仍为十进制中的数码，10~15 这 6 个数通常用字符 A、B、C、D、E、F 表示，每个数位上的权是 16 的某次幂。

**【实例 1-4】**将十六进制数 $(23AB.4C)_{16}$ 按权展开。

解： $(23AB.4C)_{16}=2\times 16^3+3\times 16^2+10\times 16^1+11\times 16^0+4\times 16^{-1}+12\times 16^{-2}$

### 3. 常用进位制的比较

对于各种进位计数制，除了使用下标法外，还可以在数的末尾加一个英文字母以示区别。为了便于对照，在表 1-1 中列出了四种不同进位计数制的表示方法。

表 1-1 十进制、二进制、八进制、十六进制数的关系对照表

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	0000B	0Q	0H	8	1000B	10Q	8H
1	0001B	1Q	1H	9	1001B	11Q	9H
2	0010B	2Q	2H	10	1010B	12Q	AH
3	0011B	3Q	3H	11	1011B	13Q	BH
4	0100B	4Q	4H	12	1100B	14Q	CH
5	0101B	5Q	5H	13	1101B	15Q	DH
6	0110B	6Q	6H	14	1110B	16Q	EH
7	0111B	7Q	7H	15	1111B	17Q	FH

其中：B 是 Binary 的缩写，用来表示二进制数；O 是 Octal 的缩写，用来表示八进制数（为了避免与“0”混淆，则写成 Q）；H 是 Hexadecimal 的缩写，用来表示十六进制数；D 是 Decimal 的缩写，用来表示十进制数。十进制数的后缀可省略，但其他进制数的后缀不可省略。例如：

$$\begin{aligned} (331.25)_{10} &= 331.25 = (101001011.01)_2 = (513.2)_8 = (14B.4)_{16} \\ &= 101001011.01B = 513.2Q = 14B.4H \end{aligned}$$

### 1.2.2 数制之间的转换

计算机处理数据时使用的是二进制数，而人们习惯于十进制数，于是就带来了不同数制间的转换问题。常用的数制转换有：二进制、八进制、十六进制数和十进制数之间的相互转换。

## 1. r 进制转换到十进制

一个用 r 进制表示的数，都可用通式  $\sum_{i=m}^{n-1} a_i r^i$  转换为十进制数，通常使用的转换方法是按

“权”相加法。转换时只要把各位数码与它们的权相乘，再把乘积相加，就得到了一个十进制数，这种方法称为按权展开相加法。

**【实例 1-5】** 将  $(100011.1011)_2$  按权展开为十进制数。

解：  $(100011.1011)_2 = 1 \times 2^5 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-4} = (35.6875)_{10}$

**【实例 1-6】** 将  $(37.2)_8$  按权展开为十进制数。

解：  $(37.2)_8 = 3 \times 8^1 + 7 \times 8^0 + 2 \times 8^{-1} = (31.25)_{10}$

**【实例 1-7】** 将  $(AF8.8)_{16}$  按权展开为十进制数。

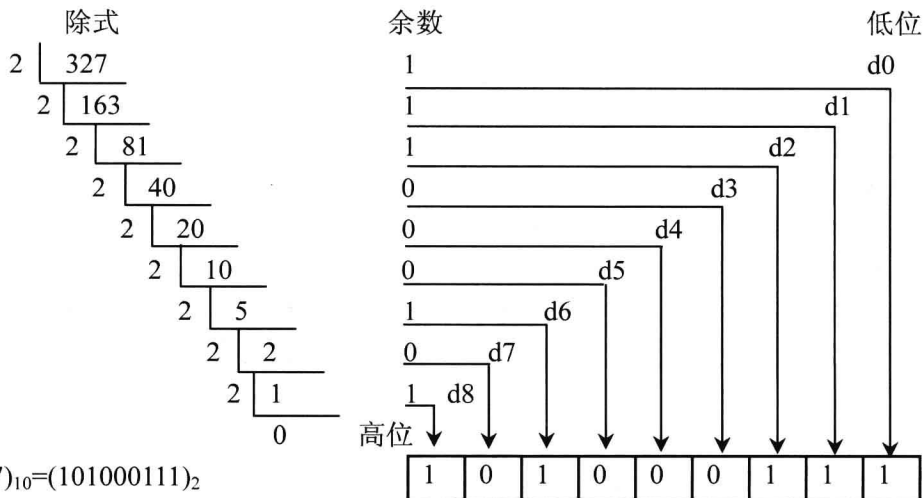
解：  $(AF8.8)_{16} = 10 \times 16^2 + 15 \times 16^1 + 8 \times 16^0 + 8 \times 16^{-1} = (2808.5)_{10}$

