

21世纪高等学校规划教材 | 计算机应用

程序设计基础教程

（C语言描述）（第二版）

丁海军 金永霞 景雪琴 何坤金 编著



清华大学出版社

013066719

21世纪高等学校规划教材

TP312C
2201

内容简介

本书全面、系统地介绍了C语言程序设计的基本知识、方法和技巧。全书共分10章。第1章介绍C语言的发展概况、特点、应用及基本语法。第2章介绍数据类型、运算符及表达式。第3章介绍控制语句。第4章介绍数组。第5章介绍指针。第6章介绍字符串。第7章介绍结构体。第8章介绍共用体。第9章介绍文件。第10章介绍C语言的高级应用。本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机工作的工程技术人员参考。



程序设计基础教程

(C语言描述)(第二版)

丁海军 金永霞 景雪琴 何坤金 编著



北航

C1674582

清华大学出版社
北京

TP 312C
2201

013066173

内 容 简 介

本书以 C 语言作为程序设计描述语言,介绍了程序设计的概念和一些重要的程序设计算法。全书体系编排新颖,内容丰富,通俗易懂。通过将重要的 C 语言语法集中介绍,读者对 C 语言可以快速入门,在此基础上,可以更好地学习程序设计思想、方法和技巧。

书中介绍了大量的基础算法,例题丰富,具有趣味性。

本书可以作为理工科大学生程序设计基础或者 C 语言课程教材,也可作为大学生程序设计竞赛的基础训练教程。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

程序设计基础教程(C语言描述)/丁海军等编著. —2版. —北京:清华大学出版社,2013
21世纪高等学校规划教材·计算机应用
ISBN 978-7-302-33148-3

I. ①程… II. ①丁… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 159624 号

责任编辑:闫红梅 薛 阳

封面设计:傅瑞学

责任校对:梁 毅

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:21.75

字 数:539千字

版 次:2009年8月第1版 2013年8月第2版

印 次:2013年8月第1次印刷

印 数:1~3000

定 价:35.00元

产品编号:046963-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tup.tsinghua.edu.cn



第二版说明

本书第一版于 2009 年 8 月出版,经过几年使用,作者在征求任课教师和学生意见的基础上,对第一版作出了修订。其一,增加了一章计算机基础知识的内容;其二,对第一版 C 语言程序设计的内容做了部分调整,力求使难点相对分散、更加易于理解;其三,对全书的习题做了调整和精简。

第二版中,对于某些有一定难度的内容,以(*)标注,这些内容在初次讲授时,教师可以酌情跳过,在章节结束后以总结的形式介绍给学生,或者可以直接略过不讲。

在教学中作者发现,很多学生学完整个课程后还不会调试程序,遇到程序出现错误就无从下手,程序调试已经成为学生编程实践中的绊脚石和拦路虎。所以,配合本书,又专门编写了《程序设计实践教程》一书,以实验为重点,介绍程序调试的原理和方法。

作 者

2013 年 5 月

第一版前言

随着信息技术的迅速发展和深入应用,程序设计已经成为高校理工科各专业普遍开设的一门公共基础课。通过学习该课程,学生可以从以下几方面受益:掌握通过编程解决问题的方法和思路,训练逻辑思维能力;掌握正确、规范的编程方法;培养严谨的科学作风;加深对计算机基本概念的理解。

正如学生学习了高等数学但并不需要所有学生都从事数学研究一样,学习程序设计后,并不要求所有学生今后都要从事计算机程序设计工作。作者认为,程序设计作为一门基础课程,其主要作用在于培养、锻炼学生的逻辑思维能力和问题解决能力。程序设计与数学在培养学生逻辑思维能力方面具有不同的作用。数学的主要功能在于培养学生的演绎逻辑思维能力,而程序设计的功能在于培养学生归纳、综合的逻辑思维能力。

由于历史和现实原因,目前国内的大部分C语言教材都是以C语言的语法规则为主线编写,教学重点放在C语言语法规则的学习方面。教学中过分注重C语言语法和一些细节的讲解,教材和课程基本上是以C语言语法体系为脉络展开的。这样,学生虽然花费了很大精力去学习语法,但还是不知道如何编写一个比较简单的程序,不知道怎样通过编程去解决实际问题。

针对这样一些问题,作者在多年教学经验的基础上,决定编写本教材。正如教材名称所蕴含的意义,本教材的重点放在程序设计的思路和方法上面,C语言只是描述程序的工具和手段。本教材具有如下几点特色:

