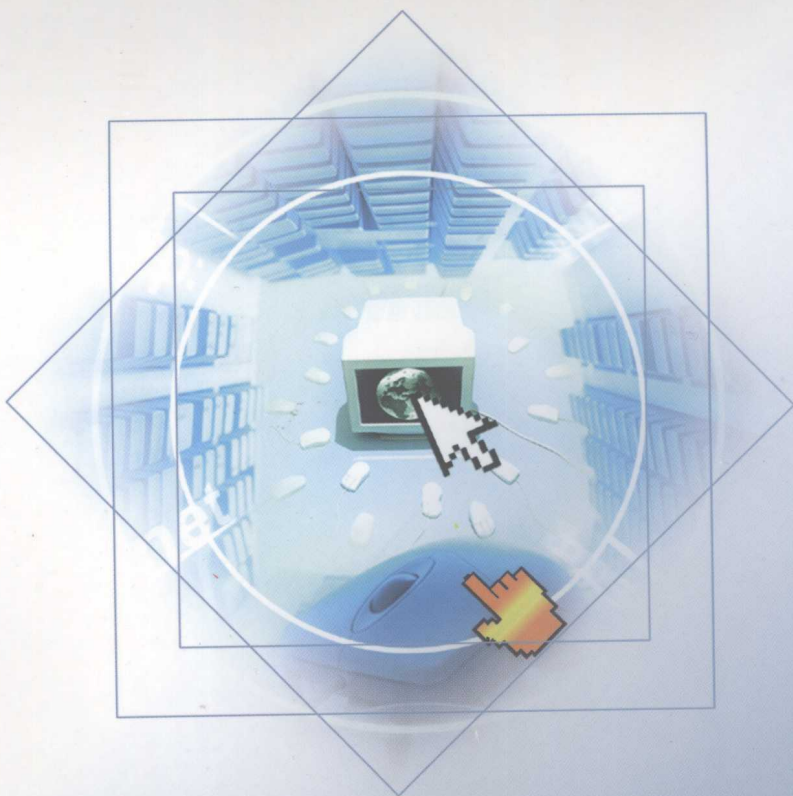


21 世纪高等院校计算机系列教材

Computer

□ 韩海 桂超 主编

C 语言与程序设计



华中科技大学出版社

21 世纪高等院校计算机系列教材

C 语言与程序设计

主 编：韩 海 桂 超

副主编：邓 鸣 李 祥

华中科技大学出版社

图书在版编目(CIP)数据

C语言与程序设计/韩海 桂超 主编
武汉:华中科技大学出版社,2005年2月
ISBN 7-5609-3336-X

- I. C...
- II. ①韩... ②桂...
- III. C语言-程序设计
- IV. TP312

C语言与程序设计

韩海 桂超 主编

责任编辑:彭保林 曾光
责任校对:章红

封面设计:刘卉
责任监印:熊庆玉

出版发行:华中科技大学出版社
武昌喻家山 邮编:430074 电话:(027)87557473

录排:北京搜获科技有限公司
印刷:华中科技大学印刷厂

开本:787mm×1092mm 1/16 印张:18.25 字数:410 000
版次:2005年2月第1版 印次:2006年12月第2次印刷 定价:27.00元
ISBN 7-5609-3336-X/TP·554

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

随着计算机的发展和普及,掌握计算机知识、应用计算机解决实际问题无疑是培养新型人才的一个重要环节。C 语言是目前国内外普遍使用的一种计算机语言,具有广泛的代表性,因此 C 语言往往作为学习程序设计的第一门课程,是进一步学习其他软件编程知识的基础。

本书针对初学者的特点,精心策划,从计算机基础知识出发,由浅入深地介绍 C 语言的语法知识。前两章简要介绍计算机的基本知识,以及 C 语言编程的系统环境和操作过程,作为学习的铺垫;第 3 章至第 7 章讲述 C 语言的基本概念、程序的基本结构和数组的操作;第 8 章介绍程序调试的概念和基本方法,让学习者具备查错、纠错的能力;第 9 章至第 11 章是提高部分,讲述指针、函数、结构体及链表等较深入的内容;第 12 章和第 13 章讲述文件、预处理和位运算的有关知识。

本书内容全面、结构合理、逻辑性强,可以作为高等学校计算机专业的教材;本书语言流畅,内容深入浅出,以日常生活中的实例讲解 C 语言相应的处理方法,也是自学者的理想参考书。