## 2. 十进制转换到 r 进制

将十进制转换为 r 进制的常用方法是把十进制整数和小数分别进行处理，称为基数乘法。对于整数部分用除基取余法；对于小数部分用乘基取整法，最后把它们合起来。

(1) 除基取余法：整数部分用基值重复相除的方法，即除基值取余数。设  $r=2$ ，则对被转换的十进制数逐次除 2，每除一次必然得到一个余数 0 或 1，一直除到商为 0。最先的余数是二进制数的低位，最后的余数是二进制的高位。

**【实例 1-8】**  $327 = (?)_2$ ，求解过程如下：



即  $(327)_{10} = (101000111)_2$

用与此类似的方法也可以进行十进制整数  $\rightarrow$  八进制整数的转换或十进制整数  $\rightarrow$  十六进制整数的转换，所不同的只是用 8 或 16 去除。

**【实例 1-9】**  $(1109)_{10} = (?)_8$



$(1109)_{10} = (2125)_8$

(2) 乘基取整法：用基数 2 去乘十进制纯小数，如整数部分为 1，则先得到所求二进制

小数的最高位，然后去掉乘积的整数部分再用 2 去乘余下的纯小数部分，如此继续，直到乘积全部为整数或已满足要求的精度为止，所得各次整数就是所求二进制小数的各位值。

【实例 1-10】 $(0.5625)_{10}=(?)_2$

解：

	整数部分	0.5625
高位	1	$\times 2$ .1250
	0	$\times 2$ .2500
	0	$\times 2$ .5000
低位	1	$\times 2$ .0000

即  $(0.5625)_{10}=(0.1001)_2$

用与此类似的方法也可以进行十进制小数 $\rightarrow$ 八进制小数的转换或十进制小数 $\rightarrow$ 十六进制小数的转换，所不同的只是用 8 或 16 去乘。

【实例 1-11】 $(0.6328125)_{10}=(?)_8$

	整数部分	0.6328125
高位	5	$\times 8$ .0625000
	0	$\times 8$ .5000000
低位	4	$\times 8$ .0000000

即  $(0.6328125)_{10}=(0.504)_8$

由此可以看出，实型数与整型数的转换方法完全相同。如果一个数既有整数部分又有小数部分，则先将这两部分分别按除基取余法和乘基取整法，然后合并，就得到结果。

### 3. 二进制数、八进制数、十六进制数间的转换

由于二进制的权值  $2^i$  和八进制的权值  $8^i=2^{3i}$  及十六进制的权值  $16^i=2^{4i}$  都具有整数指数倍数关系，即一位八进制数相当于三位二进制数；一位十六进制数相当于四位二进制数，故可按如下方法进行转换。

(1) 二进制数转换成八进制数：将二进制数转换成八进制数，则按三位分组，整数不足在高位添 0，凑足三位；小数不足在低位添 0，凑足三位。例如：

01110101 . 10100111B 分组如下：

↑ ↓  
以小数点 . 为分组起点

添 0 凑足三位    001   110   101 . 101   001   110   添 0 凑足三位

↑    ↑    ↑    ↑    ↑    ↑    ↑

对应八进制数：    1    6    5    .    5    1    6

即  $01110101.10100111B=165.516Q$

(2) 二进制数转换成十六进制数：将二进制数转换成十六进制数，则按四位分组，整数不足在高位添 0，凑足四位；小数不足在低位添 0，凑足四位。例如：

01110101 . 10100111B 分组如下：

↑ ↓  
以小数点 . 为分组起点

添 0 凑足四位    0111    0101    .    1010    0111

↑            ↑            ↑            ↑

对应十六进制数： 7            5            .            A            7

即 01110101.10100111B=165.516Q=75.A7H

从常用计数制中我们可以找到一种规律，那就是八进制数和十六进制数都可用二进制数来描述。因为二进制的权值是  $2^i$ ，八进制的权值是  $8^i=2^{3i}$ ，十六进制的权值  $16^i=2^{4i}$ ，它们之间具有整指数倍数关系，即一个八进制数恰好用三个二进制位来描述；一个十六进制数恰好用四个二进制位来描述。也正是因为这样，才能使电路器件得到充分利用。

以上转换，反之也成立。一位八进制数变成三位二进制数；一位十六进制数变成四位二进制数，将其排列起来，即为对应的二进制数。在计算机中，常用数制的转换有二进制、八进制、十进制、十六进制，它们之间的相互转换关系如图 1-1 所示。

