



普通高等教育“十一五”国家级规划教材

中国高等院校计算机基础教育课程体系规划教材

C 程序设计教程

谭浩强 著

- 《C程序设计》（已发行900万册）的姊妹篇
- 适应“少学时”的课程需要
- 使C语言更容易学习
- 突出重点，紧扣最基本的教学要求
- 采用Visual C++作为编译环境



清华大学出版社



普通高等教育“十一五”国家级规划教材

中国高等院校计算机基础教育课程体系规划教材

C 程序设计教程

谭浩强 著

- 《C程序设计》（已发行900万册）的姊妹篇
- 适应“少学时”的课程需要
- 使C语言更容易学习
- 突出重点，紧扣最基本的教学要求
- 采用Visual C++作为编译环境

清华大学出版社
北京

内 容 简 介

C 语言是国内外广泛使用的计算机语言,学会使用 C 语言进行程序设计是计算机工作者的一项基本功。

本书作者具有丰富的教学经验和编写教材的经验,善于把复杂的问题简单化,用读者易于理解的方法和通俗易懂的语言阐述清楚复杂的概念,化解了学生学习过程中的许多困难。

本书是作者针对全国大多数高校近年来在计算机教育中的实际情况,在作者所著的《C 程序设计》(清华大学出版社)一书的基础上重新组织和改写而成,在编写中注意了压缩篇幅,突出重点,紧扣基本要求,使广大读者更加容易学习。

本书针对初学者的特点,精心策划,准确定位,内容新颖,概念清晰,体系合理,例题丰富,深入浅出,文字流畅,通俗易懂,是学习 C 语言的理想教材。

凡具有计算机初步知识的读者都能读懂本书。本书可作为高等学校各专业的教材,也是一本供自学的好教材。本书还配有辅助教材《C 程序设计教程学习辅导》。

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 13501256678 13801310933

图书在版编目(CIP)数据

C 程序设计教程/谭浩强著. —北京:清华大学出版社,2007.7

(中国高等院校计算机基础教育课程体系规划教材)

ISBN 978-7-302-15157-9

I. C… II. 谭… 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(2007)第 063676 号

责任编辑:张 民

责任校对:白 蕾

责任印制:何 芊

出版发行:清华大学出版社

<http://www.tup.com.cn>

c-service@tup.tsinghua.edu.cn

社总机:010-62770175

投稿咨询:010-62772015

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮购热线:010-62786544

客户服务:010-62776969

印 刷 者:清华大学印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:21.75

字 数:504 千字

版 次:2007 年 7 月第 1 版

印 次:2007 年 7 月第 1 次印刷

印 数:1~10000

定 价:25.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:025832-01

中国高等院校计算机基础教育课程体系规划教材

编审委员会

主 任：谭浩强

委 员：(按姓氏笔画为序)

王路江 冯博琴 刘瑞挺 吴文虎 吴功宜

张 森 高 林 龚沛曾 焦金生 焦 虹

策划编辑：张 民

从20世纪70年代末、80年代初开始,我国的高等院校开始面向各个专业的全体大学生开展计算机教育。特别是面向非计算机专业学生的计算机基础教育,牵涉的专业面广、人数众多,影响深远。高校开展计算机基础教育的状况将直接影响我国各行各业、各个领域计算机应用的发展水平。这是一项意义重大而且大有可为的工作,应该引起各方面的充分重视。

20多年来,全国高等院校计算机基础教育研究会和全国高校从事计算机基础教育的老师始终不渝地在这片未被开垦的土地上辛勤工作,深入探索,努力开拓,积累了丰富的经验,初步形成了一套行之有效的课程体系和教学理念。20年来高等院校计算机基础教育的发展经历了3个阶段:20世纪80年代是初创阶段,带有扫盲的性质,多数学校只开设一门入门课程;20世纪90年代是规范阶段,在全国范围内形成了按3个层次进行教学的课程体系,教学的广度和深度都有所发展;进入21世纪,开始了深化提高的第3阶段,需要在原有基础上再上一个新台阶。

在计算机基础教育的新阶段,要充分认识到计算机基础教育面临的挑战:

(1) 在世界范围内信息技术以空前的速度迅猛发展,新的技术和新的方法层出不穷,要求高等院校计算机基础教育必须跟上信息技术发展的潮流,大力更新教学内容,用信息技术的新成就武装当今的大学生。

(2) 我国国民经济现在处于持续快速稳定发展阶段,需要大力发展信息产业,加快经济与社会信息化的进程,这就迫切需要大批既熟悉本领域业务,又能熟练使用计算机,并能将信息技术应用于本领域的新型专门人才。因此需要大力提高高校计算机基础教育的水平,培养出数以百万计的计算机应用人才。

(3) 从21世纪初开始,信息技术教育在我国中小学中全面开展,计算机教育的起点从大学下移到中小学。水涨船高,这样也为提高大学的计算机教育水平创造了十分有利的条件。

迎接21世纪的挑战,大力提高我国高等学校计算机基础教育的水平,培养出符合信息时代要求的人才,已成为广大计算机教育工作者的神圣使命和光荣职责。全国高等院校计算机基础教育研究会和清华大学出版社于2002年联合成立了“中国高等院校计算机基础教育改革课题研究组”,集中了一批长期在高校计算机基础教育领域从事教学和研究的专家、教授,经过深入调查研究,广泛征求意见,反复讨论修改,提出

了高校计算机基础教育改革思路和课程方案，并于2004年7月公布了《中国高等院校计算机基础教育课程体系2004》（简称CFC 2004）。CFC 2004公布后，在全国高校中引起强烈的反响，国内知名专家和从事计算机基础教育工作的广大教师一致认为CFC 2004提出了一个既体现先进又切合实际的思路和解决方案，该研究成果具有开创性、针对性、前瞻性和可操作性，对发展我国高等院校的计算机基础教育具有重要的指导作用。根据近年来计算机基础教育的发展，课题研究组对CFC 2004进行了修订和补充，使之更加完善，于2006年7月公布了《中国高等院校计算机基础教育课程体系2006》（简称CFC 2006），由清华大学出版社出版。

为了实现课题研究组提出的要求，必须有一批与之配套的教材。教材是实现教育思想和教学要求的重要保证，是教学改革中的一项重要的基本建设。如果没有好的教材，提高教学质量只是一句空话。要写好一本教材是不容易的，不仅需要掌握有关的科学技术知识，而且要熟悉自己工作的对象、研究读者的认识规律、善于组织教材内容、具有较好的文字功底，还需要学习一点教育学和心理学的知识等。一本好的计算机基础教材应当具备以下5个要素：

(1) 定位准确。要十分明确本教材是为哪一部分读者写的，要有的放矢，不要不问对象，提笔就写。

(2) 内容先进。要能反映计算机科学技术的新成果、新趋势。

(3) 取舍合理。要做到“该有的有，不该有的没有”，不要包罗万象、贪多求全，不应把教材写成手册。

(4) 体系得当。要针对非计算机专业学生的特点，精心设计教材体系，不仅使教材体现科学性和先进性，还要注意循序渐进、降低台阶、分散难点，使学生易于理解。

(5) 风格鲜明。要用通俗易懂的方法和语言叙述复杂的概念。善于运用形象思维，深入浅出，引人入胜。

为了推动各高校的教学，我们愿意与全国各地、各学校的专家和老师共同奋斗，编写和出版一批具有中国特色的、符合非计算机专业学生特点的、受广大读者欢迎的优秀教材。为此，我们成立了“中国高等院校计算机基础教育课程体系规划教材”编审委员会，全面指导本套教材的编写工作。

这套教材具有以下几个特点：

(1) 全面体现CFC 2004和CFC 2006的思路和课程要求。本套教材的作者多数是课题研究组的成员或参加过课题研讨的专家，对计算机基础教育改革的方向和思路有深切的体会和清醒的认识。因而可以说，本套教材是CFC 2004和CFC 2006的具体化。