“21 世纪高等院校计算机系列教材”丛书编委会

主任 何炎祥

委员 (按姓氏拼音排序):

戴光明	都志辉	桂 超	金先级
柯敏毅	李康顺	李克清	李禹生
刘腾红	卢强华	陆 迟	吕顺营
沈海波	石 清	王江晴	王伟军
王 忠	叶骏民	余敦辉	湛为芳

序 言

21世纪是信息时代，以计算机为核心的信息技术是21世纪科技发展的大趋势。作为计算机专业人才培养基地的大学计算机专业和相关专业，如何适应这种发展，培养出符合时代要求和社会欢迎的人才，是近年来计算机教育界讨论的热门话题，也是我们长期思考并努力探索的课题。

教材是人才培养的基础。在华中科技大学出版社的委托下，我们组织了有关高等院校的部分专家、教授共同编写了这套“面向21世纪计算机系列教材”，以期在适应21世纪的教材建设方面做出自己的努力。由于计算机行业发展日新月异，“21世纪计算机系列教材”编委会将负责系列教材的选题、每本教材大纲的编写和审定，以及教材、教学辅导书和课件的修订、更新等工作，以确保教材的正确性和先进性，使这套教材努力走在同类教材的前列。

这套系列教材包括计算机专业课和部分专业基础课教材，以及与之配套的实践课教材和教学辅导书等等。

我们希望这套教材具有以下特点：

1. 注重基础性和先进性的结合。计算机学科的一个显著特点就是知识和技术更新快，这对教学内容、课程知识结构的选取和组织提出了新的要求。我们把编写的重点放在基础知识、基本技能和基本方法上，希望在提高学生的理论素养和分析问题、解决问题的能力同时，注重介绍新的技术和方法，以拓展学生的知识面，激发他们学习的积极性和创新意识。

2. 注重理论性与应用性的结合。良好的理论素养是应用的前提，而掌握理论的目的就是为了更好的应用。在教材的编写过程中，我们注意理论的系统性，在讲深讲透主要知识的基础上，融理论性和应用性于一体，注意基本方法的讲授，以培养学生应用理论和技术的能力。

3. 注重时代性和实用性的结合。力求精简旧的知识点，增加新的知识点，体现教材的时代特征。而且充分考虑一般高校目前所拥有的师资条件和教学设备，注重教材的实用性。

4. 注重科学性与通俗性的结合。概念、原理、新技术的阐述力求准确、精练；写作上尽量通俗易懂、深入浅出、图文并茂，增强可读性，便于学生自学。

5. 网络技术辅助教学。针对本系列教材我们开发有专门的网站(<http://www.hzpress.org>)、课件发布演示系统和考试系统等，以便为任课老师的教学提供更便捷、更全面的服 务，并将通过网站开展各种形式的教材网上专家答疑、内容修订发布、课件定期升级等活动，以与读者随时互动，为读者提供立体化的服务。

教学改革是需要不断探索的课题。要达到以上目标，还需要不断地努力实践和完善。欢迎使用这套教材的教师、学生和其他读者提出宝贵意见。

最后，对参加这套教材编写的所有作者，对为这套教材的编写提供支持的有关学校、院系的领导和老师表示诚挚的谢意！感谢华中科技大学出版社为本系列教材的出版所付出的辛勤劳动！

教材编委会主任 **何炎祥**

(教授、博导、武汉大学计算机学院院长)

前 言

自从现代电子计算机在 1946 年诞生以来,计算机在人类的日常生活中扮演着越来越重要的角色,把计算机应用于解决实际生活问题也逐渐成为现代人生活中的一项重要技能。掌握计算机的基本知识和使用技能,学习用计算机语言编写小型应用软件,进而较深入地学习和运用计算机知识是驾驭计算机的几个重要阶段。

计算机语言经历了从低级到高级的多个发展阶段。在众多的计算机语言中,C 语言以其灵活性和实用性受到广大计算机应用人员的喜爱。当今流行的 Windows、Unix、Linux 等各种操作系统都支持以 C 语言进行软件开发。

