高等学校计算机规划教材

程序设计实训教程

主　编　李　敏　薛冰冰
副主编　俞卫华　高艳平　万志伟

電子工業出版社
Publishing House of Electronics Industry
北京・BEIJING

内 容 简 介

本书是河南省省级精品课程《C 语言程序设计》（ISBN 978-7-121-18839-8）的配套教材，全书的内容从软件设计基础讲起，通过实例，详细讲述了综合程序设计的过程，以及综合程序设计实训的内容，最后，介绍了综合程序设计报告的书写规范。本书结构清晰、通俗易懂，综合性例题典型丰富，注重读者进行综合程序设计方法的训练，注重培养读者编写、调试大型程序的能力及良好的程序设计风格。

本书既可作为高等院校各专业《C 语言程序设计》课程的实训教材，也可作为计算机程序设计人员的参考书。

图书在版编目（CIP）数据

程序设计实训教程 / 李敏，薛冰冰主编．—北京：电子工业出版社，2015.5

ISBN 978-7-121-24715-6

Ⅰ. ①程… Ⅱ. ①李… ②薛… 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（2014）第 260380 号

策划编辑：袁　玺
责任编辑：郝黎明
印　　刷：
装　　订：
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：787×1 092　1/16　印张：13　字数：332.8 千字
版　　次：2015 年 5 月第 1 版
印　　次：2015 年 5 月第 1 次印刷
定　　价：32.00 元

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 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前言

PREFACE

学好程序设计语言，不仅要掌握基本的语法规则和基本的语句，以及基本的编程算法，更重要的是要针对实际问题进行综合程序设计的训练，通过大量的实训，规范综合程序设计的方法，积累编程的经验，提高程序设计的能力，为进行大型的软件设计打下良好的基础。

该教材由长期从事一线教学的优秀教师编写，凝聚了教师多年来讲授程序设计课程的宝贵经验，引入软件工程的方法，注重实训例题和习题与相关专业相结合，各部分内容均以训练学生实际编程能力、分析解决问题能力为出发点，注重学生计算机综合程序设计应用能力的培养和综合素质的提高。突出实际应用，强调理论联系实际，提高学生软件设计能力，体现规范的软件设计理念。

全书共分十二个教学实训项目，主要综合了 C 语言程序设计的基本结构、函数、数组、指针、结构体、链表和文件等内容，在编写每个实训案例时，都针对案例进行问题分析、总体设计、详细设计、代码编写、程序调试、程序文档制作六个阶段的训练，并在代码中给出大量的注释，系统地训练了综合程序设计的能力，掌握结构化程序设计的基本方法、程序设计的基本开发过程、调试和测试程序的基本技巧。

本书编者具有多年的 C 语言程序设计和相关专业课程的教学经验。本书由李敏、薛冰冰担任主编，俞卫华、高艳平、万志伟担任副主编。具体分工如下：李敏编写了第 1 章；高艳平编写了第 2 章；俞卫华编写了第 3 章；薛冰冰编写了第 4 章和附录 A、B；万志伟参加了部分程序的调试工作。在本书的编写过程中，参阅并引用了国内外诸多同行的著作，并在互联网上摘录了部分例题，在此向他们致意。

由于作者学术水平有限，书中错误和不妥之处在所难免，敬请读者批评指正，在此表示由衷的感谢。

目录

CONTENTS

第 1 章　软件设计基础

软件是计算机程序、规程以及运行程序所需的相关文档和数据。例如，开发一个学生信息管理系统，就要对学生的基本信息、班级信息、课程信息、成绩信息等进行管理，通过该系统，可以做到信息的规范管理、科学的统计和快速的查询，从而减少管理方面的工作量，提高信息管理工作的效率。开发这样一个系统，就像盖一座大楼，是一项工程，这就是软件工程技术研究的内容，几万个学生的各种信息要存储在计算机中，这些信息不是杂乱无章的，而是按照一定的原则来组织和存储的，那么就需要数据库技术和软件设计。计算机软件作为一种逻辑系统，它和计算机硬件有着显著的差别，它主要是对软件进行定义、开发和维护。

本章主要介绍软件的基本概念、软件开发的过程、软件的详细设计，而软件工程是计算机专业的一门专业课程，要提高软件的开发能力还需要进一步学习相关课程，这里仅仅简单介绍软件设计的一些基本概念。

1.1　软件概述

软件的规模大小、复杂程度决定了软件开发的难度。对一个软件而言，它的程序复杂性将随着程序规模的增加而呈指数级上升趋势。因此，必须采用科学的软件开发方法，采用抽象、分解等科学方法降低复杂度，以工程的方法管理和控制软件开发的各个阶段，以保证大型软件系统的开发具有正确性、易维护性、可读性和可重用性。

1.1.1　软件定义

软件是程序、数据及相关文档的集合。其中，程序是软件开发人员根据用户需求开发的、用程序设计语言描述的、适合计算机执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护及使用密切相关的图文资料的总体。

1.1.2　软件发展

软件的发展大致可划分为以下四个阶段。

1. 程序设计阶段（1950 ~ 1965 年）

这个阶段是计算机软件发展的早期阶段，软件开发采用低级语言，开发效率比较低，应用领域基本局限于科学和工程的数值计算，人们不重视软件文档的编制，只注重代码的编写。计算机编程很简单，没有系统化的方法，在硬件的生产已经开始趋于标准化的时候，软件的生产仍是个体化，软件产品处在初级阶段，大多数软件是由使用者自己开发的。

2. 程序系统阶段（1965 ~ 1974 年）

这个阶段是计算机软件发展的第二阶段，相继诞生了大量的高级程序设计语言，程序

开发的效率明显提高，实时系统和第一代数据库管理系统也相继出现，并产生了成熟的操作系统和数据库管理系统。到后期，由于软件规模不断扩大，复杂程度大幅度提高，许多程序的个人化特性使得它们根本不能维护，在软件维护上所花费的精力和消耗资源的速度是惊人的，这就产生了“软件危机”，同时出现了有针对性地进行软件开发方法的理论研究和实践。

3. 软件工程阶段——结构化方法（1974～1989 年）

这个阶段是计算机软件发展的第三阶段，在这一阶段，软件规模越来越大，结构越来越复杂，软件开发管理既困难又复杂，软件开发费用不断增加。软件的开发技术落后，生产方式落后，仍采用手工方式，开发工具也落后，生产效率低，这就是软件危机产生的原因。

在这一阶段，以软件的产品化、系列化、工程化、标准化为特征的软件产业发展起来，消除了软件生产的个体化特征，有了可以遵循的软件工程化的设计原则、方法和标准，并且结构化方法也得到发展。

