



工业和信息化人才培养规划教材  
Industry And Information Technology Training Planning Materials

Technical And Vocational Education  
高职高专计算机系列

# 数据结构 实例教程 (C 语言版)

## Data Structure in C

李刚 冯卫刚 ◎ 主编

顾理琴 郜继红 ◎ 副主编 张洪斌 ◎ 主审

实例教学，深入浅出  
视频动画，直观易懂  
配备“多媒体教学系统”和课程网站



人民邮电出版社  
POSTS & TELECOM PRESS





工业和信息化人才培养规划教材  
Industry And Information Technology Training Planning Materials

Technical And Vocational Education

高职高专计算机系列

# 数据结构 实例教程 (C语言版)

Data Structure in C

李刚 冯卫刚 ◎ 主编

顾理琴 郜继红 ◎ 副主编 张洪斌 ◎ 主审

顾江洪 副主编

顾理琴 副主编

顾洪斌 主审

王 王 蒋峰升 参审

王 王 蒋峰升 参审

王 王 蒋峰升 参审

王 王 蒋峰升 参审

王 王 蒋峰升 参审

王 王 蒋峰升 参审

王 王 蒋峰升 参审

王 王 蒋峰升 参审

王 王 蒋峰升 参审

人民邮电出版社

北京



## 图书在版编目(CIP)数据

数据结构实例教程：C语言版 / 李刚, 冯卫刚主编

— 北京：人民邮电出版社，2013.9

工业和信息化人才培养规划教材·高职高专计算机系列

ISBN 978-7-115-32281-4

I. ①数… II. ①李… ②冯… III. ①数据结构—高等职业教育—教材②C语言—程序设计—高等职业教育—教材 IV. ①TP311.12②TP312

中国版本图书馆CIP数据核字(2013)第169585号

## 内 容 提 要

本书以“学生为中心”的理念为指导思想，内容精练，通俗易懂，既便于教学，又适合自学。教材每章均配有知识实践项目，并配有数据结构综合应用实例。

本书主要分为两大部分：前10章为基础知识部分，第11章为综合应用部分。基础知识部分包括线性结构模块、非线性结构模块和简单应用模块。综合应用部分包括新生报到信息注册系统设计模块、万达停车场管理系统设计模块和最短时间旅游路线查询系统设计模块。

本书采用模块化的编写方法，体现“易教、易学、易练”的特色，让学生明白“是什么”→“怎么做”→“怎么用”3个环节。第一步：采用生活化的实例介绍模块涉及的基础知识。第二步：采用计算机描述语言讲解模块涉及的数据结构和基本操作。第三步：系统应用知识提高相关模块的结构分析与应用能力。最后还通过综合实践让学生有思考和扩展的空间，达到学以致用、学以致用的教学目的。

本书可作为高职高专院校计算机专业的教材，也可供专接本学生参考。

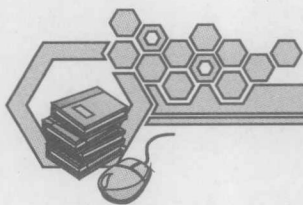
- 
- ◆ 主 编 李 刚 冯卫刚  
副 主 编 顾理琴 郜继红  
主 审 张洪斌  
责任编辑 王 威  
责任印制 沈 蓉 焦志炜
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号  
邮编 100061 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
大厂聚鑫印刷有限责任公司印刷
- ◆ 开本：787×1092 1/16  
印张：10.75 2013年9月第1版  
字数：269千字 2013年9月河北第1次印刷
- 

定价：32.00元（附光盘）

读者服务热线：(010)67170985 印装质量热线：(010)67129223

反盗版热线：(010)67171154

广告经营许可证：京崇工商广字第0021号



数据结构是软件技术、计算机应用技术等计算机类专业的一门重要的专业基础课程，它是软件开发的基础，提高学生逻辑思维能力的核心。使读者学会如何把现实世界的问题转化为计算机内部的表示和处理。本书是为“数据结构”课程编写的教材，其内容选取符合高职高专教学大纲要求。

本书主编具有 11 年的数据结构课程一线教学经验，对高职高专和专接本学生学习有自己的教学方法和教学理念，本书采用模块化的编写方法，体现“易教、易学、易练”的特色，让学生明白“是什么”→“怎么做”→“怎么用”三个环节。第一步：采用生活化的实例介绍模块涉及到的基础知识。第二步：采用计算机描述语言讲解模块涉及的数据结构和基本操作。第三步：系统应用知识提高相关模块的结构分析与应用能力。最后还通过综合实践让学生够有思考和扩展的空间，达到学以致用教学目的。

