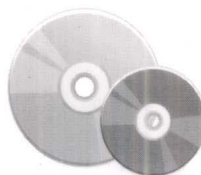


- 1 由浅入深：从C语言基础开始讲解，逐渐深入到高级技术
- 2 内容全面：覆盖了C语言中的所有重要知识点
- 3 重点突出：对C语言实现数据结构的概念进行了重点讲解
- 4 实例丰富：用320余个示例详细展示了C语言程序设计的重点
- 5 视频讲解：配大量多媒体教学视频进行讲解，学习效率高

C语言

从初学到精通

徐志伟 陈曦 王永静 等编著



34小时
多媒体教学视频



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

• 从初学到精通系列 •

C 语言

从初学到精通

徐志伟 陈曦 王永静 等编著

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

C 语言是一种计算机程序设计语言,它功能强大,可以用来编写系统软件及应用软件,应用范围很广泛。作为一门计算机编程语言,C 语言有自己的规范,所以对初学者而言概念比较多、不易理解。

本书分为 5 篇,共 23 章,循序渐进地从基本概念到具体实践讲述 C 语言,并对具体知识点进行详细的实例讲解。第一部分包括第 1~2 章,主要讲述 Turbo C 的安装、开发环境的介绍,以及如何在 Turbo C 中调试运行程序。第二部分包括第 3~7 章,主要介绍 C 语言中的基本概念,变量与常量、运算符与表达式、输入输出函数、C 语言三种基本结构。第三部分包括第 8~11 章,介绍 C 语言中的重点与难点,包括数组、函数、指针、结构体与共用体。第四部分包括第 12~20 章,介绍数据结构的相关内容,包括栈、队列、串、树、图,这几部分内容是数据结构的重点。第五部分包括第 21~23 章,介绍 C 语言中文件及编译部分,以及错误调试的内容。

本书的特点是不仅详细介绍了 C 语言中各个概念,而且在每个知识点后都配套有实例讲解,详细描述 C 语言中各个概念的实际应用及注意点,非常实用,方便读者模拟实践。本书适合没有编程基础的 C 语言初学者;正在学习 C 语言基础或深入了解数据结构的用户也可以阅读;并可作为大、中专院校师生或培训班教材;对于开发人员,本书也可以作为参考手册。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

C 语言从初学到精通 / 徐志伟等编著. —北京:电子工业出版社, 2010.6
(从初学到精通系列)
ISBN 978-7-121-10639-2

I. ①C… II. ①徐… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 056615 号

责任编辑:高洪霞

印 刷:北京天宇星印刷厂

装 订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:36.75 字数:898 千字

印 次:2010 年 6 月第 1 次印刷

印 数:4000 册 定价:69.00 元(含 DVD 光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

从此初学，走向精通

*The book is the nourishment in the whole world.
Life the inside have no book, seem to have no sunlight;
The inside of the intelligence has no book, seem the bird have no wing.*

书籍是全世界的营养品。生活里没有书籍，
就好像没有阳光；智慧里没有书籍，就好像鸟儿没有翅膀。

——莎士比亚



为什么要写这样一本书

C 语言是使用非常广泛的基础语言，从推出以来就备受广大开发者的关注。同时，它也被认为是最适合编程开发者入门的语言。本书总结了笔者学习 C 语言的经验，又结合多年开发经验，详细讲解了 C 语言的基础内容。本书从基本概念入手，循序渐进地讲解了 C 语言的编程技术；同时，还花了大量的篇幅讲解数据结构的相关内容。如果读者希望编写高质量的程序代码，数据结构是十分重要的内容；如果读者希望深入研究算法，数据结构也是最基本的内容。本书的第四部分详细讲解了如何使用 C 语言实现数据结构。和其他讲解数据结构的书籍强调基本概念和理论不同，本书在讲解数据结构的时候，依然十分重视结合实际开发。让读者了解到各种不同的数据结构的适用方法，以及如何从实际问题中抽象出相应的数据结构。在讲解具体实例的时候，本书还对其中关键代码或内容进行解析，有助于读者熟练掌握相关概念和知识。