4. 软件工程阶段——面向对象方法（1989～至今）

这个阶段是计算机软件发展的第四阶段，在这一阶段，已经不再着重于单台计算机和计算机程序，而是面向计算机和软件的综合影响。由复杂的操作系统控制的强大的桌面机、广域网络和局域网络，配以先进的软件应用已成为标准。计算机体系结构迅速地从集中的主机环境转变为分布的客户机/服务器环境。世界范围的信息网提供了一个基本结构，信息高速公路和网际空间连通已成为令人关注的热点问题。事实上，Internet 可以看做能够被单个用户访问的软件。

计算机发展正朝着社会信息化和软件产业化方向发展，由于软件编程方法及软件设计思想不断更新，导致软件工程进入了面向对象方法的时代，出现了占据主导地位的面向对象技术，它将在许多领域中迅速取代传统的软件开发方法。同时，软件开发技术继续发展，并逐步转向智能化、自动化、集成化、并行化和工程化。另外，计算机网络技术、分布式技术对软件工程的发展也起到了促进作用，使得当前采用面向对象技术开发的软件系统越来越多。

1.1.3 软件特点

1. 工具性特征

计算机软件包括程序和文档两个部分。计算机程序包括源程序和目标程序，源程序是用计算机高级语言（如 BASIC、Algol、C、C++等）编写的程序，表现为数字、文字和符号的组合，构成符号化指令序列或符号化语句序列；目标程序是用机器语言编写的，体现为电脉冲序列的一串二进制数(0 和 1)指令编码，直接用于驱动计算机硬件工作，保证计算机系统发挥各项功能，获得一定结果，因而又具有工具性特征。软件在调入计算机运行之前，首先表现为作品性，人们无法通过“阅读”或“欣赏”计算机程序与文档而制造任何有形产品和实现任何操作。但当软件调入计算机运行时，则更主要地表现为工具性，即通过控制计算机硬件动作过程，获得某种结果。

2. 软件开发工作量大、成本高，但复制容易且成本低

软件开发必须经过功能限定、逻辑设计和编码三个步骤，要求软件开发人员必须具有丰富而超前的专业和相关知识，极强的逻辑和形象思维能力，了解计算机硬件的最新发展状况与发展前景，熟练掌握和使用编程语言。开发具有实用商业价值的计算机软件，通常需要按照专业化分工、流水线作业的方式由一批人共同完成。可见，开发计算机软件必须具备相应的物质和技术条件，有充足的开发资金和良好的开发环境。复制是对计算机软件的客观再现，不改变软件内容和本身的价值，复制后的软件以一定的客观物质形式体现，具有可感知性。计算机软件的可复制性决定了其可以广泛传播和有效利用，创造经济和社会效益。

3. 软件具有无形性，可以多次使用

计算机软件是智力劳动产生的精神产品，如计算机程序、说明程序的文档等都是智力劳动的直接产物，不具有任何形状，人们只有借助于一定的物质载体和工具才能感知其存在。计算机软件在同一时间可以为若干人分别使用，软件只要不受操作失误、计算机病毒等影响，就可以无限制反复使用。但是，计算机软件又具有工具性，主要通过“使用”而发挥其功用，因而应该具有使用寿命，使用寿命在流通领域表现为商业寿命。在科学技术飞速发展、新软件层出不穷的今天，计算机软件的商业寿命正在日益缩短。

与硬件相比，软件的特点如下。

① 表现形式不同：软件是逻辑部件，具有很高抽象性，缺乏可见性；硬件是物理部件，看得见、摸得着。

② 生产方式不同：软件的开发，是人的智力的高度发挥，不是传统意义上的硬件制造，软件的成本主要在开发和研制。开发和研制后，通过复制可大量生产。

③ 要求不同：硬件产品允许有误差，而软件产品不允许有误差。

④ 维护不同：由于磨损和老化，硬件会用旧用坏，解决的办法是更换一个相同的备件。而在理论上，软件不会用旧用坏，但软件是有生命周期的，存在退化现象，软件维护要比硬件复杂得多。

1.1.4 软件危机

20 世纪 60 年代初，美国的专业软件公司只有十几家，1968 年发展到 1300 多家。这些公司研制、生产和销售各种应用软件。尽管软件业发展迅速，但随着计算机应用范围的扩大，人们对软件的需求越来越大，软件的规模也越来越大，结构越来越复杂。例如，IBM 公司 60 年代研制的 IBM 360 操作系统，参与开发的单位美国有 11 个、欧洲有 6 个，参与开发的软件工作人员有 700 余人，其他辅助人员 1000 多人，从 1963 到 1966 年历时 4 年，花费 5 亿美元开发的系统仍包含了大量的错误，以后通过不断地被修改、补充，但每个版本还是有上千种错误。因此，在 20 世纪 60 年代硬件迅速发展的同时，软件的发展遇到越来越大的困难，人们称这一现象为“软件危机”。软件危机主要表现在：对软件开发成本和进度的估计常常很不准确，经费预算经常突破，完成时间一再拖延；开发的软件不能满足用户要求，用户对软件不满意的现象经常发生；开发的软件可维护性差、可靠性差。产生

软件危机的原因主要：软件规模越来越大，结构越来越复杂；软件开发管理困难而复杂；软件开发费用不断增加；软件开发技术落后；生产方式落后，即采用手工方式；开发工具落后，生产效率低。核心原因是软件系统的复杂度远大于硬件，计算机硬件产品的制造已经标准化、工程化、产业化，但软件生产离此目标相距甚远。

从软件危机的种种表现和软件作为逻辑产品的特殊性可以总结出软件危机产生的原因。

1. 用户需求不明确

在软件开发过程中，用户需求不明确问题主要体现在以下四个方面。

① 在软件开发之前，用户也不清楚软件的具体需求。

② 用户对软件需求的描述不精确，可能有遗漏、有二义性，甚至有错误。

③ 在软件开发过程中，用户又提出修改软件功能、界面、支撑环境等方面的要求。

④ 软件开发人员对用户需求的理解与用户本来的愿望有差异。

2. 缺乏正确的理论指导

第二个原因是缺乏有力的方法学和工具方面的支持。由于软件不同于大多数其他工业产品，其开发过程是复杂的逻辑思维过程，其产品极大程度地依赖于开发人员的智力投入。由于过分地依靠程序设计人员在软件开发过程中的技巧和创造性，加剧了软件产品的个性化，这也是发生软件危机的一个重要原因。

3. 软件规模越来越大

随着软件应用范围的增大，软件规模愈来愈大。大型软件项目需要组织一定的人力共同完成，而多数管理人员缺乏开发大型软件系统的经验，多数软件开发人员又缺乏管理方面的经验。各类人员的信息交流不及时、不准确，有时还会产生误解。软件项目开发人员不能有效地、独立自主地处理大型软件的全部关系和各个分支，因此容易产生疏漏和错误。

4. 软件复杂度越来越高

软件开发不仅在规模上快速发展扩大，其复杂性也在急剧增加。软件开发产品的特殊性和人类智力的局限性，导致人们无力处理“复杂问题”。所谓“复杂问题”是相对的，一旦人们采用先进的组织形式、开发方法和工具提高了软件开发效率和能力，新的、更大的、更复杂的问题又会出现。