本书主要分为两大部分：前 10 章为基础知识部分，第 11 章为综合应用部分。基础知识部分包括线性结构模块：介绍了线性表、栈与队列、串、数组和广义表；非线性结构模块：介绍了树和图；简单应用模块：介绍了查找、排序和文件。综合应用部分包括新生报到信息注册系统设计模块：提高线性表知识的综合应用能力；万达停车场管理系统设计模块：提高栈和队列知识的综合应用能力；最短时间旅游路线查询系统设计模块：提高图知识的综合应用能力。

全书由李刚、冯卫刚任主编，顾理琴、郜继红任副主编，张洪斌主审。本书编写分工如下：李刚编写第 1 章、第 6 章、第 7 章、第 8 章，冯卫刚编写第 11 章综合应用部分，顾理琴编写第 2 章、第 3 章，郜继红编写第 9 章、第 10 章和知识实践部分，程乐编写第 4 章、第 5 章，陆彩霞编写知识实践部分。本书在编写过程中还得到了管曙亮和刘万辉老师的帮助和支持，在此表示衷心的感谢。

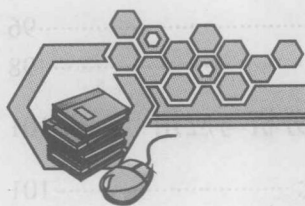
本书配套光盘提供丰富的教学资源，包含程序完整代码、拓展实例和相关自学材料。

由于编者水平有限，书中难免存在不妥之处，恳请广大读者批评指正。

任课老师可访问人民邮电出版社教学服务与资源网（[www.ptpedu.com.cn](http://www.ptpedu.com.cn)）下载电子教案 PPT 等教学辅导材料。

作者

于 2013 年夏



## 第 1 章 绪论及 C 语言介绍..... 1

- 1.1 基本概念与术语..... 1
- 1.2 学习数据结构的意义..... 3
- 1.3 算法的描述和分析..... 5
- 1.4 C 语言相关知识介绍..... 6
- 知识实践一 学生管理系统登录  
模块设计..... 9
- 习题..... 10

## 第 2 章 线性表的结构

### 分析与应用..... 12

- 2.1 线性表的逻辑结构..... 12
- 2.2 线性表的顺序存储结构..... 13
  - 2.2.1 顺序表定义及地址计算..... 13
  - 2.2.2 顺序表基本运算..... 13
- 2.3 线性表的链式存储结构..... 16
  - 2.3.1 单链表..... 16
  - 2.3.2 循环链表..... 18
- 2.4 顺序表和链表的比较..... 19
- 知识实践二 学生管理系统成绩插入、删除模块设计..... 20
- 知识实践三 嵌入式系统中任务的创建和删除模拟设计..... 23
- 习题..... 25

## 第 3 章 栈和队列的结构

### 分析与应用..... 27

- 3.1 栈..... 27
  - 3.1.1 栈的定义及基本运算..... 27
  - 3.1.2 顺序栈及操作实现..... 28
  - 3.1.3 链栈及操作实现..... 30
- 3.2 队列..... 32
  - 3.2.1 队列的定义及基本运算..... 32

3.2.2 顺序队列及操作实现..... 32

3.2.3 链队列及操作实现..... 35

- 知识实践四 计算器中进制转换  
模块设计..... 37
- 知识实践五 单片机或者嵌入式  
系统中断模拟设计..... 39
- 知识实践六 学生舞会舞伴  
配对系统设计..... 42
- 习题..... 44

## 第 4 章 字符串的结构

### 分析与应用..... 46

- 4.1 串的定义及其运算..... 46
  - 4.1.1 串的基本概念..... 46
  - 4.1.2 串的基本运算..... 47
- 4.2 串的存储结构..... 47
  - 4.2.1 串的顺序存储结构..... 47
  - 4.2.2 串的链式存储结构..... 48
  - 4.2.3 子串的定位运算..... 48
- 知识实践七 学生管理系统  
家庭情况模块设计..... 49
- 习题..... 50

## 第 5 章 二维数组及广义表的结构分析

### ..... 51

- 5.1 二维数组的存储结构及  
求址方法..... 51
- 5.2 矩阵的压缩存储..... 52
  - 5.2.1 特殊矩阵..... 52
  - 5.2.2 稀疏矩阵..... 53
- 5.3 广义表的概念..... 54
- 知识实践八 求二维数组元素在  
内存中的存储位置..... 55
- 习题..... 56