(1) 体系编排和内容组织上,全书强调程序设计过程就是问题求解过程,注重问题求解能力的培养。为了突出“程序设计为主,语法规则为辅”这样一种教学理念,本书将C语言的重要语法规则集中在第2章和第3章介绍,通过这两章的学习,学生可以编写出绝大部分常用程序。这样就为后续程序设计的学习扫清了障碍。

(2) Pascal语言设计者、图灵奖得主Niklaus Wirth教授曾提出一个著名论断“程序=算法+数据结构”。这个论断的本质是说,程序的核心是算法,算法的本质是处理数据,算法与数据不可分离。本教材很好地体现了这一思想,全书大部分内容都是围绕算法和数据组织展开,第4章~第8章都是围绕不同的数据组织方式来编写的。

(3) 在程序设计(算法设计)方面,本书严格按照“自顶向下、逐步求精”这样结构化的程序设计原则进行例题讲解,摒弃了大多数C语言教材中“提出问题,给出源程序,解释程序”的教学思路,而是根据学生的学习规律,采用“提出问题,分析问题,设计算法框架,算法细化,程序实现,程序测试”这样一个步骤来组织内容,这样的组织方式可以更好地培养学生的程序设计能力。

(4) 教材中增加了一定的程序阅读训练方面的内容。程序阅读能力,是进行软件逆向工程的一项基本技能,作者希望通过这样的训练,为学生打下一定的基础。

(5) 教材中涉及的算法尽可能分类组织,从算法设计策略上讲,教材中主要介绍了穷举

策略、递推策略、递归策略、回溯策略。从算法所涉及领域来讲,主要包括数论算法、排序算法、查找算法、字符串算法、大整数算法、数值算法等。

对于初学者来说,程序设计课程具有一定的挑战性,在学习过程中要注意总结学习方法。在教学过程中,作者发现许多数学、物理学习成绩很好的同学,对程序设计课程也是非常头疼和畏惧,究其原因,主要是因为还没有掌握正确的学习方法。那么什么是程序设计课程正确的学习方法呢?

我们知道,以数学为基础的数理学科的知识发展逻辑是演绎逻辑,教材一般是按演绎推理为线索以线性方式组织。对这类学科的学习,一般采用“循序渐进、步步为营”的方式学习,即彻底搞清楚前面的知识单元后,才能很好地学习后面的知识单元。

程序设计课程有不同的特点,主要表现在两个方面:其一,程序设计的基础逻辑思想是归纳逻辑,其内容组织方式呈现网状特征,也就是说,在学习前面的内容时,很可能会涉及后面才会详细介绍的知识点;其二,程序设计是一门实践性很强的课程,任何一个程序设计问题,最终都是需要计算机帮助计算,而这必须反复调试才能得出正确结果。

根据这样一些特点,作者建议读者在学习时注意下面一些学习方法:

(1) 对于该课程,建议采用“整体学习方法”,即在学习前面的内容时,如果遇到不懂的内容,先暂时接受,不要考虑“为什么”,继续学习后面的内容,注意各章节的重点内容。整个内容学习结束后,再回头解决前面的问题。这样反复学习,每一次学习都会有新的收获。

(2) 程序设计是一门实践性很强的课程,学习方法也重在实践,这有两个方面的含义:其一,正如学习数学要有足够量的练习才能融会贯通一样,学习程序设计,也需要足够的上机编程实践才能找到编程的感觉;其二,即使是有经验的程序员,除了极其简单的问题可以一次写出正确的程序外,对于大多数问题,初次写出的程序都会存在或多或少的错误,因此上机调试是学习程序设计过程中必须锻炼的一项基本技能。在程序员中流传着一个“2/8定律”,即对于一个需要编程解决的问题,编写出初步程序所用时间占总时间的20%,而程序调试、排错、修改、正确性测试等要用到总时间的80%。因此不要期望看一两本C语言或程序设计方面的书,就能学好这门课程,这种想法违背了课程特点和学习规律。

(3) 注意学习程序设计的基本思维模式。其一,掌握基本思维过程,从固定思维模式入手。在程序设计中,要掌握数据的“输入→处理→输出”这样一个基本顺序,不能颠倒,这里输入是为处理做准备,处理是为输出做准备。其二,正确理解和应用“自顶向下,逐步求精”的基本原则和思想方法。单个函数不要太大,如果某个函数的内容太多,应该考虑将其中功能相对独立的部分独立出来,编写成函数。