针对上述情况，1968 年北大西洋公约组织在德国召集了 50 名一流的编程人员、计算机科学家和工业界巨头，制定摆脱软件危机的办法，首次提出了软件工程（Software Engineering）这一概念，倡导以工程的原理、原则和方法进行软件开发，以期解决“软件危机”问题。软件工程是当时的一门新兴的、指导计算机软件开发和维护的工程学科。被定义为“运用系统的、规范的和可定量的方法来开发、运行和维护软件。”

软件工程的主要研究对象是大型软件。软件工程研究的内容主要包括：软件质量保证和质量评价；软件研制和维护的方法、工具、文档；用户界面的设计以及软件管理等。软件工程的最终目的是摆脱手工生产软件的状况，逐步实现软件研制和维护的规范化。

1.1.5 软件生命周期

1. 软件工程过程

软件工程过程是把用户的需求转化为软件产品的一系列活动。为获得软件产品，在软件工具的支持下，由软件开发人员完成的一系列工程活动。每个软件开发机构都可以规定自己的软件工程过程，针对不同类型的软件产品，软件开发机构也可能使用多种软件工程过程。但不管哪一种情况，软件工程过程一般包括四种基本的过程活动。这些活动包括计划（Plan）、开发（Do）、确认（Check）、维护（Action）。实际上，软件工程过程是一个软件开发机构针对某一类软件产品为自己规定的工作步骤。

2. 软件生命周期

一个软件从提出开发、实现、使用、维护直到停止使用的过程称为软件生命周期。软件生命周期一般包括：可行性分析和项目开发计划、需求分析、概要设计、详细设计、编码实现、测试、交付使用及维护等具体环节。大体可分为三个时期：计划阶段（问题定义和可行性分析、需求分析），开发阶段（软件设计、编码、测试）和运行阶段（软件维护）。各阶段的任务及产生的相应文档如表 1-1 所示。

表 1-1 软件生命周期各阶段的任务

时　期	阶　段	任　务	文　档
软件计划	问题定义	理解用户要求，划清工作范围	计划任务书
	可行性分析	可行性方案及代价	
	需求分析	软件系统的目标及应完成的工作	需求规格说明书
软件开发	概要设计	系统的逻辑设计	软件概要设计说明书
	详细设计	系统模块设计	软件详细设计说明书
	软件编码	编写程序代码	程序、数据、详细注释
	软件测试	单元测试、综合测试	测试后的软件、测试大纲、测试方案与结果
软件运行	软件维护	运行和维护	维护后的软件

（1）软件计划

问题定义阶段即进行调研和分析，弄清用户想干什么，不想干什么，以确定工作范围。通过调查抽象出“用户想要解决的问题是什么”。

在上述工作的基础上进行可行性分析，本阶段的具体工作如下：分析所需研制的软件系统是否具备必要的资源，技术上、经济上的可能性和社会因素的影响，回答“用户要解决的问题能否解决”，即确定项目的可行性。

需求分析要解决“做什么的问题”。经过问题定义，可行性分析后，需求分析阶段要考虑所有的细节问题，以确定最终的目标系统做哪些工作，形成目标系统完整的准确要求。

该阶段最后提交说明系统目标及对系统要求的规格说明书。

（2）软件开发

软件开发包括概要设计、详细设计、软件编码和软件测试四个阶段。

概要设计又称为总体设计、逻辑设计。该阶段要回答“怎样实现目标系统”的问题。首先应考虑实现目标系统的可能方案，并选择一个最佳方案。确定方案后应完成系统的总体设计，即确定系统的模块结构，给出模块的相互调用关系，并产生概要设计说明书。

详细设计阶段回答“应该怎样具体实现目标系统”的问题。在概要设计的基础上，要给出模块的功能说明和实现细节，包括模块的数据结构和所需的算法，最后产生详细设计说明书。

详细设计完成后进入软件编码阶段，程序员根据系统的要求和开发环境，选用合适的高级程序设计语言或部分选用汇编程序设计语言编写程序代码。

软件测试分为单元测试和综合测试两个阶段。单元测试指对每一个编制好的模块进行测试，发现和排除程序中的错误。综合测试指通过各种类型的测试检查软件是否达到预期的要求。

（3）软件维护

软件维护阶段是长期的过程，是在软件投入使用以后的时期，因为发生经过测试的软件可能还有问题；或者用户的要求还会发生变化；或者软件运行的环境可能发生变化，在上述情况发生时，都要进行软件的维护。因此，交付使用的软件仍然需要继续排错、修改和扩充，这就是软件维护。