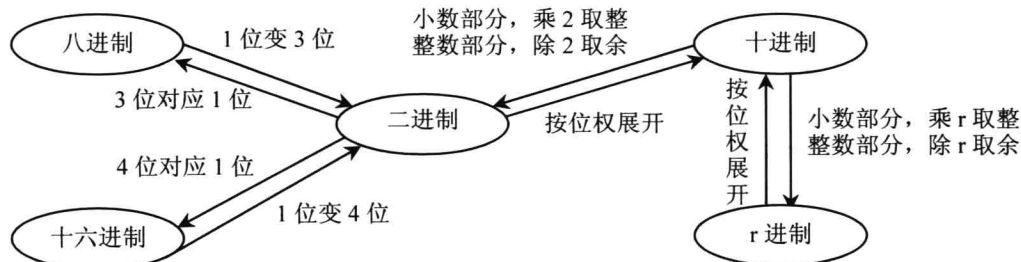


图 1-1 计算机中的数据转换关系

由于十进制数有十个数码（0，1，2，…，9），所以用二进制码表示至少需要四位。四位二进制码能表示十六种状态，其中有六种是冗余状态。如何从十六种状态中选取十种状态，便形成了多种不同的编码方法。用四位二进制码表示一位十进制数的编码方案很多，选择编码方案的原则是既要便于运算，又要便于与二进制转换，并便于校正错误等。

我们通常取的四位二进制数表示一位十进制数，这四位二进制数自左至右，每位的权分别是 8、4、2、1，因而被称为 8-4-2-1 码。这种编码最简单，也最容易理解和记忆，每位的权和二进制数的位是一致的。这里需注意的是：1010、1011、1100、1101、1110、1111 六种数码是不用的。例如十进制数 5016 用 8-4-2-1 码表示可写为：

0101	0000	0001	0110
↑	↑	↑	↑
5	0	1	6

其中每四位表示一位，这和 5016 等值二进制数是不一样的。

### 1.2.3 数值数据的四则运算

#### 1. 二进制数的四则运算

二进制数的算术运算与十进制数的算术运算一样，也包括加、减、乘、除四则运算，但运算规则更加简单，从而使计算机的结构大为简化。在十进制的加、减法运算中，采用“逢十进一”和“借一当十”的运算规则，在二进制数的运算过程中，采用“逢二进一，借一当二”



的运算规则。

(1) 二进制数的加法运算：运算法则是  $0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=0$ 。

【提示】被加数和加数为 1，结果本位为 0，按逢二进一向高位进位 1。

【实例 1-12】将两个二进制数  $(1111)_2$  和  $(1011)_2$  相加。相加过程如下：

$$\begin{array}{r}
 \text{进位} \quad \quad \quad 1 \ 1 \ 1 \ 1 \\
 \text{被加数} \quad \quad \quad 1 \ 1 \ 1 \ 1 \ \cdots \cdots (15)_{10} \\
 \text{加数 } (+) \quad \quad \quad 1 \ 0 \ 1 \ 1 \ \cdots \cdots (11)_{10} \\
 \hline
 \text{和数} \quad \quad \quad 1 \ 1 \ 0 \ 1 \ 0 \ \cdots \cdots (26)_{10}
 \end{array}$$

由该算式可知，两个二进制数相加时，每一位有 3 个数相加：本位的被加数、加数和来自低位的进位（进位为 1 或者 0）。

(2) 二进制数的减法运算：运算法则是： $0-0=0$ ,  $1-0=1$ ,  $0-1=1$ ,  $1-1=0$ 。

【提示】被减数为 0，减数为 1，结果本位为 1，向高位借位。

【实例 1-13】计算二进制数  $(110000)_2 - (10111)_2$ 。相减过程如下：

$$\begin{array}{r}
 \text{借位} \quad \quad \quad 1 \ 1 \ 1 \ 1 \ 1 \\
 \text{被减数} \quad \quad \quad 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ \cdots \cdots (48)_{10} \\
 \text{减数 } (-) \quad \quad \quad 1 \ 0 \ 1 \ 1 \ 1 \ \cdots \cdots (23)_{10} \\
 \hline
 \text{结果} \quad \quad \quad 1 \ 1 \ 0 \ 0 \ 1 \ \cdots \cdots (25)_{10}
 \end{array}$$

由该算式可知，两个二进制数相减时，每一位最多有 3 个数：本位被减数、减数和向高位的借位数。按照减法运算法则可得到本位相减的差数和向高位的借位。

(3) 二进制数的乘法运算：运算法则是  $0 \times 0=0$ ,  $1 \times 0=0$ ,  $0 \times 1=0$ ,  $1 \times 1=1$ 。