(4) 程序的核心是算法,算法的核心是数据处理,只有经过有效组织的数据才便于算法处理。因此,数据组织方式在程序设计中有重要的作用,在计算机科学中,数据的组织方式称为数据结构。C语言中简单的数组组织方式是基本数据类型的变量,高级数据组织方式主要有数组、字符串、链表、结构体、数据文件等内容。在学习过程中,对不同的数据组织方式,要充分理解它们的存储特点。

(5) 无论是自然语言或者是程序设计语言,语法规则都不难学,困难的是能够用语言正确地表达思想。正如我们为了学好英语,必须掌握一定的“习惯用法”才能很好地使用英语一样,程序设计也有一些固定的“习惯用语”,这就是基本编程模式和简单算法。例如,交换两个变量的值,多项数据求和,多项数据求最大值、最小值,多项数据按指定格式输出等这样

一些程序片段称不上是算法,但是会经常用到,可以称其为编程模式。另外,基本的算法如最大公约数算法、排序算法、查找算法等也是在编写大型程序中会经常用到的算法。对于基本编程模式和基本算法,要求在学习过程中能牢记并能熟练运用。

(6) 教材提供了丰富的例题,在学习过程中,最基本的要求是将所有的例题自己输入计算机,自己调试。很多程序只是在纸上阅读,不一定能看懂,但是在调试过程中通过观察中间结果,可以慢慢理解程序思想和设计方法。对每一个例题或者算法,希望从两个方面注意总结。其一,其中用到的算法思想和算法设计过程与方法;其二,其中涉及的重要语法规则及应用方式。

在本教材编写过程中,参考了 C99 规范,但是目前没有任何一个编译器完全支持 C99 规范,因此只使用了 C99 规范中的“//”注释符,这样处理也便于与 C++ 语言接轨,学生后续学习 C++ 语言会比较自然。

考虑到目前计算机配置普遍以 32 位系统为主体,教材中的例题在 32 位平台上 VC++ 6.0 环境下调试通过。

本书不仅可以用于高等院校理工科学生程序设计课程的教材,也非常适合于 C 语言程序设计爱好者自学,同时,也可以用于程序设计竞赛基础训练教程。

感谢牟艳、周小芹两位老师通读全书,并提出了许多宝贵意见。

感谢北京航空航天大学出版社李文轶编辑提出的宝贵意见,对于她认真负责的敬业精神,作者深表钦佩。

书中使用了作者从网络收集的部分资料,由于过去时间较长,许多资料已经难于找到原创作者。在此,我们对向本书提供营养和素材的网络作者表示衷心感谢。

尽管我们希望做到最好,但因作者水平和成书时间所限,本书仍有许多不足之处,敬请同行教师和读者批评指正。作者联系方式:doceanh@163.com。

编 者

2009 年 7 月

目 录

第 1 章 计算机工作的基本原理	1
1.1 计算的基本概念	1
1.1.1 什么是计算	1
1.1.2 什么是可计算性	1
1.2 数制系统	2
1.2.1 数的进位制	2
1.2.2 不同进制数字转换	4
1.3 整数的二进制原码、反码和补码	7
1.3.1 原码	8
1.3.2 反码	8
1.3.3 补码	8
1.3.4 利用补码作加、减法运算	9
1.4 浮点数的编码	10
1.5 计算机工作原理	10
1.5.1 内存储器	11
1.5.2 运算器	12
1.5.3 控制器	12
1.5.4 指令系统	12
1.5.5 一个简单的模型计算机	14
习题	15
第 2 章 C 语言程序设计基础	16
2.1 C 语言程序概述	16
2.1.1 C 语言程序入门例子	16
2.1.2 C 语言程序组成特点	18
2.2 标识符与关键字	19
2.2.1 标识符命名规则	19
2.2.2 关键字	20
2.3 数据类型	20
2.3.1 基本数据类型	21
2.3.2 数据类型修饰符(*)	22
2.4 常量	23