3. 软件生命周期模型

软件生命周期模型（Life-Cycle Mode）也称软件过程模型，是软件系统开发项目总貌的一种描述，着眼于对项目管理的控制和逐步逼近的策略。

（1）瀑布模型

瀑布模型（Waterfall Mode）是 1976 年由 B. W. Boehm 提出的传统的软件生命周期模型，如图 1-1 所示。

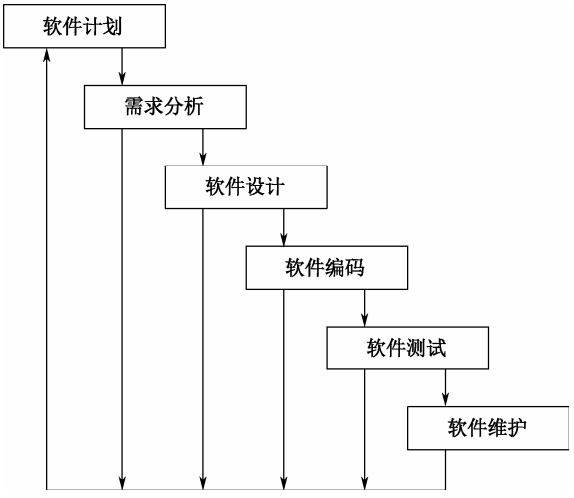


图 1-1　瀑布模型

由图可见，该模型将软件开发过程划分为六个阶段，这六个阶段是顺序进行的，前一阶段的工作完成后，下一阶段的工作才能开始；前一阶段产生的文档是下一阶段工作的依

据。该模型适合在软件需求比较明确，开发技术比较成熟的场合下使用，它是软件工程中应用最广泛的模型。

（2）快速原型法模型

快速原型法模型（Rapid Prototyping）是针对瀑布模型中的缺点提出的一种改进模型，如图 1-2 所示。

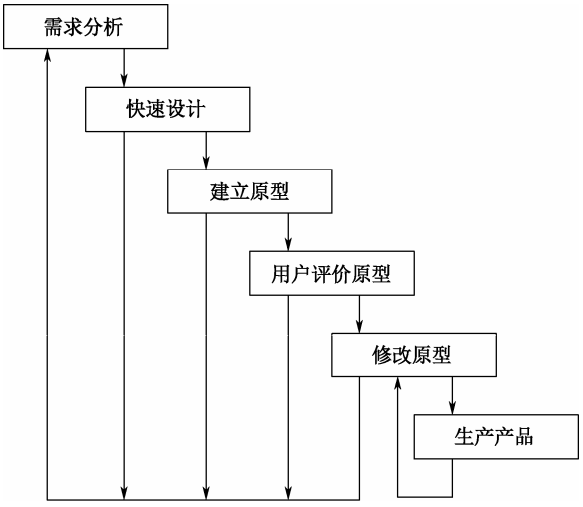


图 1-2　快速原型法模型

快速原型法也从了解需求开始，开发人员和用户一起来定义所有目标，确定哪些需求已经清楚，哪些还需要进一步定义；然后是快速设计，快速设计主要集中在用户能看得见的一些软件表示方面（如输入方法、输出形式等）。快速设计可产生一个原型（试验性产品）。用户有了原型，即可对其评价然后修改要求。重复上述各步骤，直到该原型能满足用户的需求为止。

软件生命周期模型还包括许多其他模型，如螺旋模型（The Spiral Model）、四代技术（Fourth-Generation Techniques，4GT）、面向对象生存期模型（Object-Oriented Life-Cycle Model）等。

近年来，面向对象（Object-Oriented，OO）方法日益受到人们的重视。面向对象方法遵循人类习惯的思维方式，开发出的软件（产品）的稳定性、可复用性和可维护性等都比传统的开发方法好。

目前，经过多年理论完善和实践，传统的瀑布模型已形成了一个较完整的体系，仍是软件开发中使用的最基本的理论基础和技术手段。

1.1.6　软件开发过程

软件开发过程是指软件工程人员为了获得软件产品，在软件工具支持下实施的一系列软件工程活动。科学、有效的软件开发过程应该定义一组适合所承担的项目特点的任务集合。

软件开发过程一般分为如下五个阶段。

1. 问题的定义及规划

此阶段是软件开发与需求者共同讨论的，主要确定软件的开发目标及其可行性。

2. 需求分析

在确定软件开发可行性的情况下，对软件需要实现的各个功能进行详细需求分析。需求分析是一个很重要的阶段，将这一阶段做好，会为整个软件项目的开发打下良好的基础。“唯一不变的是变化本身”，同样，软件需求也是在软件开发过程中不断变化和深入的，因此，必须定制需求变更计划来应付这种变化，以保护整个项目的正常进行。

3. 软件设计

此阶段中要根据需求分析的结果，对整个软件系统进行设计，如系统框架设计、数据库设计等。软件设计一般分为总体设计和详细设计。好的软件设计将为软件程序的代码编写打下良好的基础。

4. 程序编码

此阶段是将软件设计的结果转化为计算机可运行的程序代码。在程序编码中必须制定统一、符合标准的编写规范，以保证程序的可读性、易维护性，提高程序的运行效率。

5. 软件测试

在软件设计完成之后要进行严密的测试，发现软件在整个软件设计过程中存在的问题并加以纠正。整个测试阶段分为单元测试、组装测试、系统测试三个阶段进行。

测试方法主要有白盒测试和黑盒测试。白盒测试时将程序看做一个透明的盒子，即测试人员完全了解程序的内部结构和处理过程。所以测试时按照程序内部的逻辑测试程序，检验程序中的每条通路是否都能按预定的要求正确工作。黑盒测试时完全不考虑程序内部的结构和处理过程，只按照规格说明书的规定来检查程序是否符合要求，黑盒测试只测试功能。

1.1.7 软件开发工具

软件开发工具是为支持软件人员开发和维护活动而使用的软件。它可以帮助开发人员完成一些烦琐的程序编制和调试问题，使软件开发人员将更多的精力和时间投到最重要的软件需求和设计上，提高软件开发的速度和质量。

软件开发工具主要是语言开发工具如，C++、Java、Delphi 开发工具等。软件开发工具的功能如下。

1）认识与描述客观系统。

2）存储及管理开发过程中的信息。

3）代码的编写与生成。

4）文档的编制或生成。

5）软件项目的管理。

软件开发工具的特性如下。

1）表达能力或描述能力。

2）保持信息一致性的能力。
3）使用的方便程度。
4）工具的可靠性。

1.2 软件的详细设计

1.2.1 基本概念

软件设计是一个把软件需求转化为软件表示的过程，即把分析结果加工为在程序细节上接近于源程序的软件表示（软件描述）。软件设计的目标是对将要实现的软件系统体系结构、系统的数据、系统模块间的接口，以及所采用的算法给出详尽的描述。

软件设计阶段通常分为两步：一是系统的总体设计或概要设计，它的任务是确定软件系统结构；二是系统的详细设计，即进行各模块内部的具体设计。软件设计方法有多种，如面向数据流分析（Data Flow Analysis, DFA）的设计，也称为结构化设计（Structured Design, SD），还有面向数据结构的设计，如 Jackson 系统开发（Jackson System Development，JSD）方法和逻辑构造程序（Logically Constructed Program）方法。本节重点采用结构化设计方法来介绍概要设计和详细设计。

结构化设计方法是计算学科的一种典型的系统开发方法，也是被广泛使用的一种传统的软件开发方法。它的基本思想如下：采用自顶向下的模块化设计方法，按照模块化原则和软件设计策略，将需求分析得到的数据流图，映射成由相对独立、单一功能的模块组成的软件结构。

用结构化方法开发软件的过程如下：从系统需求分析开始，运用结构化分析方法建立环境模型（即用户要解决的问题是什么，以及要达到的目标、功能和环境）；需求分析完成后采用结构化设计方法进行系统设计，确定系统的功能模型；进入软件开发的实现阶段，运用结构化程序设计方法确定用户的实现模型，完成系统的编码和调试工作。

结构化方法是由结构化程序设计语言发展而来的。早期的计算机程序设计都是手工式的设计方法，20 世纪 60 年代软件危机的出现促使人们开始对程序设计方法进行研究，经过多年的研究与实践，逐步形成了结构化程序设计的方法。结构化程序设计就是选用最佳的程序设计语言实现编码，其目的是要使程序具有一个合理的结果，以保证程序的正确性。

1. 概要设计

概要设计也称总体设计，任务是确定软件结构。采用结构化设计方法来设计结构，其目标是根据系统的需求，分析资料，确定软件应由哪些系统或模块组成，它们采用什么方式连接，接口如何，才能构成一个好的软件结构，如何用恰当的方法把设计结果表达出来。同时，要考虑数据库的逻辑设计。

概要设计是根据系统分解与抽象的原则，自顶向下、逐步求精地完成系统的开发过程，用模块结构图来表示程序模块之间的关系。模块化是把程序划分成独立命名的、独立访问的模块。每个模块完成一个子功能，将这些模块组合起来就构成了一个整体，可以完成指