(2) 教材内容体现了信息技术发展的趋势。由于信息技术发展迅速，教材需要不断更新内容，推陈出新。本套教材力求反映信息技术领域中新的发展、新的应用。

(3) 按照非计算机专业学生的特点构建课程内容和教材体系，强调面向应用，注重培养应用能力，针对多数学生的认知规律，尽量采用通俗易懂的方法说明复杂的概念，使学生易于学习。

(4) 考虑到教学对象不同, 本套教材包括了各方面所需要的教材(重点课程和一般课程; 必修课和选修课; 理论课和实践课), 供不同学校、不同专业的学生选用。

(5) 本套教材的作者都有较高的学术造诣, 有丰富的计算机基础教育的经验, 在教材中体现了研究会所倡导的思路和风格, 因而符合教学实践, 便于采用。

本套教材统一规划、分批组织、陆续出版。希望能得到各位专家、老师和读者的指正, 我们将根据计算机技术的发展和广大师生的宝贵意见随时修订, 使之不断完善。

全国高等院校计算机基础教育研究会会长
“中国高等院校计算机基础教育课程体系规划教材”编审委员会主任

谭浩强

C 语言是在国内外广泛使用的一种计算机语言。学会使用 C 语言进行程序设计是计算机工作者的一项基本功。

自 20 世纪 90 年代以来, C 语言在国内迅速普及, 不仅为计算机专业工作者所使用, 而且为广大计算机应用人员(包括大量原来是非计算机专业的使用计算机的人们)所喜爱和使用。几乎所有的理工科大学不仅在计算机系开设了 C 语言课程, 而且在非计算机专业也开设了 C 语言课程。全国计算机等级考试、全国计算机应用技术证书考试(NIT)和全国各地区各部门组织的计算机考试都将 C 语言列入了考试范围。学习 C 语言已经成为广大计算机应用人员和广大青年学生的迫切要求。

作者在 1991 年编著了《C 程序设计》一书, 由清华大学出版社出版。该书出版后受到广大读者的热烈欢迎; 许多读者说“C 语言原来是比较难学的, 但自从《C 程序设计》出版后, C 语言变得不难学了”, 该书被全国大多数高校选为正式教材, 许多高校都指定该书为研究生入学考试必读教材。该书已成为国内读者学习 C 语言的主流用书。国内许多介绍 C 语言的书籍大多以此书为蓝本, 许多在职干部和计算机爱好者通过自学本书掌握了 C 语言程序设计。十多年来, 该书已出了三版, 重印了 100 多次, 累计发行了 900 多万册, 创同类书的全国发行最高纪录。近年来每年印刷量都在 80 万册以上。作者到全国各高校和各企事业单位访问时, 许多在校师生和已毕业参加了工作的人士都说他们学过这本书, 印象很深, 作者在内心深切地感受到广大读者对作者的殷切期望。

各校师生普遍认为《C 程序设计》内容系统、讲解详尽, 包含了许多其他教材中没有的内容, 尤其是针对编程实践中容易出现的问题作了提醒和分析, 是学习 C 语言程序设计的理想教材。同时有的学校提出, 由于各校情况不完全相同(例如, 学校的类型不同, 教学要求不同, 安排的学时数不同, 学生的基础不同……), 希望在保持原有的优点的基础上, 能提供适用于不同要求的教材。作者和清华大学出版社征求了多方面的意见, 进行了反复的研究, 决定除了继续完善《C 程序设计》以外, 针对学时较少的学校, 另外出版一本新的《C 程序设计教程》, 该书以《C 程序设计》一书的内容为基础, 适当减少内容, 压缩篇幅, 突出重点, 紧扣最基本的要求。经教育部批准, 《C 程序设计教程》已正式列入教育部普通高等教育“十一五”国家级教材规划, 向全国各高校推荐。

和《C 程序设计》相比, 本书在以下几方面作了调整:

(1) 取消原来的第 2 章(算法), 把算法的思想分散到各章, 结合具体程序进行介绍, 这样可以使读者更早地直接进入程序设计, 能提高读者的兴趣。

(2) 对于初学者暂时用得不多的部分内容(如“预处理命令”、“位运算”)不再保留。对“链表”只作简单的介绍而不深入展开。对输入输出格式着重介绍最常用的部分, 不详细介绍其他复杂和不常用的格式。对不同类型数据之间的赋值只介绍比较基本的内容。“常见错误和程序调试”的内容由主教材移放到《C 程序设计教程学习辅导》中。这样教材由原来的 14 章减少到 10 章, 篇幅由 378 页减少到 323 页。