2.4.1	字面常量	23
2.4.2	符号常量	23
2.5	变量	24
2.5.1	变量定义语句	24
2.5.2	变量定义语句的工作过程(*)	24
2.6	简单输入输出	25
2.6.1	简单输出函数	25
2.6.2	简单输入函数	27
2.7	运算符和表达式	28
2.7.1	赋值运算符	28
2.7.2	算术运算符	30
2.7.3	增量运算符	31
2.7.4	关系运算符	32
2.7.5	逻辑运算符	33
2.7.6	复合赋值运算符	34
2.7.7	表达式的数据类型及类型转换	35
2.7.8	表达式求值过程(*)	36
2.7.9	逗号表达式(*)	37
2.8	格式化输入输出函数高级用法(*)	38
2.8.1	格式化输出	38
2.8.2	格式化输入	40
2.9	关于数据类型的高级问题(*)	41
2.9.1	类型长度运算符	41
2.9.2	类型重定义	42
习题		42
第3章	程序基本结构	45
3.1	程序语句	45
3.1.1	语句概述	45
3.1.2	结构化程序设计	46
3.2	顺序结构	47
3.3	选择结构	48
3.3.1	选择结构的基本语法格式	48
3.3.2	选择结构举例	50
3.3.3	选择结构的嵌套	51
3.3.4	条件运算符	54
3.3.5	多分支选择结构	55
3.3.6	浮点数的相等性比较	60
3.4	循环结构	60

3.4.1	for 循环结构	61
3.4.2	while 循环结构	62
3.4.3	do...while 循环结构	63
3.4.4	循环辅助语句	63
3.4.5	三种基本循环结构的比较	64
3.4.6	循环嵌套	67
3.4.7	goto 语句(*)	69
3.5	函数的定义与使用	69
3.5.1	函数定义	71
3.5.2	函数调用	72
3.5.3	形式参数与实际参数	72
3.5.4	函数原型说明	73
3.5.5	函数应用举例	74
3.6	数据存储特性(*)	77
3.6.1	变量作用域	77
3.6.2	静态变量	79
3.6.3	变量的生存期	80
3.7	预处理指令	82
3.7.1	文件包含指令	82
3.7.2	宏定义指令	82
3.8	综合编程实例	84
	习题	89
第4章	程序设计基本方法	96
4.1	算法与程序设计方法学	96
4.1.1	算法的概念	96
4.1.2	算法设计原则	99
4.1.3	几种基本的算法设计方法和策略	100
4.1.4	编程解决问题的一般步骤	100
4.2	穷举法程序设计	100
4.3	穷举法求解逻辑推理问题	103
4.4	整数相关算法	107
4.4.1	素数	107
4.4.2	最大公约数和最小公倍数	109
4.4.3	歌德巴赫猜想(*)	111
4.4.4	穷举法求解整数相关问题实例	112
4.5	递推与迭代算法 I	116
4.5.1	斐波那契级数	117
4.5.2	整数拆分	118

4.5.3	非线性方程求根	119
	习题	122
第5章	数组	125
5.1	一维数组	125
5.1.1	定义格式	125
5.1.2	一维数组的使用	127
5.2	二维及多维数组	130
5.2.1	定义格式	130
5.2.2	二维数组初始化	130
5.2.3	二维数组的使用	131
5.3	数组与函数	134
5.3.1	数组元素作函数参数	134
5.3.2	一维数组名作为函数参数	135
5.3.3	多维数组作函数参数	140
5.4	程序阅读与理解	142
5.5	排序算法程序设计	145
5.5.1	插入排序	145
5.5.2	选择排序	148
5.5.3	冒泡排序	150
5.6	查找算法程序设计	151
5.6.1	顺序查找算法	151
5.6.2	二分查找算法	152
5.7	集合运算程序设计	154
5.7.1	集合的表示	154
5.7.2	求两个集合的并集算法	154
5.7.3	求两个集合的交集算法	157
5.8	矩阵程序设计	158
5.8.1	矩阵乘法	158
5.8.2	矩阵鞍点	158
5.8.3	螺旋矩阵	161
5.8.4	魔方矩阵	164
5.9	递推与迭代算法Ⅱ	166
5.9.1	沙漠储油点	166
5.9.2	约瑟夫环	168
	习题	170
第6章	指针	178
6.1	指针的概念	178