## 第 6 章 树和二叉树的结构

### 分析与应用 ..... 57

- 6.1 树的概念 ..... 57
- 6.2 二叉树 ..... 59
  - 6.2.1 二叉树的定义 ..... 59
  - 6.2.2 二叉树的性质 ..... 60
  - 6.2.3 二叉树的存储结构 ..... 60
- 6.3 二叉树的遍历 ..... 64
- 6.4 线索二叉树 ..... 66
- 6.5 树和森林 ..... 67
  - 6.5.1 树、森林与二叉树的相互  
转换 ..... 67
  - 6.5.2 树的存储结构 ..... 69
  - 6.5.3 树和森林的遍历 ..... 71
- 6.6 哈夫曼树及其应用 ..... 72
  - 6.6.1 哈夫曼树的定义 ..... 72
  - 6.6.2 哈夫曼树的构造 ..... 73
  - 6.6.3 哈夫曼树编码 ..... 74

- 知识实践九 利用二叉树遍历实现  
学生成绩排序模块设计 ..... 74
- 习题 ..... 76

## 第 7 章 图的结构分析与应用 ..... 80

- 7.1 图的概念及相关术语 ..... 80
  - 7.1.1 图的概念 ..... 80
  - 7.1.2 图的相关术语 ..... 81
- 7.2 图的存储结构 ..... 83
  - 7.2.1 邻接矩阵表示法 ..... 83
  - 7.2.2 邻接表表示法 ..... 84
- 7.3 图的遍历 ..... 86
  - 7.3.1 深度优先遍历 ..... 87
  - 7.3.2 广度优先遍历 ..... 89
- 7.4 最小生成树 ..... 90
  - 7.4.1 普里姆算法 ..... 91
  - 7.4.2 克鲁斯卡尔算法 ..... 91
- 7.5 最短路径 ..... 92
  - 7.5.1 单源最短路径 ..... 92
  - 7.5.2 每一对顶点之间的  
最短路径 ..... 93

- 知识实践十 江苏省地级市组成的

- 地图遍历 ..... 96
- 习题 ..... 98

## 第 8 章 查找的分析与应用 ..... 101

- 8.1 基本概念 ..... 101
- 8.2 线性表查找 ..... 102
  - 8.2.1 顺序查找 ..... 102
  - 8.2.2 二分查找 ..... 103
  - 8.2.3 分块查找 ..... 105
- 8.3 二叉排序树 ..... 106
  - 8.3.1 二叉排序树定义 ..... 106
  - 8.3.2 二叉排序树的  
插入和生成 ..... 107
  - 8.3.3 二叉排序树的删除 ..... 108
  - 8.3.4 二叉排序树的查找 ..... 109
- 8.4 散列技术 ..... 109
  - 8.4.1 散列表的概念 ..... 109
  - 8.4.2 散列函数的构造方法 ..... 110
  - 8.4.3 处理冲突的方法 ..... 110

- 知识实践十一 利用二分法查找  
实现学生信息查询模块设计 ..... 111
- 习题 ..... 113

## 第 9 章 排序的分析与应用 ..... 114

- 9.1 排序的基本概念 ..... 114
  - 9.1.1 排序的定义 ..... 114
  - 9.1.2 相关概念 ..... 115
- 9.2 插入排序 ..... 116
  - 9.2.1 直接插入排序 ..... 117
  - 9.2.2 希尔排序 ..... 120
- 9.3 交换排序 ..... 121
  - 9.3.1 冒泡排序 ..... 121
  - 9.3.2 快速排序 ..... 123
- 9.4 选择排序 ..... 126
  - 9.4.1 直接选择排序 ..... 126
  - 9.4.2 堆排序 ..... 128
- 9.5 归并排序 ..... 131
- 9.6 各种内部排序算法的比较 ..... 133
- 知识实践十二 学生总成绩  
排序模块设计 ..... 134
- 习题 ..... 136

<b>第 10 章 文件</b> .....	138		
10.1 文件的概念.....	138		
10.1.1 文件的基本概念.....	138		
10.1.2 文件的逻辑结构和 物理结构.....	139		
10.1.3 文件的操作.....	139		
10.2 顺序文件.....	140		
10.2.1 存储在顺序存储器上的 顺序文件.....	140		
10.2.2 存储在直接存储器上的 顺序文件.....	140		
10.3 索引文件.....	140		
10.4 索引顺序文件.....	141		
10.4.1 索引顺序文件的特点.....	141		
10.4.2 VSAM 文件的 组织方法.....	141		
10.5 散列文件.....	142		
10.5.1 散列文件的组织方式.....	142		
10.5.2 散列文件的操作.....	142		
10.6 多关键字文件.....	143		
10.6.1 多关键字文件概念.....	143		
10.6.2 倒排文件.....	144		
		10.6.3 多重表文件.....	144
		习题.....	146
<b>第 11 章 数据结构综合应用</b> .....	147		
11.1 综合应用一：新生报到 信息注册系统设计.....	147		
11.1.1 案例需求分析.....	147		
11.1.2 案例知识目标.....	148		
11.1.3 案例核心算法及实现... ..	148		
11.1.4 其他参考代码.....	151		
11.2 综合应用二：万达停车场 管理系统设计.....	152		
11.2.1 案例需求分析.....	152		
11.2.2 案例知识目标.....	154		
11.2.3 案例核心算法及实现... ..	154		
11.2.4 其他参考代码.....	158		
11.3 综合应用三：最短时间 旅游路线查询系统设计.....	158		
11.3.1 案例需求分析.....	158		
11.3.2 案例知识目标.....	159		
11.3.3 案例核心算法及实现... ..	160		
11.3.4 其他参考代码.....	162		