本书的特点

1. 细致体贴的讲解

本书假定读者没有任何的编程技能基础，讲解时尽可能地站在读者的角度，循序渐进，由浅入深，用准确的语言总结概念►用直观的图示演示过程►用详细的注释解释代码►用形象的比方帮助记忆。效果如下：

① **知识点介绍** 全面、准确、清晰是其显著特点，一般放在每一节开始位置，让零基础的读者了解相关概念，顺利入门。

② **范例** 本书精心选择的编程示例，可实践性强，注意让读者能从实例运用中举一反三，既有利于初学者对基本知识点的理解和掌握，又锻炼了读者的编程能力。

③ **范例代码** 与范例编号对应，层次清楚、语句简洁、注释丰富，体现了代码优美的原则，有利于读者养成良好的代码编写习惯。读者可以直接根据这些源程序快速编写程序，直接切入相关应用。

④ **运行结果** 对范例给出运行结果和对应图示，帮助读者更直观地理解范例代码、验证结果。

⑤ **代码解析** 本书对范例代码中的关键代码配以图示并进行详细解释，有助于读者掌握相关概念和知识。

⑥ **贴心的提示** 为了便于读者阅读，全书还穿插着一些技巧、提示等小贴士，体例约定如下：

- **提示：**通常是一些贴心的提醒，让读者加深印象或提供建议，或者解决问题的方法。
- **注意：**提出学习过程中需要特别注意的一些知识点和内容，或者相关信息。
- **警告：**对操作不当或理解偏差将会造成的灾难性后果做警示，以加深读者印象。

⑦ **小结** 每章最后提供对本章所介绍知识的简单总结，有利于读者加深印象，巩固总结。

经作者多年的培训和授课证明，以上讲解方式是最适合读者学习的方式，读者按照这种方式，会非常轻松、顺利地掌握本书知识。

①

②

③

④

⑤

15.4 队列的循环链表表示与实现

循环队列是队列的一种顺序存储表示，和顺序栈类似。在使用单链队列时就会发现这样一个问题：当队列小于入队时，在队列上方的空间就会空出来，而又有元素要插入时，就要开辟新空间，这样无疑会浪费空间资源。这里就引入循环队列的概念，充分利用队列上方空出来的那部分空间。

15.4.2 循环队列的操作

循环队列基本操作比较多，这里逐一举例介绍。首先要构造一个空队列，即初始化，才能进行一系列的操作。

【范例 15-9】 循环队列基本操作综合示例。

```
代码 15-9
01 typedef int QElemType;
02 typedef int Status;
03 #include<stdio.h>
04 #include<stdlib.h>
05 #define OK 1
06 #define TRUE 1
07 #define ERROR 0
08 #define OVERFLOW 0
09 #define NULL 0
10 #define MAXSIZE 100
11 struct queue
12 { QElemType *base;
13   int front;
14 };
15
16 按组合键 Ctrl+F9 运行后，从键盘输入如下内容。
17 3 0 2 3 9 54 Enter
18
19 按组合键 Alt+F5 显示结果，如图 15-9 所示。
```



图 15-9 循环队列综合示例运行结果

【代码解析】

- 第 43~61 行是主函数，第 48 行要从键盘获取 5 个数据作为队列的元素。第 49~53 行将每个元素插入到队列中。
- 第 54 行调用删除元素的函数，因为队列是在队头删除元素的，所以这里要删除的元素第 55 行打出来。

第 15 章 队列

➤ 第 54~60 行输出这个队列中的元素，但是第 54 行已删除了一个元素，所以这里输出的队列元素只有 4 个，因为要整个输出队列中的元素，所以这里使用的是将队列中的元素一个删除，在删除的过程中将元素打印出来。

1. <x:transform>标签

使用该标签可以轻松地实现 XML 到 XSLT 的转化，其具备一种语法格式，不管哪种格式，都需要指定 XSLT 文件的地址和 XML 文件的地址。具体的参数说明见表 4-22。