(3) 由于许多学校把 C 语言的教学安排在一年级, 而学生还未学完高等数学, 在本书中不包括有关高等数学知识的例题。

(4) 过去学习 C 语言的读者, 大多数是学习过其他计算机语言的(如 BASIC 语言), 有初步的程序设计知识, 而现在大多数学校把 C 语言作为第一门计算机语言来学习, 学生缺乏程序设计的基础知识, 为减少学习难度, 减少了难度较大的例题, 对比较难懂的程序也作了比较详细的说明。

(5) 在章节编排上, 作了适当的改变, 在叙述概念和解释难点时, 采取了更加通俗易懂的方法, 使读者更易于理解。

(6) 采用 Visual C++ 作为编译环境, 使程序的编译和运行更加方便。

由于《C 程序设计》的内容和写作风格已经为广大读者熟悉, 在教学的实践中取得了很好的效果, 因此本书保持了《C 程序设计》的基本内容和写作风格, 可以认为本书是《C 程序设计》的简明教程。在学习本书的过程中, 如需要进一步深入, 可以参考《C 程序设计》一书。

对于要求较高的学校, 建议仍选用《C 程序设计》作为教材, 可以学得深入一些。

对于怎样学习程序设计, 作者提出以下几点看法:

(1) 近年来, 有一些面向对象的计算机语言陆续问世, 受到欢迎。有些人认为面向过程的 C 语言已经过时了, 不必学了。这是一种误解。不应把面向对象和面向过程对立起来, 在面向对象程序设计中仍然要用到面向过程的知识。作为计算机程序开发人员, 既要掌握面向对象程序设计的知识, 又要掌握面向过程程序设计的知识。面向过程程序设计仍然是计算机工作者的基本功。

C++ 是为开发大型程序而研制的, 它比 C 语言要复杂得多, 学习 C++ 也比学习 C 语言困难得多。事实上, 将来并不是每个人都需要用 C++ 编制大型程序。对于计算机专业人员, 学习和使用 C++ 是理所当然的。对于非计算机专业人员, 可以先学习 C 语言, 打下程序设计的基础, 在以后需要时再学习和使用 C++。有了 C 语言基础, 再学习 C++ 就容易多了。也可以在学习 C 语言的基础上, 再学习一些面向对象程序设计的初步知识, 为以后进一步学习和使用 C++ 打下基础。

(2) 在学校中, 学习程序设计课程的目的是掌握设计程序的思路, 学会用计算机语言编写程序, 以实现所需处理的任务。要正确处理算法与语法的关系, 算法是程序的核心、是灵魂, 语法是外壳、是工具。本书不是一本介绍语法的书, 而是帮助读者通过大量实例了解怎样设计和运用算法, 怎样运用语言工具和编程环境去实现编程, 从而处理实际问题。不应把学习重点放在语法规则上, 语法是重要的, 不掌握语法规则就无法编写出正确的程序, 但是只学会语法, 甚至能把语法背得滚瓜烂熟, 也不可能

编写出好的程序。一定要把重点放在解题的思路，通过大量的例题学习怎样设计一个算法，构造一个程序。在学习初期更不要在语法细节上死背死抠。请记住：重要的是学会编程序，而不是背语法。一开始就要学会看懂程序，编写简单的程序，然后逐步深入。有一些语法细节是需要通过较长期的实践才能熟练掌握的。初学时，切忌过早地滥用C语言的某些容易引起错误的细节（如不适当地使用++和--的副作用）。

(3) 不能设想今后一辈子只使用在学校里学过的某一种语言。但是，无论用哪一种语言进行程序设计，其基本规律是一样的。在学习时一定要活学活用，举一反三，掌握规律，在以后需要时能很快地掌握其他新的语言进行工作。