本书的编者都是长期从事于计算机教学的专业教师。从多年的教学经验中我们体会到 C 语言作为学习计算机程序设计的第一门基础课,不仅要学习 C 语言的基本语法,更重要的是领会过程式程序设计的思想,为进一步学习面向对象的程序设计打好基础。因此我们在编写本书时,力求把 C 语言语法知识和程序设计的思想有机结合在一起,以前者为手段,以后者为目的。在讲述 C 语言的重要语法时,多数都结合日常生活中的实例进行讲解,使读者易于理解和掌握。本书内容深入浅出,结构合理,逻辑性强,即使是初学者,读懂本书亦非难事。

在参考目前已出版的各种 C 语言教材的基础上,我们对各个知识点进行了适当的处理。一方面,弱化了 printf 和 scanf 等输入输出方法中过于繁杂的格式规定,仅保留基本的输入输出手段;对于指向函数的指针等不太常用的章节仅作简单介绍。另一方面,详细描述了函数调用过程(包括递归的实现过程);明确提出了二级指针的概念;加强了关于链表基本操作方法的内容;把一些经典的问题作为例题讲解,所选例题具有较好的代表性和趣味性,如“水仙花数”、“验证哥德巴赫猜想”、“杨辉三角形”、“奇幻方”等。为了配合相关章节的学习,本书每一章都配有丰富的习题,而且习题与知识点联系紧密,以巩固读者的学习效果。

本书由韩海和桂超主编,第 1 章和第 6 章由邓鸣编写,第 2 章和第 3 章由李祥编写,第 4 章、第 5 章、第 12 章和第 13 章由桂超编写,第 7 章至第 11 章由韩海编写。

本书在编写过程中还得到了江汉大学数学和计算机科学学院以及湖北经济学院教师们的大力支持,谨在此表示衷心感谢。

由于编者水平有限,书中难免有错漏之处,敬请广大读者批评指正。

编 者

2005 年 1 月于武汉

目 录

第 1 章 计算机基础知识.....	(1)
1.1 数制及相互转换.....	(1)
1.1.1 数制.....	(1)
1.1.2 十进制数转换成其他进制数.....	(2)
1.1.3 非十进制数转换成十进制数.....	(3)
1.1.4 二进制数与十六进制数相互转换.....	(3)
1.1.5 数的书写方法.....	(4)
1.2 计算机系统.....	(5)
1.2.1 计算机系统的组成.....	(5)
1.2.2 内部存储器.....	(6)
1.3 数据在计算机中的存储.....	(7)
1.3.1 在内存中存储字符.....	(7)
1.3.2 在内存中存储整数.....	(8)
1.3.3 在内存中存储实数.....	(9)
本章要点.....	(9)
习题 1.....	(10)
第 2 章 C 语言概述.....	(11)
2.1 计算机语言.....	(11)
2.2 C 语言的发展与特点.....	(13)
2.3 C 语言的工作环境.....	(14)
2.4 最简单的 C 程序.....	(17)
2.5 流程图.....	(19)
本章要点.....	(20)
习题 2.....	(21)
第 3 章 常量、变量、表达式.....	(22)
3.1 数据类型.....	(22)
3.2 常量.....	(23)
3.2.1 字符常量.....	(23)
3.2.2 整型常量.....	(24)
3.2.3 实型常量.....	(24)
3.2.4 字符串常量.....	(25)
3.3 变量.....	(25)
3.3.1 变量的定义.....	(25)
3.3.2 变量的初始化.....	(26)
3.4 运算符与表达式.....	(27)

3.4.1	算术运算符	(27)
3.4.2	自增和自减运算符	(28)
3.4.3	数据类型的自动转换	(29)
3.4.4	赋值运算符	(29)
3.4.5	复合赋值运算符	(30)
3.4.6	强制类型转换运算符	(30)
3.4.7	圆括号	(31)
3.4.8	字节数运算符 sizeof	(32)
3.5	常用的数学函数	(32)
3.6	广义表达式	(33)
3.6.1	逗号表达式	(33)
3.6.2	条件表达式(问号表达式)	(34)
3.7	表达式应用举例	(34)
	本章要点	(35)
	习题 3	(35)
第 4 章	顺序结构程序设计	(37)
4.1	顺序执行语句	(37)
4.1.1	表达式语句	(38)
4.1.2	函数调用语句	(38)
4.1.3	C 语言程序的一般结构及书写规则	(38)
4.1.4	复合语句	(39)
4.2	标准输入/输出函数	(40)
4.2.1	字符的输入/输出函数 getchar()/putchar()	(40)
4.2.2*	字符串的输入/输出函数 gets()/puts()	(41)
4.2.3	格式化输入/输出函数 scanf()/printf()	(42)
	本章要点	(48)
	习题 4	(48)
第 5 章	分支结构程序设计	(50)
5.1	关系运算符和关系表达式	(50)
5.1.1	关系运算符	(50)
5.1.2	关系表达式	(51)
5.2	逻辑运算符和逻辑表达式	(51)
5.2.1	逻辑运算符	(51)
5.2.2	逻辑表达式	(52)
5.3	if 语句	(53)
5.3.1	简单 if 语句	(53)
5.3.2	标准 if 语句	(54)
5.3.3	复合的 if 语句	(56)
5.4	switch 和 break 语句	(58)