【语法 1】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 2】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 3】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 4】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 5】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 6】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 7】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 8】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 9】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 10】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 11】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 12】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 13】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 14】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 15】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 16】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 17】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 18】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 19】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 20】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 21】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 22】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 23】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 24】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 25】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 26】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 27】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 28】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 29】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 30】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 31】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 32】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 33】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 34】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 35】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 36】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 37】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 38】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 39】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 40】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 41】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 42】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 43】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 44】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 45】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 46】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 47】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 48】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 49】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 50】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 51】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 52】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 53】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 54】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 55】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 56】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 57】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 58】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 59】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 60】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 61】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 62】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 63】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 64】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 65】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 66】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 67】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 68】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 69】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 70】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 71】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 72】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 73】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 74】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 75】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 76】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 77】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 78】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 79】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 80】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 81】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 82】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 83】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 84】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 85】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 86】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 87】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 88】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 89】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 90】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 91】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 92】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 93】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 94】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 95】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 96】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 97】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 98】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 99】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

【语法 100】 <x:transform doc="xmlDoc" xslt="XSLTstylesheet" [docSystemId="xmlSystemId"] [result="result"] [var="name"] [scope="scopeName"] [xsltSystemId="xsltSystemId"] />

⑥

⑦

2. 实用超值的 DVD 光盘

为了帮助读者比较直观地学习，本书附赠 DVD 光盘，内容包括多媒体视频、电子教案（PPT）、编程参考宝典电子书、各章习题答案和实例源代码等。

● 多媒体视频

配有长达 9 小时手把手教学视频，讲解关键知识点界面操作和书中的一些综合练习题。作者亲自配音、演示，手把手教会读者使用。此外，还随书免费赠送 25 小时相关知识多媒体视频，帮助读者巩固和加深对所学内容的理解和掌握。

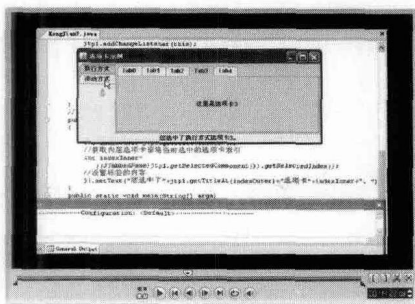
● 电子教案（PPT）

本书可以作为高校相关课程的教材或课外辅导书，所以笔者特别为本书制作了电子教案（PPT），以方便老师教学使用。

- ④ C语言从初学到精通
- ④ PHEI Broadview 2008专业书目 第一期
- ④ pheibook
- ④ 编程参考宝典电子书
- ④ 电子教案（PPT）
- ④ 多媒体视频
- ④ 源代码
- ④ 赠送多媒体视频
- ④ 基础知识多媒体视频

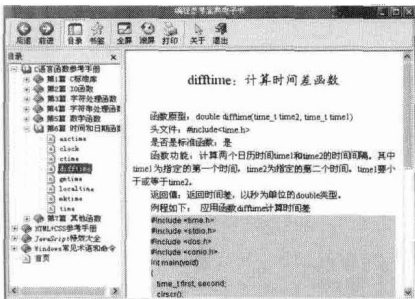
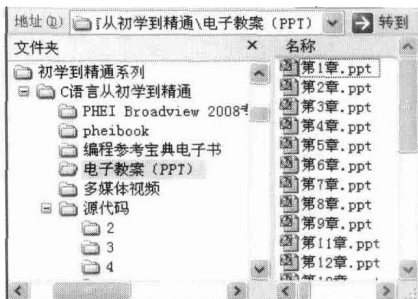
4

此为试读，需要完整PDF请访问：www.ertongbook.com



● 编程参考宝典电子书

为方便广大读者学习，特别制作了编程开发参考电子书，供读者查阅和参考。



3. 提供完善的技术支持

本书提供了论坛：<http://www.rzchina.net>，读者可以在上面提问交流。另外，论坛上还有一些小的教程、视频动画和各种技术文章，可帮助读者提高开发水平。