(4) 在学校学习阶段，主要是学习程序设计的方法，进行程序设计的基本训练，为将来进一步学习打下基础。对多数学生来说，不可能通过几十小时的学习，由一个门外汉变成编程高手，编写出大型而实用的程序。学习程序设计课程时，应该把精力放在最基本、最常用的内容上，学好基本功。如果对学生有较高的程序设计要求，应当在学习本课程后，安排一次集中的课程设计环节，按照实际工作的要求，完成有一定规模的程序设计。

(5) 程序设计是一门实践性很强的课程，既要掌握概念，又要动手编程，还要上机调试运行，希望读者一定要重视实践环节，包括编程和上机。既会编写程序，又会调试程序。衡量这门课学习的好坏，不是看你“知不知道”，而是“会不会干”。考核的方法不能主要用是非题和选择题，而应当把重点放在编制程序和调试程序上。

(6) 使用哪一种编译系统并不是原则问题，重要的是编程能力的培养。程序编好以后，用哪一种编译系统进行编译都可以。读者不应该只会用某一种编译环境，应当了解、接触和使用不同的编译环境。不同的编译系统，其功能和使用方法有些不同，编译时给出的信息也不完全相同，要注意参阅使用说明书，特别要在使用中积累经验，举一反三。

(7) 本书的写法既通俗又精练，许多叙述是在大量实践经验基础上提炼和总结出来的。学习时一定要对照程序反复琢磨和思考，深入掌握有关概念及其应用的方法，千万不要浅尝辄止，一带而过。有些重要的叙述最好反复阅读两遍甚至多遍。

为了帮助读者学习本书，作者还编了一本《C程序设计教程学习辅导》，提供本书中各章习题的参考答案以及上机实习指导。该书由清华大学出版社于2007年出版。

作者从事计算机教育和计算机普及工作近三十年，对这项工作有深厚的感情和深切的体会。在许多年前我就下了决心，把自己的后半生献给我国的计算机教育和计算机普及事业，要安于平凡，脚踏实地，把看似平凡的工作做成伟大。我最大的愿望是：通过全社会的共同努力，使全体人民都能享受到信息技术发展的成果。我提出了一个口号：“要把计算机从少数计算机专家手中解放出来，使它成为广大群众手中的工具”，要使广大群众轻松愉快、兴趣盎然地进入计算机的天地。经过近二十年的努力，这个愿望正在逐步变成现实。

我始终认为，作为作者，不能想怎么写就怎么写，作为教师，不能想怎么讲就怎么讲，必须考虑怎样才能使学生易于理解。作者心中要永远装着读者，处处为读者着想。要写出一本好书是不容易的，要讲好一堂课也是不容易的，要下很大的功夫。

要深入了解自己工作的对象，有的放矢，准确定位；要根据应用的需要，合理取舍，精选内容；要认真研究学习者的认识规律，采用读者容易理解的方法，深入浅出，通俗易懂；要善于把复杂问题简单化，而不能把简单问题复杂化。写书不是简单地把有关的技术内容告诉读者，而是要考虑怎样写才能使读者容易理解。有时为了找到一个好的例子或一个通俗的比喻，我往往苦苦思索好几天，每一句话都要反复斟酌推敲。希望所有的教师和作者共同努力，把每一本书、每一门课程都做成精品，得到千万学生和读者的肯定和赞扬，这才是对我们劳动的最高奖赏。

由于作者水平有限，本书肯定会有不少缺点和不足，热切期望得到专家和读者的批评指正。

谭浩强

2007年5月于清华园

第 1 章 C 语言概述	1
1.1 计算机与程序、程序设计语言.....	1
1.2 C 语言出现的历史背景	2
1.3 C 语言的特点	3
1.4 简单的 C 语言程序	4
1.5 运行 C 程序的步骤与方法	8
1.5.1 运行 C 程序的步骤	8
1.5.2 上机运行 C 程序的方法	8
本章小结	15
习题	16
第 2 章 数据类型、运算符与表达式	17
2.1 C 语言的数据类型.....	17
2.2 常量与变量.....	17
2.2.1 常量	18
2.2.2 变量	19
2.3 整型数据.....	20
2.3.1 整型常量的表示形式	20
2.3.2 整型变量	21
2.3.3 整型数据在内存中的存放形式	22
2.3.4 整型数据的溢出	23
2.3.5 整型常量的类型	24
2.4 浮点型数据.....	24
2.4.1 浮点型常量的表示形式	24
2.4.2 浮点型变量	25
2.4.3 浮点型常量的类型	26
2.5 字符型数据.....	27
2.5.1 字符常量	27