# 第1章

## 绪论及 C 语言介绍

### 【学习目标】

1. 了解数据、数据元素、数据项等基本概念。
2. 掌握数据结构的概念和三大方面内容。
3. 了解学习数据结构的意义。
4. 掌握时间复杂度的求解方法。
5. 巩固和深入掌握与数据结构课程相关的 C 语言知识。

自从 1946 年美国宾夕法尼亚大学埃克特等人研制第一台计算机 ENIAC 开始, 计算机的应用已经深入到国民经济的各个领域, 人类已跨入信息化时代。与飞速发展的计算机硬件相比, 计算机软件的发展相对缓慢。因为软件的核心是算法, 所以对算法的深入研究必将促进计算机软件的发展。而算法实际上是加工数据的过程, 因此, 研究数据结构对设计高性能算法及高性能软件至关重要。同时软件应用于各个领域, 所以可以说数据结构是各个领域的桥梁。

### 1.1 基本概念与术语

为了深入学习和研究数据结构这门课程, 我们先来学习一下数据结构相关的概念和术语。下面介绍一些基本概念和常用术语。

数据: 能够被计算机识别、存储和加工处理的信息载体。例如, 当今计算机可以处理的图像、声音等。实例演示如图 1.1 所示。



图 1.1 数据的实例演示图



数据元素：数据的基本单位。在某些情况下，数据元素也称为元素、结点、顶点或者记录。数据元素要准确描述一个对象。数据元素有时可以由若干数据项组成，数据项是具有独立含义的最小标识单位。实例演示如图 1.2 所示。熊猫的数据元素（记录）是由若干数据项（名称、编号、类型、体重、特点）组成的。



图 1.2 数据元素的实例演示图

数据结构：数据之间的相互关系，即数据的组织形式。为了增加对数据结构的感性认识，下面举例来进一步说明有关数据结构的概念。

【例 1.1】学生成绩表，如表 1.1 所示。

表 1.1 学生成绩表

学号	姓名	C 语言	英语	数据库
31103001	孙彬彬	90	90	85
31103002	朱明秀	85	90	85
31103003	李芳	88	88	88
31103004	季红梅	90	80	85

我们把表 1.1 称为一个数据结构，表中的每一行是一个数据元素（或记录、结点），它由学号、姓名及各科成绩等数据项构成。

直接前趋：对表中任意一个结点，与它相邻且在它前面的结点称为该结点的直接前趋。第一条记录是第二条记录的直接前趋。

直接后继：对表中任意一个结点，与它相邻且在它后面的结点称为该结点的直接后继。第二条记录是第一条记录的直接后继。

(1) 数据结构一般包括以下 3 方面内容，如图 1.3 所示。

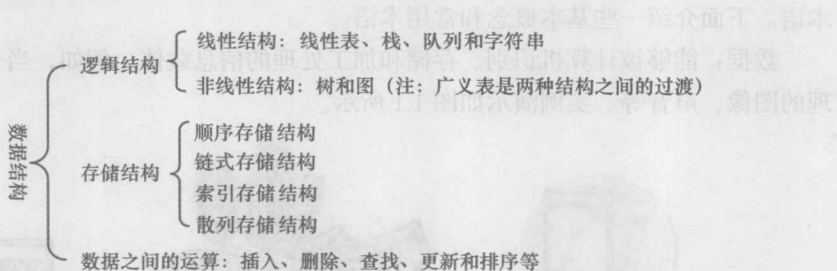


图 1.3 数据结构内容归纳

① 数据元素之间的逻辑关系，也称数据的逻辑结构。数据的逻辑结构是从逻辑关系上描述数

据,与数据的存储无关,是独立于计算机的。数据的逻辑结构可以看作是从具体问题抽象出来的数学模型。

② 数据元素及其关系在计算机存储器内的表示称与数据的存储结构。

③ 数据的运算,即对数据施加的操作。数据的运算定义在数据的逻辑结构上,每种逻辑结构都有一个运算的集合。最常用的检索、插入、删除、更新、排序等运算实际上只是在抽象的数据上所施加的一系列抽象的操作。