4. 丰富的额外素材下载

相关的开发素材文件，在 www.broadview.com.cn 提供下载。

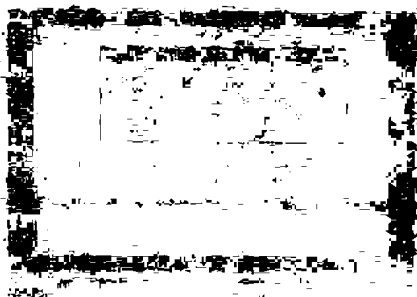
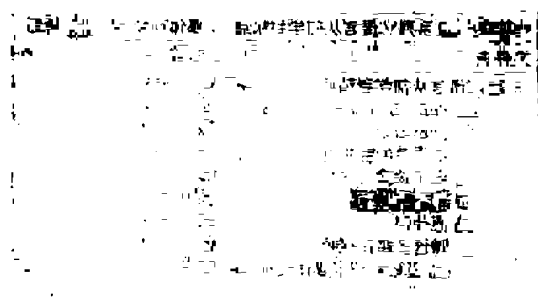
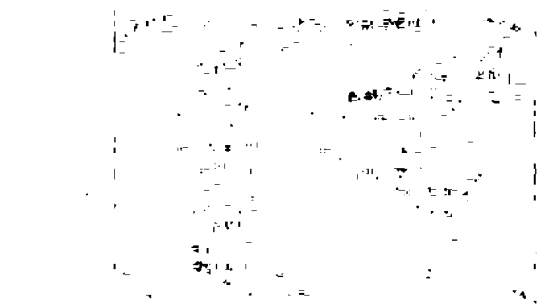
本书适合哪些读者阅读

- C 语言程序设计的初学者；
- 需要在了解 C 语言后进一步了解数据结构的学员；
- 想系统地学习 C 语言知识的学员；
- 需要用 C 语言程序进行毕业设计的计算机专业学生；
- 大、中院校的培训人员。

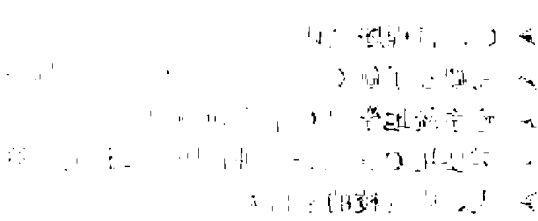
本书作者

本书第一部分、第二部分、第四部分由徐志伟、王霞、陈曦编写，第三部分由朱晶、王永静编写，第五部分由崔金英、刘娜编写。其他参与编写的人员有昊燃、方振宇、陈冠佐、傅奎、陈勤、梁洋洋、毕梦飞、陈庆。

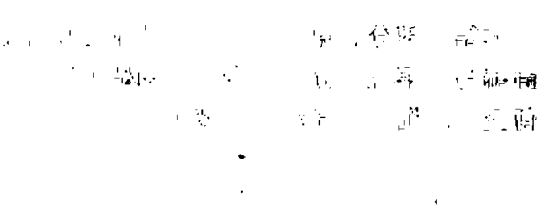
编者

[illegible][illegible][illegible]

三、**經濟學與社會科學**



青 科 生 主



目 录

第一部分 C 语言开发环境篇

第 1 章 Turbo C 介绍	24
------------------------	----

本章主要介绍了安装 Turbo C 3.0 的步骤、相关设置,以及环境界面中各个菜单的详细介绍。读者在学习本章内容时,尤其是在安装 Turbo C 3.0 时,最好在计算机上跟着介绍的步骤逐步操作,学会安装并且了解整个界面,以及一些菜单的使用等。

1.1 Turbo C 的配置与安装	24
1.1.1 Turbo C 简介	24
1.1.2 安装 Turbo C 3.0	24
1.2 初识 Turbo C 环境界面	27
1.2.1 设置 Turbo C	27
1.2.2 界面简介	29
1.3 本章小结	32