定的功能。由于模块之间是相对独立的，所以每个模块可以独立地被理解、编程、调试，模块的相对独立性能有效防止了错误在模块之间蔓延，提高了系统的可靠性，从而使大型信息系统的开发工作得以简化，缩短了软件开发周期。

2. 详细设计

详细设计的主要任务是对概要设计所得的对象类的表达做进一步的细化分析、设计和验证，为软件结构图中的每一个模块确定实现算法和局部数据结构，并用某种工具描述出来。严格地讲，详细设计应当把每个模块用到的每个函数，每个函数的每个参数的定义都精细地提供出来。一个好的详细设计，可以使程序编码的复杂性大大降低。

实际上，从需求分析到概要设计再到详细设计，一个软件项目才完成了一半，换言之，一个大型软件系统在完成了一半的时候，其实还没有开始代码的编写工作。

详细设计中可以采用逐步求精的设计方法，应采用自顶向下的策略，即在模块化分解过程中，问题一步一步细化。由于人的理解能力、记忆力有限，一个复杂问题不可能触及问题的所有方面和全部细节，为将复杂性降低到人们可以掌握的程度，常将其拆成若干个小问题再分别解决。或者说，把要解决的复杂问题分解成若干个子问题，然后分别独立解决这些子问题，再将各个子问题的解以某种方式连接起来，这就是原始问题的正确解答。如果子问题仍较复杂，则又可以把这些子问题看做新的要解决的问题，而对它们继续进行分解。这样不断地分解，最终使得子问题简单到可用若干行程序设计语言来描述，那么整个问题就解决了。这样研制出来的程序具有结构清晰的特点，有较强的可读性和可维护性。

Wirth 曾对逐步细化的方法作过如下说明：“我们对付复杂问题的最重要的办法是抽象，因此，对一个复杂的问题不应该立刻用计算机指令、数学和逻辑符号来表示，而应该用较自然的抽象语句来表示，从而得出抽象程序。抽象程序对抽象的数据进行某些特定的运算并用某些合适的符号（可能是自然语言）来表示。对抽象程序做进一步的分解，并进入下一个抽象层次。这样的精细化过程一直进行下去，直到程序能被计算机接受为止。这样的程序可能是用某种高级语言或机器指令书写的。”

详细设计之后就是按照一定的原则编制正确易懂的程序，在这一阶段，选用合适的程序设计语言、良好的编码风格，对保证程序的可读性、可靠性、可测试性和可维护性将产生重要的作用。

软件详细设计的描述工具可分为图形、表格和语言三类。下面介绍常用的详细设计描述工具。

1.2.2 传统流程图

传统流程图用于描述程序的控制流程，其基本描述符号如图 1-3 所示。

传统流程图的优点是直观，便于初学者掌握；缺点是控制流不带任何约束，可随意转移控制，使得过程的结构不清晰，不便于逐步求精。

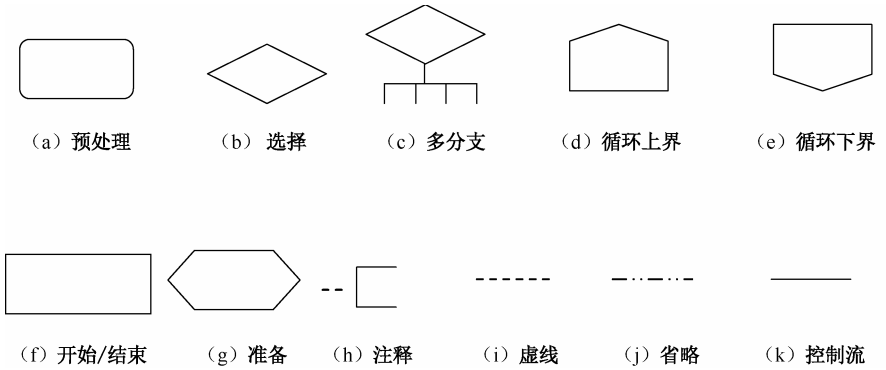


图 1-3　传统流程图的基本描述符号

1.2.3　N-S 图

N-S 图又称为盒图，它是由 Nassi 和 Shneiderman 提出的，其基本描述符号如图 1-4 所示。

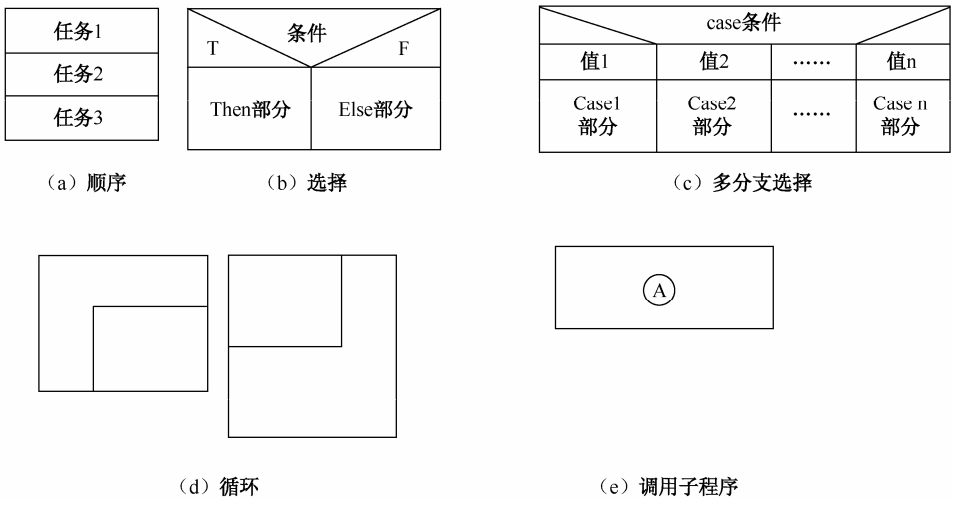


图 1-4　N-S 图基本描述符号

N-S 图易于表示结构化程序的层次结构，确定局部和全局数据的作用域。由于没有转向箭头，因此不允许随意转移控制，使得程序结构较为清晰。

1.2.4　过程描述语言

过程设计语言（PDL）又称伪码或结构化语言，现在已经有多种不同的过程设计语言。PDL 具有严格的关键字外部语法，用于定义控制结构和数据结构。同时，PDL 所表示的实际操作和内部语法通常又是灵活自由的，以便适应各种工程项目的需要。因此，一般来说，PDL 是一种“混合”语言，它用一种自然语言中的词汇和一种结构化程序设计语言中的控制结构来描述算法。

一般来说，PDL 应该具有以下几个特点。

1）关键字的固定语法，提供了结构化控制结构、数据说明和模块化的特点。为了使结构清晰和可读性好，通常在所有可能嵌套使用的控制结构的头和尾都有关键字。例如，if…endif 等。