(2) 数据的逻辑结构分类。

① 线性结构:若结构是非空集,则有且仅有一个开始结点和一个终端结点,并且所有结点都最多只有一个直接前趋和一个直接后继。线性表、栈、队列、字符串等都是线性结构。

② 非线性结构:一个结点可能有多个直接前趋和直接后继。树和图等数据结构都是非线性结构。

(3) 数据的4种基本存储方法。

① 顺序存储方法:该方法把逻辑上相邻的结点存储在物理位置上相邻的存储单元里,结点间的逻辑关系由存储单元的邻接关系来体现。由此得到的存储表示称为顺序存储结构,通常借助程序语言的数组描述。该方法主要应用于线性的数据结构。非线性的数据结构也可通过某种线性化的方法实现顺序存储。

② 链接存储方法:该方法不要求逻辑上相邻的结点在物理位置上亦相邻,结点间的逻辑关系由附加的指针字段表示。由此得到的存储表示称为链式存储结构,通常借助于程序语言的指针类型描述。

③ 索引存储方法:该方法通常在储存结点信息的同时,还建立附加的索引表,索引表由若干索引项组成。若每个结点在索引表中都有一个索引项,则该索引表称为稠密索引。若一组结点在索引表中只对应一个索引项,则该索引表称为稀疏索引。索引项的一般形式是:(关键字、地址)。关键字是能唯一标识一个结点的一个或者多个数据项。稠密索引中索引项的地址指示结点所在的存储位置;稀疏索引中索引项的地址指示一组结点的起始存储位置。

④ 散列存储方法:根据结点关键字直接计算出该结点存储地址。

4种存储方法可单独使用,也可组合起来对数据结构进行存储映像。同一逻辑结构采用不同的存储方法,可以得到不同的存储结构。

(4) 数据结构3方面的关系:数据的逻辑结构、存储结构及数据的运算这3方面是一个整体。

## 1.2 学习数据结构的意义

数据结构是软件技术、网络技术及计算机应用技术等计算机学科各专业的一门重要的专业基础课程,数据结构是软件开发的基础、提高学生逻辑思维能力的核心、各工程领域的桥梁。著名的瑞士计算机科学家沃思(N.Wirth)教授曾提出:算法+数据结构=程序。这里的数据结构是指数据的逻辑结构和存储结构,而算法是对数据运算的描述。由此可见,程序设计的实质是对实际问题选择一种好的数据结构,加之设计一个好的算法,而好的算法在很大程度上取决于描述实际问题的数据结构。请看下面的两个例子。

**【例 1.2】**电话号码查询问题。

编写一个程序,查询某个城市的私人电话号码。假设有一个含有100万人的电话登记表,若

有该人的电话号码，则要迅速地找到其电话号码；否则指出该人没有电话号码。要写出好的查找算法，取决于这张登记表的结构及存储方式。可以采用表 1.2 和表 1.3 所示的存储结构。第一种结构是从头开始依次查对姓名，第二种是先通过索引表找到姓氏的地址，再开始依次查对姓名，可以看出第二种结构查找效率要高很多。

表 1.2 顺序存储结构

姓名	电话号码
王一	83808231
王二	83808232
⋮	⋮
张一	83811231
张二	83811111
⋮	⋮
张百	83822222

表 1.3 索引存储结构

姓氏	地址
王	—
张	—
⋮	⋮

姓名	电话号码
王一	83808231
王二	83808232
⋮	⋮
张一	83811231
张二	83811111

【例 1.3】教学计划编排问题。

大学完整的课程教学计划包含多门课程，在教学计划包含的许多课程之间，有些必须按规定的先后进行，有些则没有次序要求，如表 1.4 所示。表中各门课程之间的次序关系可用一个称为图的数据结构来表示，如图 1.4 所示，很容易看出课程之间的先后关系。

表 1.4 计算机软件专业的课程设置表

课程编号	课程名称	先修课程
C1	信息基础	无
C2	数据结构	C1、C4
C3	网页制作	C1
C4	C 语言程序设计	C1
C5	asp.net	C2、C3、C4
C6	JavaScript	C3
C7	数据库	C2、C9
C8	Java	C4
C9	软件工程	C2