第 2 章 初识 C 程序	33
---------------------	----

本章主要介绍了第一个 C 语言程序,介绍程序的各个组成部分及作用。然后利用 Turbo C 3.0 环境调试这个程序,其中重点讲述了如何保存、调试、检查错误、运行显示结果。另外本章还涉及了一些算法的内容,并介绍了如何应用流程图表示程序。

2.1 第一个 C 程序	33
2.2 C 语言的后缀名	34
2.3 C 语言的上机步骤	35
2.3.1 程序运行流程	35
2.3.2 输入源程序	35
2.3.3 运行源程序	36
2.3.4 调试检查错误	36
2.3.5 显示运行结果	37
2.4 打开光盘中源程序的步骤	37
2.5 算法及表示	38
2.5.1 算法	38
2.5.2 算法的表示	39
2.6 本章小结	42

第二部分 C 语言基础知识篇

第 3 章 变量与常量 44

本章主要介绍 C 语言中变量、常量，以及数据类型。首先介绍了 C 语言的基本符号、标识符，这部分是 C 语言的基础，由此再引入变量，变量的定义及变量的常见数据类型。学会了变量的基本知识，读者就不难理解常量及其数据类型。本章知识结构安排一环扣一环，联系比较紧密，希望读者认真学好每一节，为以后的 C 语言学习打下坚实的基础。

3.1 C 语言的基本语言单位.....	44
3.1.1 基本符号.....	44
3.1.2 标识符.....	45
3.1.3 关键字.....	46
3.2 变量及变量的数据类型.....	48
3.2.1 基本概念.....	48
3.2.2 变量的定义.....	48
3.2.3 整型变量.....	49
3.2.4 实型变量.....	51
3.2.5 字符型变量.....	52
3.3 常量及常量的数据类型.....	53
3.3.1 常量的定义.....	53
3.3.2 整型常量.....	54
3.3.3 实型常量.....	54
3.3.4 字符型常量.....	56
3.3.5 字符串常量.....	56
3.3.6 符号常量.....	57
3.3.7 转义字符.....	58
3.3.8 地址常量.....	60
3.4 综合示例.....	61
3.5 本章小结.....	63

第 4 章 运算符、表达式和语句 64

本章主要介绍了 C 语言中的运算符，具体指的是算术运算符、逻辑运算符和关系运算符、逗号运算符、赋值运算符、位运算符、自增自减运算符等，这些知识都是非常基础的；另外本章在这些基础上介绍了运算符的表达式及语句。在以后编写复杂程序时，这些内容都是必备的知识，所以学好这部分内容是 C 程序编写基础中的基础。

4.1 算术运算符与表达式.....	64
4.1.1 算术运算符.....	64
4.1.2 算术表达式.....	65

4.2	自增自减运算符与表达式	66
4.3	赋值运算符与表达式	67
4.3.1	类型转换	67
4.3.2	复合赋值运算符	68
4.3.3	赋值表达式	69
4.4	逗号运算符与表达式	70
4.5	关系运算符与表达式	71
4.5.1	关系运算符	71
4.5.2	关系表达式	71
4.6	条件运算符与表达式	72
4.7	逻辑运算符与表达式	74
4.7.1	逻辑运算符	74
4.7.2	逻辑表达式	76
4.8	位运算符与表达式	77
4.8.1	按位与运算	77
4.8.2	按位或运算	78
4.8.3	按位异或运算	79
4.8.4	按位求反运算	80
4.8.5	左移运算	80
4.8.6	右移运算	81
4.9	综合示例	82
4.10	本章小结	87
第 5 章 输入/输出函数		88
输入输出函数的规则比较烦琐，而作为一项最基本的操作，在设计程序时总会有输入输出这一环节，所以本章是比较重要的一个基本内容。在学习相关规则时，着重记住一些规则就可以掌握输入/输出的关键所在。要了解和熟练掌握这部分内容，最好的办法就是在不断编写和调试程序的过程中掌握输入/输出的规则。		
5.1	字符型数据的输入/输出	88
5.1.1	字符输出	88
5.1.2	字符输入	90
5.1.3	字符串输入/输出	91
5.2	标准格式输出	91
5.2.1	输出字符	92
5.2.2	输出整数	95
5.2.3	输出浮点数	97
5.3	格式输入	99