2）用自然语言的自由语法来描述处理部分。

3）具有数据说明的手段。它应该既包括简单的数据结构(如纯量和数组)，又包括复杂的数据结构（如链表或层次的数据结构）。

4）具有模块定义和调用的机制，提供各种接口描述模式。

PDL 作为一种设计工具，具有以下优点。

1）可以作为注释直接插在源程序中。这样做的好处是，能促使维护人员在修改源程序代码的同时也修改 PDL 注释，有助于保持文档和程序的一致性，提高了文档的质量。

2）可以使用普通的正文编辑程序或文字处理系统，很方便地完成 PDL 的书写和编辑工作。

3）利用已有的自动处理程序，可以自动由 PDL 生成程序代码。

PDL 的缺点是：不如图形工具形象直观；描述复杂的条件组合与动作之间的对应关系时，不如判定表清晰简单。

1.2.5 数据字典

数据字典（Data Dictionary，DD）是开发者与用户相互沟通的有效途径之一。它能形象地向用户描述开发者的意图，使用户明白数据库可能具有的项目，可有效减小开发者和用户之间的交流障碍，也有利于用户向开发者提出自己的需求，避免因理解分歧造成的代价巨大的接口问题。

数据字典是各类数据描述的集合，它是进行详细的数据收集和数据分析后所获得的主要成果，是结构化分析方法的核心。数据字典对数据流图中的各个元素做完整的定义与说明，是数据流图的补充工具。数据流图与数据字典共同构成系统的逻辑模型。

数据字典的内容包括以下信息：图形元素的名称、别名或编号、分类、描述、定义、位置等。数据字典中所有的定义都是严密的、精确的，不可有二义性。

下面是数据字典中经常出现的一些符号的说明，设 X、a 和 b 为数据元素。

X=a+b，表示 X 由 a 和 b 组成。

X=[a,b]，X=[a/b]，表示 X 由 a 或 b 组成。

X=（a），表示 a 可在 X 中出现，也可能不出现。

X={a}，表示 X 由 0 个或多个 a 组成。

X=a .. b，表示 X 可取 a～b 的任一值。

X=m{a}n，表示 X 由 m～n 个 a 组成，即至少有 m 个 a，至多有 n 个 a。

X="a"，表示 X 为取值 a 的基本数据之类，即 a 无需进一步定义。

例 1-1　数据字典示例。数据文件“ 存折 ”的格式如图 1-5 所示。

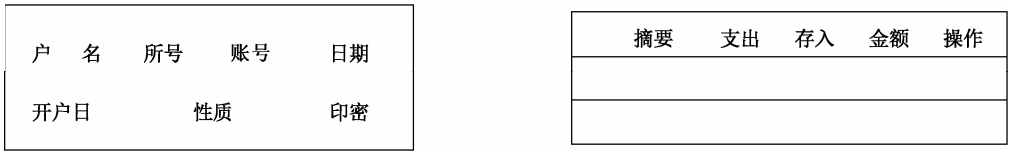


图 1-5 存折的格式

此文件在数据字典中的定义格式如下。

```
存折=户名+所号+账号+开户日+性质+（印密）+1 {存取行}50
户名=2{字母}24 （注：户名中字母至少出现2次，至多出现24次）
所号=“001”..“999” （注：所号规定为三位数）
账号=“0000000001”..“9999999999” （注：账号规定为十位数字）
开户日=年+月+日
性质=“1”..“6” （注：“1”表示普通用户，“5”表示工资用户等）
印密=“0” （注：印密在存折上不显示）
存取行=日期+（摘要）+支出+存入+余额+操作+复核
日期=年+月+日
年=“0001”..“9999”
月=“01”..“12”
日=“01”..“31”
摘要=1{字母}4 （注：表明此次操作是存还是取）
支出=金额
金额=“0000000.01”..“9999999.99”
操作=“00001”..“99999” （注：操作指银行职员代码，用五位整数表示）
……
```

1.2.6 程序设计风格

程序设计风格指一个人编制程序时所表现出来的特点、习惯、逻辑思路等。 在程序设计中要使程序结构合理、清晰，形成良好的编程习惯，对程序的要求不仅是可以在机器上执行，给出正确的结果，还要便于程序的调试和维护，增加程序的可读性。

为了提高程序的可阅读性，要建立良好的编程风格，它包括良好的代码设计，函数模块，接口功能及可扩展性等，更重要的是程序设计过程中代码的风格，包括缩进、注释、变量及函数的命名等。程序设计风格应该遵循以下几个原则。

1. 源程序文档化原则

1）标识符应按意取名（见名知意）。

2）程序应加注释。注释是程序员与读者之间通信的重要工具，用自然语言或伪码描述。它说明了程序的功能，特别在维护阶段，对理解程序提供了明确指导。注释分为序言性注释和功能性注释。

序言性注释应置于每个模块的起始部分，它要说明每个模块的用途、功能；说明模块的接口，如调用形式、参数描述及从属模块的清单；重要数据的名称、用途、限制、约束及其他信息的描述。

功能性注释嵌入在源程序内部，说明程序段或语句的功能及数据的状态。注释时，不

要每一行程序都加注释，要使用空行、缩格或括号，以便区分注释和程序。

2. 数据说明原则

1）数据说明顺序应规范，使数据的属性更易于查找，从而有利于测试、纠错与维护。

2）一个语句说明多个变量时，各变量名按字典序排列。

3）对于复杂的数据结构，要加注释，说明在程序实现时的特点。

3. 语句构造原则

简单直接，不能为了追求效率而使代码复杂化。为了便于阅读和理解，不要一行多个语句，不同层次的语句采用缩进形式，使程序的逻辑结构和功能特征更加清晰。要避免复杂的判定条件、多重的循环嵌套。表达式中使用括号以提高运算次序的清晰度等。

4. 输入输出原则

1）输入操作步骤和输入格式尽量简单。

2）应检查输入数据的合法性、有效性，提示必要的输入状态信息及错误信息。

3）输入一批数据时，使用一个特殊数据或文件结束标志，而不要用计数来控制。

4）交互式输入时，提供可用的选择和边界值。

5）当程序设计语言有严格的格式要求时，应保持输入格式的一致性。

6）输出数据规范化、图形化。

5. 追求效率原则

该原则指处理机时间和存储空间的使用，对效率的追求明确以下几点。

1）效率是一个性能要求，目标在需求分析时给出。

2）追求效率建立在不损害程序可读性或可靠性的基础上，要先使程序正确、清晰，再提高程序效率。

3）提高程序效率的根本途径在于选择良好的设计方法，良好的数据结构、算法，而不是靠编程时对程序语句做调整。

第 2 章　程序设计实训指导

程序设计实训是学习掌握程序设计语言必不可少的实践环节，是在学习程序设计课程后进行的一次综合性的程序设计训练。本章将对综合程序设计训练过程中所涉及的问题分析、总体设计、详细设计、代码编写、程序调试等各个环节进行概述。通过两个具体的实例——学生成绩管理系统和通讯录信息管理系统，对综合程序设计的内容、方法、步骤进行详细指导。