2.5.2	字符变量	28
2.5.3	字符数据在内存中的存储形式及其使用方法	28
2.5.4	字符串常量	31
2.6	各类数值型数据间的混合运算	32
2.7	算术运算符和算术表达式	33
2.7.1	C 语言运算符和 C 算术表达式简介	33
2.7.2	主要的算术运算符和算术表达式	34
2.8	逗号运算符和逗号表达式	36
	本章小结	38
	习题	38
第 3 章	最简单的 C 程序设计——顺序程序设计	40
3.1	程序的三种基本结构	40
3.2	C 语句概述	41
3.3	赋值表达式和赋值语句	43
3.3.1	赋值表达式	43
3.3.2	赋值过程中的类型转换	46
3.3.3	赋值语句	48
3.3.4	变量赋初值	48
3.4	数据输入输出的概念及在 C 语言中的实现	49
3.5	字符数据的输入输出	50
3.5.1	用 putchar 函数输出一个字符	50
3.5.2	用 getchar 函数输入一个字符	51
3.6	简单的格式输入与输出	52
3.6.1	用简单的 printf 函数输出数据	52
3.6.2	用简单的 scanf 函数输入数据	55
3.7	较复杂的输入输出格式控制	58
3.7.1	输出数据格式控制	58
3.7.2	输入数据格式控制	63
3.8	顺序结构程序设计举例	64
	本章小结	66
	习题	67
第 4 章	选择结构程序设计	69
4.1	关于算法	69
4.1.1	算法的概念	69
4.1.2	怎样表示算法	71
4.2	条件判断	73
4.2.1	关系运算和关系表达式	73

4.2.2	逻辑运算符和逻辑表达式	75
4.3	用 if 语句实现选择结构	78
4.3.1	if 语句的 3 种形式	78
4.3.2	if 语句的嵌套	81
4.3.3	条件运算符和条件表达式	84
4.4	利用 switch 语句实现多分支选择结构	87
4.5	程序举例	89
	本章小结	94
	习题	94
第 5 章	循环结构程序设计	96
5.1	循环的概念	96
5.2	用 while 语句和 do...while 语句实现循环	96
5.2.1	用 while 语句实现循环	96
5.2.2	用 do...while 语句实现循环	98
5.2.3	while 和 do...while 循环的比较	99
5.2.4	while 循环程序举例	100
5.3	用 for 语句实现循环	101
5.3.1	for 语句的一般形式和执行过程	101
5.3.2	for 语句的各种形式	102
5.3.3	for 循环程序举例	105
5.4	循环的嵌套	106
5.5	用 break 语句和 continue 语句改变循环状态	107
5.5.1	用 break 语句提前退出循环	107
5.5.2	用 continue 语句提前结束本次循环	108
5.6	几种循环的比较	109
5.7	程序举例	109
	本章小结	112
	习题	113
第 6 章	数组	115
6.1	数组的概念	115
6.2	一维数组的定义和引用	115
6.2.1	定义一维数组	115
6.2.2	引用一维数组元素	116
6.2.3	一维数组的初始化	117
6.2.4	一维数组程序举例	117
6.3	二维数组的定义和引用	120
6.3.1	定义二维数组	120
6.3.2	引用二维数组的元素	120