5.3.1 格式控制中无间隔	100
5.3.2 格式控制中以逗号间隔	100
5.3.3 格式控制中有其他字符	101
5.4 本章小结	102

第 6 章 顺序结构和选择结构..... 103

顺序结构是 C 语言程序设计三种基本结构中最简单的一种，但是它在 C 语言的编程中起着举足轻重的作用，几乎任何一个程序都有顺序结构的参与。选择结构的重点就是要找好条件，将可能的情况考虑周全，用逻辑表达式或关系表达式将条件限制准确。

6.1 顺序结构	103
6.1.1 顺序结构流程图	103
6.1.2 顺序结构应用举例	103
6.2 选择结构	105
6.2.1 if 语句一般形式	105
6.2.2 if 一般形式应用举例	106
6.2.3 if-else 结构	108
6.2.4 if-else 结构应用举例	109
6.2.5 if-else-if 结构	111
6.2.6 if-else-if 结构应用举例	112
6.2.7 if 语句的嵌套	116
6.2.8 if 语句的嵌套应用举例	117
6.2.9 switch 语句一般形式	121
6.2.10 switch 语句运行规则	122
6.2.11 带 break 语句的 switch 语句	123
6.2.12 switch 语句应用举例	124
6.3 综合示例	131
6.4 本章小结	136

第 7 章 循环结构..... 137

循环结构是 C 语言中非常重要的结构之一，它可以快捷、非常节省人力地解决很多实际问题。这一章中，主要要求大家能够熟练掌握循环的三种语句，能够融会贯通地使用，尤其是注意比较三种语句之间的异同，对于很多问题，多加思考，多动手练习，亲自编写程序，打好基础。

7.1 for 语句	137
7.1.1 for 语句的一般形式	137
7.1.2 for 语句的其他形式	138
7.1.3 for 语句应用举例	139
7.2 while 语句	142

7.2.1	while 语句的一般形式	143
7.2.2	while 语句的特殊形式	143
7.2.3	while 语句的使用规则	143
7.2.4	while 语句应用举例	144
7.3	do-while 语句	146
7.3.1	do-while 语句的一般形式	146
7.3.2	do-while 语句的使用规则	147
7.3.3	do-while 语句应用举例	147
7.4	循环结构的嵌套	148
7.4.1	几种常用的循环嵌套形式	149
7.4.2	循环结构的嵌套的运行规则	150
7.4.3	循环结构嵌套应用举例	151
7.5	转移语句	152
7.5.1	break 语句	152
7.5.2	continue 语句	153
7.5.3	goto 语句	154
7.6	综合示例	155
7.7	本章小结	159

第三部分 C 语言重点知识篇

第 8 章	数组	162
-------	----	-----

本章主要介绍了数组方面的知识，分别从一维数组、二维数组、字符数组这三点展开讲解。一维数组是数组的基础，二维数组、多维数组及字符数组都是建立在它的基础上的。使用数组可以解决很多看起来很复杂的问题，简单易懂。

8.1	一维数组	162
8.1.1	一维数组的定义	162
8.1.2	一维数组初始化	163
8.1.3	一维数组元素的引用	164
8.1.4	一维数组应用举例	164
8.2	二维数组	168
8.2.1	二维数组的定义	168
8.2.2	二维数组初始化	169
8.2.3	二维数组元素的引用	170
8.2.4	二维数组应用举例	171
8.3	字符数组	175

8.3.1	字符数组的定义	176
8.3.2	字符数组初始化	176
8.3.3	字符数组的输出	177
8.3.4	字符数组的输入	178
8.3.5	字符串操作函数	180
8.3.6	字符数组应用举例	183
8.4	综合示例	185
8.5	本章小结	192