本章所涉及的实例是在 Visual C++ 6.0 开发环境下进行调试的，所给出的示例程序是对所学 C 语言知识的综合运用，使学生了解程序设计的框架和思路，掌握综合程序设计的方法和过程。与实际商用软件相比，实例的功能还有不完善之处，初学者不必苑囿于其中的一些细节。

2.1　综合程序设计概述

在程序设计语言课程中，基础实验训练和综合程序设计训练是必不可少的两个重要实践环节。但两者的目的和要求有较大差别，课程基础实验着重于对某个章节所涉及知识点进行实践和验证，具有一定的针对性和局限性，仅让学生完成基础课程实验的实践环节，学生对所学知识的掌握是片段式，离散性的，不能很好地将所学知识前后贯穿、系统应用，进行综合程序设计的能力几乎为零。综合程序设计训练需要对各个章节的所学知识进行综合运用和掌握，让所有零散的知识点串联成一个有机的整体，让学生有机会从一个综合的、全局的高度去运用学过的所有知识，达到既能巩固所学知识点，又能提高综合程序设计能力的目的。

① 通过程序设计实训，复习巩固基本知识点：数据类型、运算符、三种基本结构（顺序结构、选择结构、循环结构）、函数的定义和调用、结构体、指针、文件操作、编译预处理等。

② 通过程序设计实训，了解软件开发的基本过程：问题的定义和规划、需求分析、软件设计、程序编码、软件测试、后期维护、文档制作等。

③ 通过程序设计实训，熟悉程序设计，掌握程序编码、程序测试，学会制作程序文档。

④ 通过程序设计实训，进一步熟悉分步调试、断点设置的方法，掌握程序系统调试、测试的方法。

2.1.1　综合程序设计过程

软件开发周期中主要包括问题定义、可行性分析、需求分析、总体设计、详细设计、软件编码、软件测试、软件维护等八个基本阶段。在程序设计实训中，为使问题简化，便

于初学者学习和理解，将上述八个阶段简化分解为六个步骤，即问题分析、总体设计、详细设计、代码编写、程序调试、程序文档制作。

1. 问题分析

根据综合训练题目的要求，分析所要解决的问题，需要输入哪些数据，数据如何存放，对数据做何种处理，即所要实现的主要功能、子功能有哪些，输出什么结果。

2. 总体设计

（1）功能图

根据问题分析的情况，画出整体系统功能图，如图 2-1 所示。

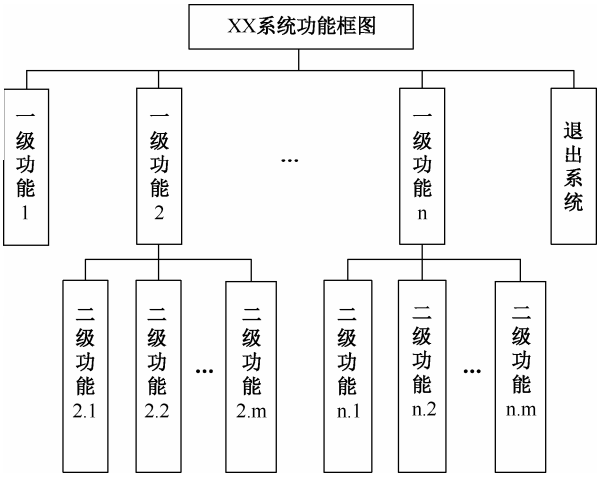


图 2-1　系统功能图

（2）数据设计

定义出整个程序都会用到的数据类型和外部变量，根据对问题的分析，主要对以下各类数据进行设计。

① 宏定义程序中的符号常量。

② 定义程序中使用的所有全局、外部变量。

③ 定义程序设计中需要用到的结构体、链表类型。

④ 确定程序中主要数据的数据类型及数据的存储结构。

（3）函数设计

设计出整个程序的所有函数模块，包括函数类型、函数名称、函数参数。函数设计一般以功能实现为主线，围绕程序的一个功能进行函数设计。有些函数之间有调用和被调用关系，在进行函数设计时需要注意顺序问题，哪些函数先设计，哪些函数后设计，而没有调用关系的函数可以并列设计。当多人合作进行一个综合程序设计时，可以并列设计的函数可由不同的人员承担。根据函数模块之间的调用关系，画出调用关系图，如图 2-2 所示。

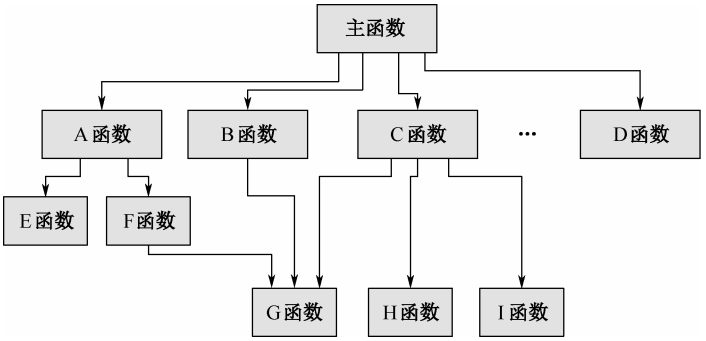


图 2-2　系统中函数调用关系

3. 详细设计

函数模块流程图：画出主要函数模块的流程图。
函数模块的数据设计：确定每个函数中用到的局部变量的类型、名称和变量的作用。

4. 代码编写

根据详细设计中确定的每个函数模块的数据名称，结合每个函数模块的流程图，编写具体代码。

5. 代码调试

程序整体调试是程序设计的必要阶段，是在前期程序设计调试的基础上进行的基本过程。需要设计一个较大规模的数据集，按照综合程序设计题目的功能要求，对组装完成的程序逐项进行功能测试和调试，直至确认程序达到设计目标为止。