本章要点	(60)
习题 5	(60)
第 6 章 循环结构	(62)
6.1 循环是相似重复	(62)
6.1.1 重复是完成工作的重要方法	(62)
6.1.2 对重复操作进行控制	(63)
6.2 for 语句控制的循环	(64)
6.2.1 for 语句的基本格式	(64)
6.2.2 用 for 语句控制计数	(67)
6.2.3 for 语句的几种特殊用法	(69)
6.3 while 语句控制的循环	(72)
6.3.1 用 while 控制循环	(72)
6.3.2 while 循环与 for 循环的比较	(74)
6.4 do...while 语句控制的循环	(76)
6.5 break 和 continue 语句	(78)
6.5.1 break 语句	(78)
6.5.2 continue 语句	(80)
6.6 循环的典型应用	(81)
6.6.1 有关数列的计算类问题	(81)
6.6.2 其他计算问题	(84)
6.6.3 完全性问题	(85)
6.6.4 存在性问题	(86)
6.7 循环嵌套	(87)
6.7.1 基本概念	(87)
6.7.2 应用举例	(89)
本章要点	(92)
习题 6	(92)
第 7 章 数组	(95)
7.1 数组变量与存储分配	(95)
7.1.1 定义数组变量	(95)
7.1.2 数组变量的存储分配	(97)
7.1.3 数组变量的初值	(98)
7.2 使用数组元素	(99)
7.2.1 赋值和取值	(99)
7.2.2 数组的典型应用	(101)
7.3 二维数组	(105)
7.3.1 二维数组的基本概念	(105)
7.3.2 二维数组的应用	(107)
7.3.3 多维数组	(111)

7.4 字符数组与字符串.....	(112)
7.4.1 基本概念.....	(112)
7.4.2 字符串的相关操作.....	(113)
7.4.3 处理字符串的系统工具.....	(116)
7.4.4 应用示例.....	(119)
本章要点.....	(120)
习题 7.....	(121)
第 8 章 调试程序.....	(125)
8.1 程序出错与处理方法.....	(125)
8.1.1 语法错误.....	(125)
8.1.2 逻辑错误.....	(127)
8.2 在 TC 3.0 环境下调试程序.....	(127)
8.2.1 TC 3.0 集成环境中的调试工具.....	(127)
8.2.2 调试示例.....	(129)
本章要点.....	(133)
习题 8.....	(134)
第 9 章 指针.....	(137)
9.1 指针的基本概念.....	(137)
9.1.1 变量名和变量的地址.....	(137)
9.1.2 存放内存地址的变量.....	(138)
9.1.3 指针与指针变量.....	(139)
9.1.4 定义指针类型标识符.....	(140)
9.2 指针的基本用法.....	(141)
9.2.1 建立指向关系.....	(142)
9.2.2 间接访问变量.....	(142)
9.2.3 用指针作为系统工具的参数.....	(144)
9.3 指向数组元素的指针.....	(145)
9.3.1 指针表达式.....	(146)
9.3.2 与指针有关的运算.....	(149)
9.3.3 用指针变量扫描数组.....	(150)
9.4 指针数组.....	(153)
9.4.1 定义指针数组.....	(153)
9.4.2 指针数组的应用.....	(153)
9.5 二级指针.....	(155)
9.5.1 指向指针变量的指针.....	(155)
9.5.2 指向数组的指针.....	(157)
9.5.3 指向数组的二级指针变量.....	(159)
本章要点.....	(160)
习题 9.....	(161)