6.3.3	二维数组的初始化	121
6.3.4	二维数组程序举例	122
6.4	字符数组	124
6.4.1	定义字符数组	124
6.4.2	字符数组的初始化	125
6.4.3	引用字符数组的元素	125
6.4.4	字符串和字符串结束标志	126
6.4.5	字符数组的输入输出	128
6.4.6	字符串处理函数	130
6.4.7	字符数组应用举例	134
	本章小结	137
	习题	138
第7章	函数调用	140
7.1	函数概述	140
7.2	函数定义的一般形式	142
7.2.1	为什么要定义函数	142
7.2.2	怎样定义无参函数	142
7.2.3	怎样定义有参函数	143
7.2.4	定义空函数	143
7.3	函数参数和函数的值	144
7.3.1	形式参数和实际参数	144
7.3.2	函数的返回值	145
7.4	函数的调用	147
7.4.1	函数调用的一般形式	147
7.4.2	函数调用的方式	148
7.4.3	对被调用函数的声明和函数原型	148
7.5	函数的嵌套调用	152
7.6	函数的递归调用	153
7.7	数组作为函数参数	160
7.7.1	数组元素作函数实参	160
7.7.2	数组名作函数参数	161
7.7.3	用多维数组名作函数参数	165
7.8	变量的作用域——局部变量和全局变量	166
7.8.1	局部变量	166
7.8.2	全局变量	167
7.9	变量的存储方式和生存期	170
7.9.1	局部变量的存储类别	171
7.9.2	全局变量的存储类别	174

7.9.3 存储类别小结	177
7.10 关于变量的声明和定义	179
7.11 内部函数和外部函数	180
7.11.1 内部函数	180
7.11.2 外部函数	181
本章小结	183
习题	184
第8章 指针	186
8.1 地址和指针的概念	186
8.2 变量的指针和指向变量的指针变量	188
8.2.1 怎样定义指针变量	188
8.2.2 怎样引用指针变量	190
8.2.3 指针变量作为函数参数	192
8.3 通过指针引用数组	197
8.3.1 数组元素的指针	197
8.3.2 指针的运算	197
8.3.3 通过指针引用数组元素	199
8.3.4 用数组名作函数参数	203
8.3.5 通过指针引用多维数组	210
8.4 通过指针引用字符串	218
8.4.1 字符串的表示形式	218
8.4.2 字符指针作函数参数	221
8.4.3 对使用字符指针变量和字符数组的讨论	224
8.5 指向函数的指针	226
8.5.1 什么是函数指针	226
8.5.2 用函数指针变量调用函数	227
8.5.3 用指向函数的指针作函数参数	230
8.6 返回指针值的函数	233
8.7 指针数组和多重指针	235
8.7.1 指针数组的概念	235
8.7.2 指向指针数据的指针	238
8.8 动态内存分配与指向它的指针变量	241
8.8.1 什么是内存的动态分配	241
8.8.2 怎样建立内存的动态分配	241
8.8.3 void 指针类型	242
本章小结	244
习题	246

第 9 章 用户建立的数据类型	248
9.1 结构体类型和结构体变量	248
9.1.1 结构体类型	248
9.1.2 定义结构体类型变量	250
9.1.3 引用结构体变量	251
9.1.4 结构体变量的初始化	252
9.2 结构体数组	253
9.2.1 定义结构体数组	253
9.2.2 结构体数组的初始化	254
9.2.3 结构体数组应用举例	255
9.3 结构体指针	256
9.3.1 指向结构体变量的指针变量	257
9.3.2 指向结构体数组的指针	258
9.3.3 结构体变量和指向结构体的指针作函数参数	260
9.4 用指针处理链表	262
9.4.1 链表概述	262
9.4.2 建立简单的静态链表	263
9.4.3 建立动态链表	264
9.4.4 输出链表	267
9.5 共用体类型	269
9.5.1 什么是共用体类型	269
9.5.2 共用体变量的引用方式	270
9.5.3 共用体类型数据的特点	271
9.6 枚举类型	274
9.7 用 typedef 命名类型	278
本章小结	280
习题	281
第 10 章 文件操作	283
10.1 C 文件的有关概念	283
10.1.1 什么是文件	283
10.1.2 文件名	284
10.1.3 文件的分类	284
10.1.4 文件缓冲区	285
10.1.5 文件指针	285
10.1.6 文件的位置指针	286
10.2 文件的打开与关闭	287
10.2.1 文件的打开	287