第9章 函数 193

本章主要介绍函数的相关内容，包括概念、定义、声明及调用等内容，程序中用户自定义一些函数，使得程序模块化，可读性强，而且优化程序。另外还介绍了函数的嵌套调用及递归调用，以及函数参数传递的内容，这部分内容中比较难的是数组名作为参数传递的时候要注意传址与传值的差别。

9.1	函数的定义	193
9.2	函数的返回值	194
9.3	函数的声明	195
9.4	函数的调用	196
9.4.1	函数的调用格式	196
9.4.2	函数的调用方式	197
9.4.3	函数的嵌套调用	198
9.4.4	函数的递归调用	199
9.5	函数参数传递	203
9.5.1	函数的形式参数与实际参数	203
9.5.2	函数的参数传递	204
9.6	数组与函数	205
9.6.1	数组元素作为函数参数	205
9.6.2	一维数组名作为函数参数	206
9.6.3	二维数组名作为函数参数	208
9.7	变量的存储类别	210
9.7.1	动态存储与静态存储	210
9.7.2	全局变量与局部变量	211
9.8	全局变量	214
9.8.1	在函数外定义	214
9.8.2	extern 变量——扩展变量的作用域	215
9.8.3	static 变量——静态变量	218
9.9	局部变量	218
9.9.1	auto 变量——自动变量	218

9.9.2	register 变量——寄存器变量	219
9.9.3	static 变量——静态局部变量	220
9.10	综合示例	222
9.11	本章小结	227
第 10 章	指针	228
本章介绍了指针的基本概念和初步应用。应该说明，指针是 C 语言中重要的概念，也是 C 语言的一个特色。由于指针的错误应用甚至会致使整个程序遭受破坏，因此，在使用指针时一定要十分小心，另外读者还应该多实践，在实际应用中熟悉它。		
10.1	地址和指针的概念	228
10.1.1	内存中数据的直接访问方式	228
10.1.2	内存中数据的间接访问方式	229
10.1.3	两种内存数据访问方式的举例	229
10.1.4	地址和指针的概念	230
10.2	变量的指针与指针变量	230
10.2.1	指针变量的定义	231
10.2.2	指针变量的引用	232
10.2.3	“&”和“*”运算符的结合方向	234
10.2.4	指针变量作为函数参数	235
10.3	数组与指针	239
10.3.1	指向数组元素的指针	240
10.3.2	通过指针引用数组元素	241
10.3.3	指针引用数组元素应用举例	242
10.3.4	使用指针变量时需注意的问题	244
10.3.5	指向数组的指针变量的运算	246
10.3.6	指向二维数组的指针变量	247
10.4	字符串与指针	250
10.4.1	用字符数组表示字符串	250
10.4.2	用字符指针表示字符串	251
10.4.3	表示字符串两种方法应用举例	252
10.4.4	字符串指针作函数参数	254
10.4.5	对使用字符指针变量和字符数组的讨论	256
10.5	指向函数的指针	259
10.5.1	用函数指针变量调用函数	259
10.5.2	用函数指针变量调用函数需注意问题	261
10.5.3	用指向函数的指针作函数参数	261

10.6	返回指针值的函数	264
10.7	指针数组和指向指针的指针	266
10.7.1	指针数组的概念	266
10.7.2	指向指针的指针	268
10.8	有关指针的数据类型和指针运算的小结	270
10.8.1	有关指针的数据类型的小结	270
10.8.2	指针运算小结	271
10.9	综合示例	272
10.10	本章小结	287

第 11 章 结构体与共用体

288

本章介绍了用户定义的数据：结构体、共用体和枚举类型，以及定义数据类型的 typedef 关键字的使用。这几种数据类型使 C 语言功能更加强大，编程更加简便。要注意类型定义与定义变量的区别，以及在使用过程中应该注意的问题。