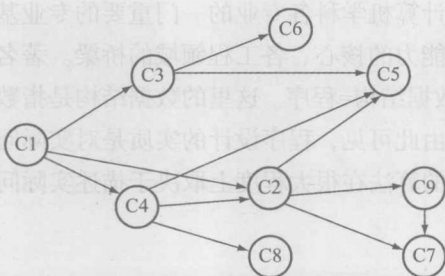


图 1.4 数据结构内容归纳

由以上 3 个例子可见，描述这类非数值计算问题的数学模型不再是数学方程，而是诸如表、树和图之类的数据结构。因此，可以说数据结构课程主要是研究非数值计算的程序设计中数据之间的逻辑关系和对数据的操作，以及如何将具有一定逻辑关系的数据存储到计算机内。与此同时，通过算法训练来提高学生的思维能力，通过程序设计的技能训练来促进学生的综合应用能力和专业素质的提高。

## 1.3 算法的描述和分析

### 1. 算法

非形式地说,算法是任意一个按定义的计算过程。它以零个或多个值作为输入,并产生一个或多个值作为输出。

(1) 一个算法可以被认为用来解决一个计算问题的工具。

(2) 一个算法是一系列将输入转换为输出的计算步骤。

### 2. 算法的描述

一个算法可以用自然语言、计算机程序语言或其他语言来说明,唯一的要求是该说明必须精确地描述计算过程。描述算法最合适的语言是介于自然语言和程序语言之间的伪语言。

### 3. 算法分析

(1) 评价算法好坏的标准:首先应该是“正确”的。此外,主要考虑以下3点:①执行算法所耗费的时间;②执行算法所耗费的存储空间,主要考虑辅助存储空间;③算法应易于理解,易于编码,易于调试等。

(2) 评价算法的效率包括时间复杂度和空间复杂度。一般情况下,算法的时间效率和空间效率是一对矛盾体。有时算法的时间效率高是以使用了更多的存储空间为代价的。有的时候又因减少存储空间,需要将数据压缩存储,从而会降低算法的时间效率。

### 4. 时间复杂度的求解方法

一段程序执行的时间是无法准确计算的,所以通常采用程序执行的次数来估算,用  $T(n)$  表示。使用大  $O$  记号表示的算法时间复杂度,称为算法的渐进时间复杂度。下面举例说明时间复杂度的求解方法的几种情况。

(1) 时间复杂度为  $O(1)$ 。

```
int i=3;           //执行1次
while(i<=99)      //执行34次
    i=i+3;         //执行33次
```

程序共执行 69 次,那么只要是执行的次数是常数的,那么  $T(n)=O(1)$ 。

(2) 时间复杂度为  $O(n)$ 。

```
int i,s=0;         //执行1次
for(i=0;i<n;i++)  //执行n+1次
    s=s+1;         //执行n次
printf("%d",s);   //执行1次
```

程序共执行  $2n+3$  次,那么只取最高级别的项(级别参照第(4)点),去掉该项的系数,那么  $T(n)=O(n)$ 。

(3) 时间复杂度为  $O(n^2)$ 。

```
int i,j,s=0;       //执行1次
for(i=0;i<n;i++)   //执行n+1次
    for(j=0;j<n;j++) //执行n(n+1)次
        s=s+1;     //执行n2次
```

程序共执行  $2n^2+2n+2$  次,那么只取最高级别的项,去掉该项的系数,那么  $T(n)=O(n^2)$ 。

(4) 常见时间复杂度, 按照数量级别递增排列, 依次为: 常数阶  $O(1)$ 、对数阶  $O(\log_2 n)$ 、线性阶  $O(n)$ 、线性对数阶  $O(n\log_2 n)$ 、平方阶  $O(n^2)$ 、立方阶  $O(n^3)$ …… $k$  次方阶  $O(n^k)$ 、指数阶  $O(2^n)$ 。

## 1.4 C 语言相关知识介绍

### 1. 程序设计 3 种结构

(1) 顺序结构: 按语句在源程序中出现的次序依次执行。

**【例 1.4】** 输入两个整数, 进行交换后输出 (数据交换流程图见图 1.5)。

```
#include "stdio.h"
main()
{ int a,b,t;
scanf("%d,%d",&a,&b);
t=a;
a=b;
b=t;
printf("a=%d,b=%d",a,b);
}
```

(2) 选择结构: 根据一定的条件有选择地执行或不执行某些语句。

if 语句也称为条件语句, 它根据一个条件的真和假有选择地执行或不执行某个语句。

if 语句有以下两种形式。

① if 格式 (流程图见图 1.6):

if (表达式) 语句 1;

② if-else 格式 (流程图见图 1.7):

if (表达式) 语句 1;

else 语句 2;