第 10 章 函数	(164)
10.1 函数的基本概念	(164)
10.1.1 函数定义	(164)
10.1.2 函数调用	(165)
10.1.3 库函数与头文件	(166)
10.1.4 函数分类	(167)
10.2 自定义函数及其调用	(168)
10.2.1 定义函数的方法	(168)
10.2.2 调用自定义函数	(171)
10.3 函数调用过程	(173)
10.3.1 main 函数	(174)
10.3.2 函数调用规则	(174)
10.3.3 函数调用与返回	(175)
10.3.4 函数的嵌套调用	(178)
10.4 变量的类别	(179)
10.4.1 变量的存储类别	(180)
10.4.2 全局变量和局部变量	(182)
10.5 函数的递归调用	(187)
10.5.1 问题分解与解的合成	(188)
10.5.2 递归函数及调用过程	(189)
10.5.3 汉诺塔问题	(192)
10.6 与函数有关的指针	(195)
10.6.1 指针作为函数的参数	(195)
10.6.2 返回指针的函数	(200)
10.6.3 指向函数的指针	(201)
10.7 模块化编程	(203)
10.7.1 运行多文件的程序	(204)
10.7.2 用 extern 声明外部函数和外部变量	(204)
本章要点	(206)
习题 10	(206)
第 11 章 结构体	(209)
11.1 在内存中存放二维表数据	(209)
11.1.1 把一列数据集中存放——数组	(209)
11.1.2 把一行数据集中存放	(210)
11.2 结构体变量	(211)
11.2.1 定义结构体类型	(211)
11.2.2 定义结构体变量	(212)
11.2.3 结构体变量的初值	(215)
11.2.4 访问结构体变量	(215)

11.2.5 结构体的嵌套.....	(217)
11.3 结构体数组.....	(218)
11.4 指向结构体的指针.....	(219)
11.4.1 指向结构体的指针和指向结构体的域的指针.....	(219)
11.4.2 用结构体或者指向结构体的指针在函数间传递数据.....	(222)
11.5 链表.....	(225)
11.5.1 动态内存分配.....	(225)
11.5.2 结构体中含有指向结构体的指针域.....	(226)
11.5.3 建立链表.....	(227)
11.5.4 插入节点.....	(230)
11.5.5 遍历链表.....	(232)
11.5.6 删除节点.....	(233)
11.5.7 关于链表操作的自定义函数.....	(235)
11.6 复杂的链式结构.....	(239)
11.7 共用体.....	(240)
本章要点.....	(242)
习题 11.....	(243)
第 12 章 文件	(245)
12.1 文件和文件指针.....	(245)
12.1.1 文件的概念.....	(245)
12.1.2 文件操作的基本过程.....	(246)
12.1.3 文件的结构体指针.....	(247)
12.2 文件的打开和关闭.....	(247)
12.2.1 文件打开函数 fopen().....	(247)
12.2.2 文件关闭函数 fclose().....	(249)
12.3 C 语言处理的文件类型.....	(249)
12.4 文件操作.....	(250)
12.4.1 文件记录指针的顺序定位操作.....	(250)
12.4.2 文件记录指针的随机定位操作.....	(253)
本章要点.....	(256)
习题 12.....	(256)
第 13 章 C 语言的预处理命令和位运算	(258)
13.1 C 语言的预处理命令.....	(258)
13.1.1 #define.....	(258)
13.1.2 #include.....	(260)
13.1.3 #error.....	(261)
13.1.4 条件编译预处理命令 #if、#else、#endif.....	(261)
13.1.5 #ifdef.....	(262)
13.1.6 #ifndef.....	(263)

13.1.7 #undef 命令	(263)
13.2 位运算.....	(264)
13.2.1 正负整型数在计算机内的表示方法.....	(264)
13.2.2 位运算符及表达式.....	(264)
本章要点	(266)
习题 13	(266)
附录	(267)
附录 1 字符与 ASCII 对照表	(267)
附录 2 运算符优先级与结合性.....	(268)
附录 3 C 语言的语法规则提要	(268)
附录 4 常用库函数.....	(273)
参考文献.....	(276)

第 1 章 计算机基础知识

从古老的结绳计数到中国人引以自豪的算盘，计算工具伴随着人类文明的发展在不断更新。1946 年第一台电子计算机 ENIAC 的诞生，标志着计算工具的质的飞跃。电子计算机不仅可以高速完成大量计算任务，而且可以进行存储、统计、分类等数据管理工作。