6.1.1	指针和内存地址	178
6.1.2	指针变量的定义	179
6.1.3	变量地址与间接寻址	180
6.2	指针的运算	182
6.2.1	指针赋值运算	182
6.2.2	指针加、减运算	183
6.2.3	指针比较运算	185
6.3	指针与数组的关系	185
6.3.1	指针与一维数组	185
6.3.2	指针与多维数组(*)	187
6.3.3	指针数组	191
6.4	多级指针	192
6.4.1	多级指针概念	192
6.4.2	多级指针的间接访问	193
6.5	动态存储分配	194
6.5.1	malloc 函数与 free 函数	195
6.5.2	动态内存分配实现可变长一维数组	195
6.5.3	动态内存分配实现可变长二维数组	197
6.6	指针作为函数参数	200
6.6.1	函数传值调用	200
6.6.2	函数传地址调用	201
6.7	函数返回值为指针	205
6.8	指向函数的指针	207
6.8.1	问题的提出	207
6.8.2	指向函数的指针	208
6.9	void 型指针(*)	211
6.9.1	void 指针概念	211
6.9.2	用 void 指针实现通用排序程序	212
	习题	215
第 7 章	字符串	218
7.1	字符数据	218
7.1.1	字符常量	218
7.1.2	转义字符	218
7.1.3	字符型变量	219
7.2	字符串	219
7.2.1	字符串概念及字符串常量	219
7.2.2	字符串的存储	220
7.2.3	字符串的结束标志	220

7.2.4	字符串的表示	220
7.3	字符串数组	224
7.3.1	字符型二维数组表示字符串数组	224
7.3.2	字符型指针数组表示字符串数组	225
7.4	字符串的输入和输出	226
7.4.1	字符串输入	226
7.4.2	字符串输出	227
7.5	处理字符串的基本库函数	228
7.5.1	求字符串长度	228
7.5.2	字符串拷贝	230
7.5.3	字符串比较	230
7.5.4	字符串连接	231
7.5.5	从字符串中查找字符	232
7.5.6	在字符串中查找子串	232
7.5.7	整数与字符串互相转换	233
7.6	字符串基本运算的编程实现	233
7.7	应用举例	236
7.7.1	字符串插入	236
7.7.2	字符串逆置	238
7.7.3	数制转换	239
7.7.4	单词分离	241
7.8	大整数的数学运算($*$)	243
7.8.1	大整数的表示	243
7.8.2	大整数的算术运算	244
7.8.3	大整数应用	249
	习题	250
第8章	文件系统	253
8.1	文件系统的使用方式	254
8.2	文件打开和关闭	255
8.2.1	文件打开	255
8.2.2	文件关闭	256
8.2.3	使用文件的一般模式	256
8.3	文件读写	256
8.3.1	字符读写函数	257
8.3.2	字符串读写函数	258
8.3.3	二进制读写函数	258
8.3.4	格式化读写函数 fscanf 和 fprintf	259
8.4	标准流	261

8.5 文件定位操作	261
8.6 命令行参数	262
习题	264
第 9 章 结构体、链表	266
9.1 结构体	266
9.1.1 问题提出	266
9.1.2 结构体类型声明及变量定义	266
9.2 结构体变量的使用	270
9.2.1 结构体作为整体使用	270
9.2.2 使用结构体成员	270
9.3 结构体与函数	273
9.4 联合体	274
9.4.1 联合体类型声明、变量定义及成员引用	274
9.4.2 联合体变量的特点	275
9.5 枚举类型	276
9.6 链表	277
9.6.1 链表概念	278
9.6.2 链表的基本运算	280
9.6.3 单链表的插入排序	285
9.6.4 链表的逆置(*)	289
9.6.5 其他类型链表(*)	290
习题	291
第 10 章 位处理	293
10.1 整数的计算机表示	293
10.2 位运算符	294
10.2.1 位逻辑运算	294
10.2.2 移位运算	296
10.2.3 位运算赋值运算符	296
10.3 位域	297
10.3.1 位域的定义	298
10.3.2 位域的使用	299
10.4 位运算的应用(*)	299
10.4.1 利用位运算实现整数值交换	299
10.4.2 二进制 GCD 算法	300
10.4.3 均匀二分查找	301
习题	302

第 11 章 函数递归调用及应用	304
11.1 函数递归调用	304
11.1.1 函数调用原理	304
11.1.2 递归调用	305
11.2 函数递归调用的应用	309
11.2.1 算法设计的分治法策略(汉诺塔问题)	309
11.2.2 算法设计的回溯法策略(八皇后问题)	311
习题	315
附录 A ASCII 码表	316
附录 B C 语言保留字(关键字)	318
附录 C C 语言运算符优先级	320
附录 D C 语言主要标准库函数	322
参考文献	327