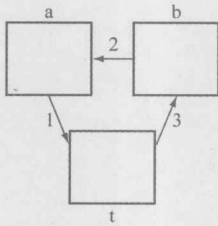


图 1.5 数据交换流程图

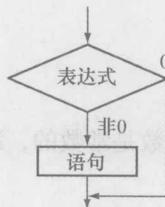


图 1.6 if 格式流程图

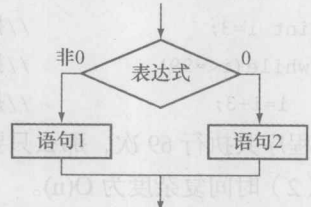


图 1.7 if-else 格式流程图

**【例 1.5】** 求一元二次方程  $ax^2+bx+c=0$  的根。

```
#include "stdio.h"
#include "math.h"
main()
{ double a,b,c,t,x1,x2;
scanf("%f,%f,%f",&a,&b,&c);
t=b*b-4*a*c;
if(t>=0)
{ x1=(-b+sqrt(t))/(2*a);
x2=(-b-sqrt(t))/(2*a);
printf("%f,%f",x1,x2);
}
else
```

```
printf("无根");
}
```

(3) 循环结构：在一定条件下重复执行相同的语句。

while 语句（流程图见图 1.8）的一般形式：

```
while (表达式)
    语句;
```

【例 1.6】求  $3+6+9+\dots+99$  的和。

```
#include "stdio.h"
main()
{ int i,sum=0;
  i=3;
  while(i<=99)
  { sum=sum+i;
    i=i+3;
  }
  printf("sum=%d",sum);
}
```

for 语句（流程图见图 1.9）的一般形式：

```
for(表达式1;表达式2;表达式3)
    语句;
```

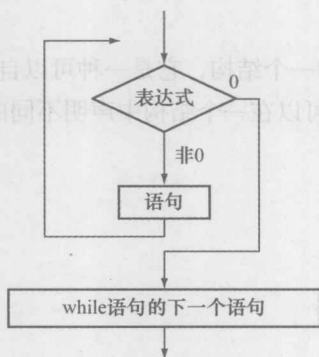


图 1.8 while 语句流程图

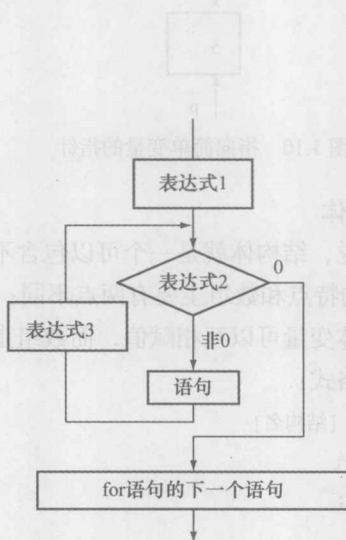


图 1.9 for 语句流程图

【例 1.7】求  $3+6+9+\dots+99$  的和。

```
#include "stdio.h"
main()
{ int i,sum=0;
  for(i=3;i<=99;i=i+3)
    sum=sum+i;
  printf("sum=%d",sum);
}
```

## 2. 指针

程序中任何变量都占据一定数目的存储单元，所需存储单元的数目由变量的类型决定。变量所占据的存储单元的首地址就是变量的地址，变量的地址表示为： $\&$ 变量名。存放变量的地址的变量就是指针变量。

**【例 1.8】** 指向简单变量的指针 (见图 1.10)。

```
main()
{ int x;
  int *p;
  x=5;
  p=&x;
  printf("x=%d,*p=%d",x,*p);
}
```

程序的结果为: x=5,\*p=5

**【例 1.9】** 指向一维数组的指针 (见图 1.11)。

```
main()
{ int a[5]={1,2,3,4,5};
  int *p;
  p=a;//不需要加&
  printf("a[2]=%d,* (p+2)=%d",p[2],*(p+2));
}
```

程序的结果为: a[2]=3, \*(p+2)=3

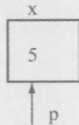


图 1.10 指向简单变量的指针

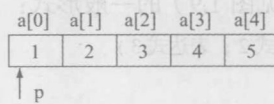


图 1.11 指向一维数组的指针

**3. 结构体**

简单来说, 结构体就是一个可以包含不同数据类型的一个结构, 它是一种可以自己定义的数据类型。它的特点和数组主要有两点不同: 首先, 结构体可以在一个结构中声明不同的数据类型; 其次, 结构体变量可以互相赋值, 而数组是做不到的。