11.1	结构体变量	288
11.1.1	结构体变量定义的方法	289
11.1.2	结构体变量的初始化	292
11.1.3	结构体变量的引用	293
11.1.4	结构体变量的数据传递	295
11.1.5	用结构体构成链表	296
11.2	结构体数组	298
11.2.1	结构体数组的定义	298
11.2.2	结构体数组的初始化	299
11.3	结构体与指针	302
11.3.1	指向结构体变量的指针	302
11.3.2	指向结构体数组的指针	303
11.4	结构体指针与链表	304
11.4.1	链表概述	304
11.4.2	对链表的操作	306
11.4.3	建立链表	306
11.4.4	取链表中数据	310
11.4.5	插入数据	310
11.4.6	删除数据	312
11.4.7	清空链表	314
11.4.8	合并链表	314
11.5	共用体	316

11.5.1 共用体类型定义的方法	316
11.5.2 共用体变量的引用	319
11.6 枚举类型	321
11.7 用 typedef 定义类型	323
11.8 综合示例	324
11.9 本章小结	329

第四部分 数据结构篇

第 12 章 数据结构概论	332
---------------	-----

本章主要介绍了数据结构的一些基本概念和术语，包括数据的基本概念、数据的类型及算法。这是数据结构部分的基础，为下面学习数据结构提供了基本概念。学习数据结构时重点也在于掌握数据结构的结构特点。

12.1 基本概念	332
12.1.1 数据结构的定义	332
12.1.2 数据的逻辑结构	333
12.1.3 数据的存储结构	334
12.1.4 数据的运算结构	335
12.2 数据类型和抽象数据类型	336
12.2.1 数据类型	336
12.2.2 抽象数据类型	336
12.3 算法及算法分析	337
12.3.1 算法特性	337
12.3.2 算法描述	338
12.3.3 算法设计的要求	338
12.3.4 算法分析	338
12.4 本章小结	341

第 13 章 线性表	342
------------	-----

在本章中，首先学习了关于线性表的一些基本概念，包括什么是线性表，它的一些基本操作集合是什么。然后，分别介绍了线性表的两种表示形式，一种是线性表的顺序存储方式，即顺序表；另一种是线性表的链式存储方式，即链表。同时分别给出了相应的操作和具体的程序。最后，比较了两种存储方式的优缺点，并给出使用意见。

13.1 线性表基本概念	342
13.2 线性表的顺序存储结构	343
13.2.1 创建顺序表	345
13.2.2 操作顺序表	346

13.2.3 顺序表的插入操作	348
13.2.4 顺序表的删除操作	349
13.2.5 顺序表的综合案例	351
13.3 线性表的链式存储结构	352
13.3.1 创建链表	354
13.3.2 操作单链表	356
13.3.3 单链表的插入操作	358
13.3.4 单链表的删除操作	359
13.3.5 单链表的综合案例	361
13.3.6 循环链表	363
13.4 两种存储方式的比较	364
13.5 本章小结	364

第 14 章 栈

栈是特殊的线性表，是操作受限的线性表，它的操作是线性表操作的子集。但是正是因为栈具有这种特点，所以在解决一些特殊问题时可使问题变得简单。所以本章主要的目标是学会栈的各种操作，包括初始化、插入、删除、取栈顶元素等，还要学会利用栈的特殊性来解决问题。

14.1 生活中的例子	365
14.2 栈的定义	365
14.3 栈的表示与实现	366
14.3.1 栈的表示	367
14.3.2 栈的操作	367
14.4 栈的应用	371
14.5 本章小结	374

第 15 章 队列

队列与栈相同，都是操作受限的线性结构，所不同的是，栈的操作是“先进后出”，即先插入到栈的元素先到栈底的，在删除元素时是先删掉后进的元素，先进的元素是后删除的；而队列的操作是“先进先出”，即队列是一端插入一端删除的，先进入的元素从另一端删除，遵循先进先出的规则。

15.1 生活中的例子	375
15.2 队列的定义	376
15.3 队列的单链式表示与实现	377
15.3.1 链队列的定义	377
15.3.2 单链队列的操作	377
15.4 队列的循环链式表示与实现	382
15.4.1 循环队列的定义	382
15.4.2 循环队列的操作	383