现代电子计算机除了能够存放数据之外，一个重要特征是存储程序，人们可以把一连串的操作命令和数据一起存放到计算机中，然后命令计算机连续完成这些操作步骤，对相应的数据进行处理。人们用计算机能够识别的形式编排各个操作步骤，即称之为计算机程序，无论是程序还是数据，在计算机中都是以二进制形式存储的。

1.1 数制及相互转换

人们在日常生活中每天都要与数打交道，通常书写的数是十进制数，而计算机内部使用的是由 0 和 1 两种符号构成的二进制数。二进制数在书写时过于冗长、容易出错，所以在计算机领域还经常使用八进制数和十六进制数。同一个数值可以用不同的数制书写，不同数制之间可以相互转换。在此将仅说明正整数在不同数制间的转换方法。

1.1.1 数制

描述一个数制需要说明三个方面的问题：有哪些计数符号、每个数位代表的含义和基本的计数规则。

1. 计数符号

书写数据必然使用各种符号，某一数制所有计数符号构成的集合称作该数制的数符集。 k 进制的数符集包含 k 个符号。比如：

二进制数符集当中只有两个符号 0 和 1；

八进制数符集当中有 8 个符号 0, 1, 2, 3, 4, 5, 6 和 7；

十进制数符集当中有 10 个符号 0, 1, 2, 3, 4, 5, 6, 7, 8 和 9；

十六进制数符集当中有 16 个符号 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E 和 F。

各个计数符号是有顺序的，二进制、八进制和十进制的每个计数符号就是它的序号值，十六进制的后 6 个计数符号的序号值依次是 10, 11, 12, 13, 14 和 15。

2. 基数和权

如果用k进制书写一个整数 $a_n a_{n-1} \dots a_1 a_0$, 从右往左把每一位依次记作第0位、第1位……第n位, 则第i位上的数符 a_i 所代表的含义是 $a_i \times k^i$. k称为这个数制的基数, k^i 称为k进制数第i位的权。个位、十位、百位……是十进制各个数位上的权特有的名称, 相应的权是 10^0 、 10^1 、 10^2 ……

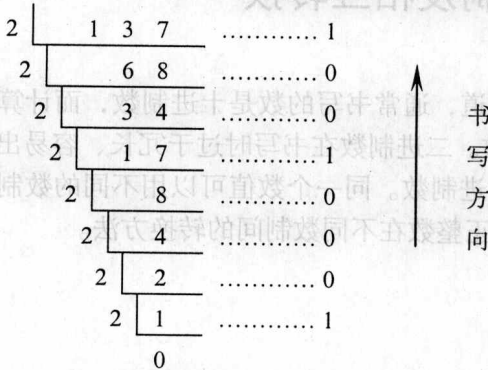
3. 计数规则

简单地说, k进制数的计数规则就是“逢k进1, 借1当k”。

1.1.2 十进制数转换成其他进制数

短除法是十进制正整数转换成二进制数的常用方法。把十进制数连续除以2, 直到商是0为止, 记录每次的余数, 没有余数则记余数为0。以记录余数的相反次序依次写出各个余数, 结果就是相应的二进制数。

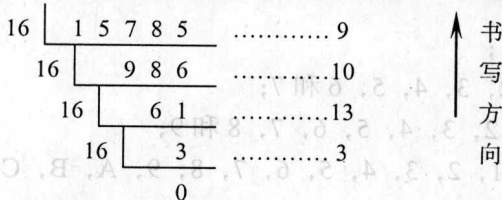
【例 1.1】 把十进制数 137 转换成二进制数。



把每次得到的余数按由下往上的次序写出, 可知与十进制数 137 等值的二进制数是 10001001。

短除法不仅用于十进制到二进制的转换, 也用于十进制数转换成其他进制数。十进制数转换成八进制或者十六进制的方法与转换成二进制的方法相同, 只是除数分别是 8 和 16 而已。在转换成十六进制时还需要把大于 9 的余数写成相应的十六进制数符。

【例 1.2】 把十进制数 15785 转换成十六进制数。



10 对应的十六进制数符是 A, 13 对应 D。所以, 与十进制数 15785 等值的十六进制数是 3DA9。