【实例 1-14】求两个二进制数的乘积  $(1111)_2 \times (0111)_2$ 。相乘过程如下：

$$\begin{array}{r}
 \text{被乘数} \quad \quad \quad 1 \ 1 \ 1 \ 1 \ \cdots \cdots (15)_{10} \\
 \text{乘数 } (\times) \quad \quad \quad 0 \ 1 \ 1 \ 1 \ \cdots \cdots (7)_{10} \\
 \hline
 \quad \quad \quad \quad \quad \quad 1 \ 1 \ 1 \ 1 \\
 \quad \quad \quad \quad \quad 1 \ 1 \ 1 \ 1 \\
 \quad \quad \quad \quad 1 \ 1 \ 1 \ 1 \\
 +) \quad 0 \ 0 \ 0 \ 0 \\
 \hline
 \text{积} \quad 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ \cdots \cdots (105)_{10}
 \end{array}$$

由该算式可知，两个二进制数相乘时，每一部分乘积都取决于乘数相应位的值。若乘数相应位值为 1，则该次的部分乘积就是被乘数；若为 0，则部分乘积为 0。乘积有几位，就有几个部分乘积。每次的部分乘积左移 1 位，将各部分乘积累加，就得到最后积。

(4) 二进制数的除法运算：运算法则是： $0 \div 0=0$ ,  $0 \div 1=0$ ,  $1 \div 0$ （无意义）， $1 \div 1=1$ 。

【实例 1-15】求二进制数  $(1001110)_2 \div (110)_2$ 。相除过程如下：

$$\begin{array}{r}
 \quad \quad \quad 1 \ 1 \ 0 \ 1 \ \cdots \cdots \text{商}(13)_{10} \\
 1 \ 1 \ 0 \ \sqrt{1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ \cdots \cdots \text{被除数}(78)_{10}} \\
 \quad \underline{1 \ 1 \ 0} \\
 \quad \quad 0 \ 1 \ 1 \ 1 \\
 \quad \quad \underline{1 \ 1 \ 0} \\
 \quad \quad \quad 1 \ 1 \ 0 \\
 \quad \quad \quad \underline{1 \ 1 \ 0} \\
 \quad \quad \quad \quad 0 \ \cdots \cdots \text{余数}(0)_{10}
 \end{array}$$

所以，最后求得结果为： $(1101)_2$ 。

## 2. 八进制数的四则运算

在八进制数的运算过程中,采用“逢八进一,借一当八”的运算规则。八进制数通常只进行加减运算。

【实例 1-16】求  $13450.567+7345.667$ 。

$$\begin{array}{r} 13450.567 \\ + 7345.667 \\ \hline 23016.456 \end{array}$$

【实例 1-17】求  $23016.456-13450.567$ 。

$$\begin{array}{r} 23016.456 \\ - 13450.567 \\ \hline 7345.667 \end{array}$$

## 3. 十六进制数的四则运算

在十六进制数的运算过程中,采用“逢十六进一”,“借一当十六”的运算规则。十六进制数通常只进行加减运算。

【实例 1-18】求  $05C3+3D25$ 。

$$\begin{array}{r} 05C3 \\ + 3D25 \\ \hline 42E8 \end{array}$$

【实例 1-19】求  $3D25-05C3$ 。

$$\begin{array}{r} 3D25 \\ - 05C3 \\ \hline 3762 \end{array}$$

### 1.2.4 二进制数的逻辑运算

计算机不仅能进行算术运算,而且还能进行逻辑运算。逻辑运算的变量称为逻辑变量。逻辑变量只有两个值:“真”(1)或“假”(0)。因此,逻辑运算没有数值大小的概念,只能表达事物内部的逻辑关系,即关系是“成立”还是“不成立”。二进制数的逻辑运算有“与”、“或”、“非”、“异或”4种。

#### 1. 逻辑“与”

逻辑“与”运算产生两个逻辑变量的逻辑积。仅当两个参加“与”运算的逻辑变量都为“1”时,逻辑积才为“1”,否则为“0”。“与”运算用符号“ $\wedge$ ”或“AND”表示。两个逻辑变量的逻辑积的真值表如表 1-2 所示,其对应的逻辑电路如图 1-2 所示。

表 1-2  $F=A \wedge B$  真值表

A	B	$F=A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

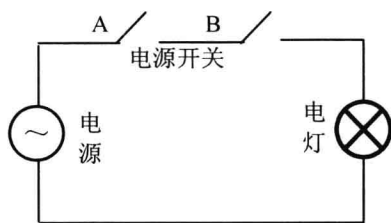


图 1-2 逻辑“与”电路

【实例 1-20】设  $X=10111001$ ,  $Y=11110011$ , 求  $X \wedge Y$ 。

解:

$$\begin{array}{r} 10111001 \\ \text{AND) } 11110011 \\ \hline 10110001 \end{array}$$

即  $X \wedge Y=10110001$

#### 2. 逻辑“或”

逻辑“或”运算产生两个逻辑变量的逻辑和。仅当两个参加“或”运算的逻辑变量都为“0”时,逻辑和才为“0”,否则为“1”。“或”运算用符号“ $\vee$ ”、“+”或“OR”表示。两个