结构体格式:

```
struct [结构名]
{ 成员 1;
  成员 2;
  ...
  成员 n;
};
```

**【例 1.10】** 定义学生数据类型 (见图 1.12), 并通过指针赋值后输出。

```
#include "stdio.h"
typedef struct student
{
  char name[10]; //姓名
  char sex; //性别
  int age; //年龄
  float score; //数据结构成绩
}stu;
main()
{
  stu t;
  stu *p;
```

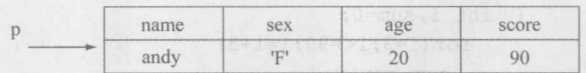


图 1.12 定义学生数据类型

```

p=&t;
p->sex='F';
p->age=20;
printf("sex=%c,age=%d",t.sex,t.age);
}

```

程序的结果为: sex=F, age=20

## 知识实践一 学生管理系统登录模块设计

### 1. 知识实践目的

- (1) 掌握程序设计的三大结构: 顺序结构、选择结构、循环结构。
- (2) 掌握结构体的定义和应用。
- (3) 掌握指针的定义和应用。
- (4) 掌握函数的编写和调用方法。

### 2. 知识实践内容

利用结构体和指针实现学生管理系统登录模块设计, 要求先对多个用户信息进行初始化, 用户信息包括账号和密码两部分, 然后输入当前用户的账号和密码进行验证, 正确则显示“登录成功!”, 错误则显示“账号或者密码错误!”。

### 3. 代码清单

```

#include "stdio.h"
#include "string.h"
//用户结构体定义
typedef struct
{
    long id;
    char pwd[6];
}user;
user users[50]; //数组定义, 用来存储多个用户信息
void init(int n) //输入用户信息
{
    user *p;
    int i;
    p=users; //指针的应用
    printf("请管理员输入用户信息:\n");
    for(i=0;i<n;i++)
    {
        scanf("%ld,%s",&p[i].id,p[i].pwd);
    }
}
main()
{
    long stuid;
    char stupwd[6];
    int i;
    int n;
    printf("请输入用户数量: ");
    scanf("%d",&n);
    init(n); //调用输入用户信息函数
}

```



```

printf("请您输入账号和密码:");
scanf("%ld,%s",&stuid,stupwd);
for(i=0;i<n;i++)
{
    if(stuid==users[i].id && strcmp(stupwd,users[i].pwd)==0)//验证
    {
        printf("登录成功!");
        break;
    }
}
if(i==5)
    printf("账号或者密码错误!");
}

```

#### 4. 结果验证

上机实验, 结果如图 1.13 所示。

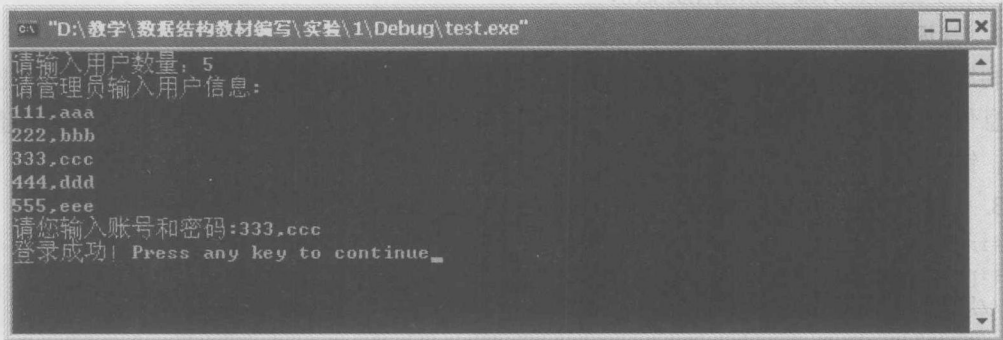


图 1.13 实验结果

## 习 题

### 一、填空题

1. 数据结构研究的 3 大方面内容是\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
2. 数据的逻辑结构包括\_\_\_\_\_和\_\_\_\_\_两大类。
3. 数据的存储结构分为顺序存储结构、\_\_\_\_\_、索引存储结构和\_\_\_\_\_。

### 二、选择题

1. 以下关于数据的逻辑结构的叙述正确的是 ( )。
  - A. 数据的逻辑结构是数据间关系的描述
  - B. 数据的逻辑结构反映了数据在计算机中的存储方式
  - C. 数据的逻辑结构分为顺序结构和链式结构
  - D. 数据的逻辑结构分为静态结构和动态结构
2. 以下术语中与数据的存储结构无关的是 ( )。
  - A. 顺序表
  - B. 链表
  - C. 散列表
  - D. 队列
3. 下列算法的时间复杂度是 ( )。

```
for(i=1;i<=n;i+ +)
```