6. 编写程序文档

综合程序设计总结是综合程序设计的最终阶段，通过对综合程序设计的各个过程进行系统全面的总结，按照指导教师的具体要求，形成综合程序设计报告。

2.2　学生成绩信息管理系统设计

综合运用所学程序设计语言知识，设计一个学生成绩管理系统，实现对学生成绩的输入、浏览、查询、统计、修改、删除、保存等功能。

2.2.1　设计要求

1. 功能要求

(1) 输入功能
通过数据文件输入；
通过键盘追加记录。

（2）浏览功能

输出所有学生的学号、姓名、各科成绩等信息。

（3）查询功能

通过学号查询学生信息；

按成绩查询各个分数段的学生信息。

（4）统计功能

统计各科平均成绩为优秀、良好、中等、及格、不及格的学生人数及所占百分比。

（5）修改功能

通过输入学号修改指定学生的所有信息。

（6）删除功能

通过输入学号删除指定学生的所有信息。

（7）保存功能

将当前学生信息保存至指定的数据文件中。

（8）退出功能

退出学生成绩管理系统。

2. 技术要求

1）每个学生记录应包括以下信息：学号、姓名、高等数学成绩、大学物理成绩、英语成绩、计算机成绩、平均成绩、总成绩等。

2）初始信息包括学号、姓名、高等数学成绩、大学物理成绩、英语成绩、计算机成绩等，信息从一个磁盘数据文件读入程序。

3）以菜单方式实现各个功能的选择控制。

4）对于学生成绩管理系统中的输入、浏览、查询、统计、修改、删除、保存等功能，要求编写独立的函数模块或主控函数来实现，每个模块所属的子功能也尽量由独立的函数实现。

5）退出程序前，将当前学生的所有信息保存到一个磁盘数据文件中。数据文件的存储位置、文件名、文件格式由设计者确定。

2.2.2 设计过程

1. 问题分析

学生成绩管理系统需要处理的数据是学生信息记录，每条记录中涉及的学号、姓名、高等数学成绩、大学物理成绩、英语成绩、计算机成绩、平均成绩、总成绩等信息，其数据类型各不相同，既有数值类型，又有字符类型，并且每条记录的各项信息是一个不可分割的整体，所以要考虑使用结构体数据类型存储学生信息记录，以结构体数组存储所有学生记录。对于所要实现的功能要求，实际上就是对数组的输入、输出、插入、查询、删除等基本操作。

2. 总体设计

（1）功能设计

依据上面的问题分析，可以将这个系统分为数据输入、数据浏览、数据统计、数据查询、数据修改、数据删除、数据保存、退出系统共八大功能模块，如图 2-3 所示。

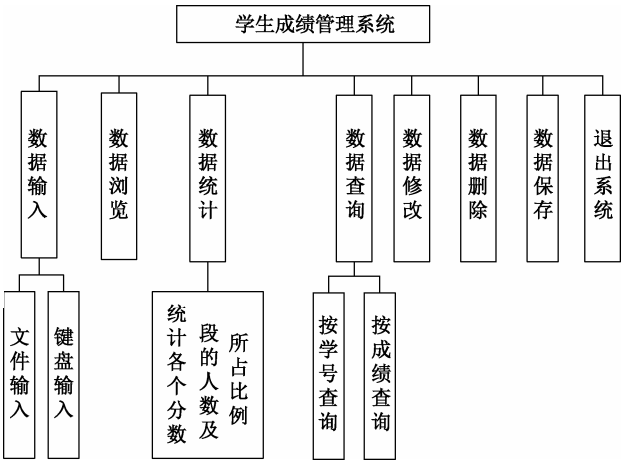


图 2-3　学生成绩管理系统功能图

（2）数据设计

根据问题分析，进行数据设计，定义一个包含学号、姓名、高等数学成绩、大学物理成绩、英语成绩、计算机成绩、平均成绩、总成绩共八个成员的结构体类型，作为学生信息记录的存储类型；定义一个结构体类型的数组存放若干学生记录；定义一个符号常量 N 作为数组的长度；定义全局变量 LENGTH 存放实际学生记录条数。

```
#define N  50;                    /* N宏定义为学生人数 */
```

定义一个学生信息的结构体类型：

```
struct  student
{
   int       num;                 /* 学号 */
   char      name[20];            /* 姓名 */
   float     math;                /* 高等数学成绩 */
   float     physics;             /* 大学物理成绩 */
   float     english;             /* 英语成绩 */
   float     computer;            /* 计算机成绩 */
   float     sumscore;            /* 总成绩 */
   float     avescore             /* 平均成绩 */
};
struct student student[N];        /* 定义结构体数组存放学生记录 */
int LENGTH;                       /* 数组中的实际信息记录条数 */
```

（3）函数设计

根据功能设计，为学生成绩管理系统设计如下函数来实现程序各模块的功能。

main()：函数的原型为 int main()，是学生成绩管理系统的主函数，控制各功能函数模

块的调用，实现整个程序的各项功能。

menu()：函数的原型为 void menu()，用于实现程序的主菜单显示。

inputdata()：函数的原型为 void inputdata()，是数据输入函数，从磁盘数据文件 stuin.txt 中读取学生初始信息，存储到程序中对应的结构体数组中，并调用 sum_average 函数。

sum_average()：函数的原型为 void sum_average()，实现计算每个学生的总分和平均分功能。

append()：函数的原型为 void append()，是数据输入函数，实现学生信息的追加，通过键盘输入一条或者多条学生记录。

display()：函数的原型为 void display()，实现所有学生信息的显示、浏览。

query_num()：函数的函数原型为 void query_num()，实现按学号查询学生的信息。

query_score()：函数的原型为 void query_score ()，实现按成绩查询学生的信息。

count()：函数的原型为 void count()，是统计功能的主控函数，统计各分数段学生人数及所占的百分比。

modify_data ()：函数的原型为 void modify_data()，实现按学号修改学生的错误信息。

del_data ()：函数的原型为 void del_data()，实现按学号删除学生信息。

save_data ()：函数的原型为 void save_data()，实现程序退出前的数据保存工作，在程序退出前，将程序中所有学生信息写入指定的磁盘文件中。

（4）函数调用关系

以上函数的相互调用关系如图 2-4 所示。

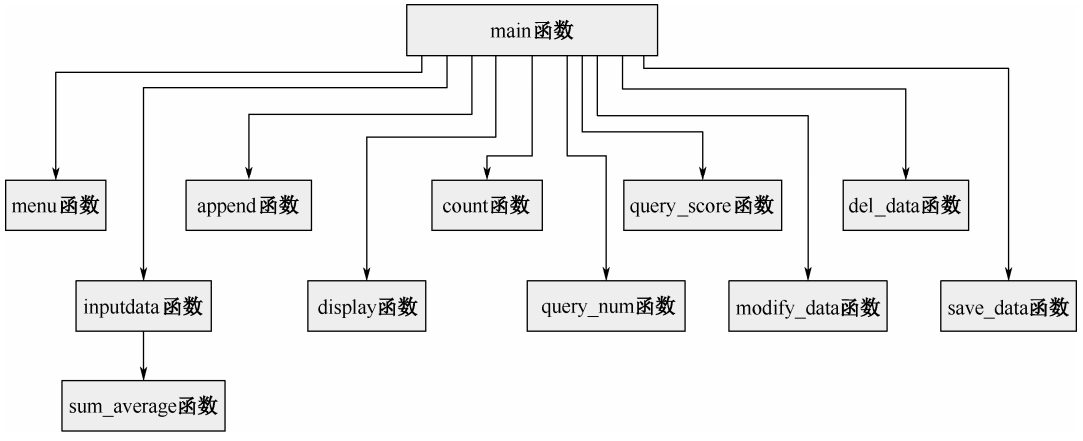


图 2-4　学生成绩管理系统函数调用关系图

3. 详细设计

（1）main()函数

main()函数模块流程图如图 2-5 所示。

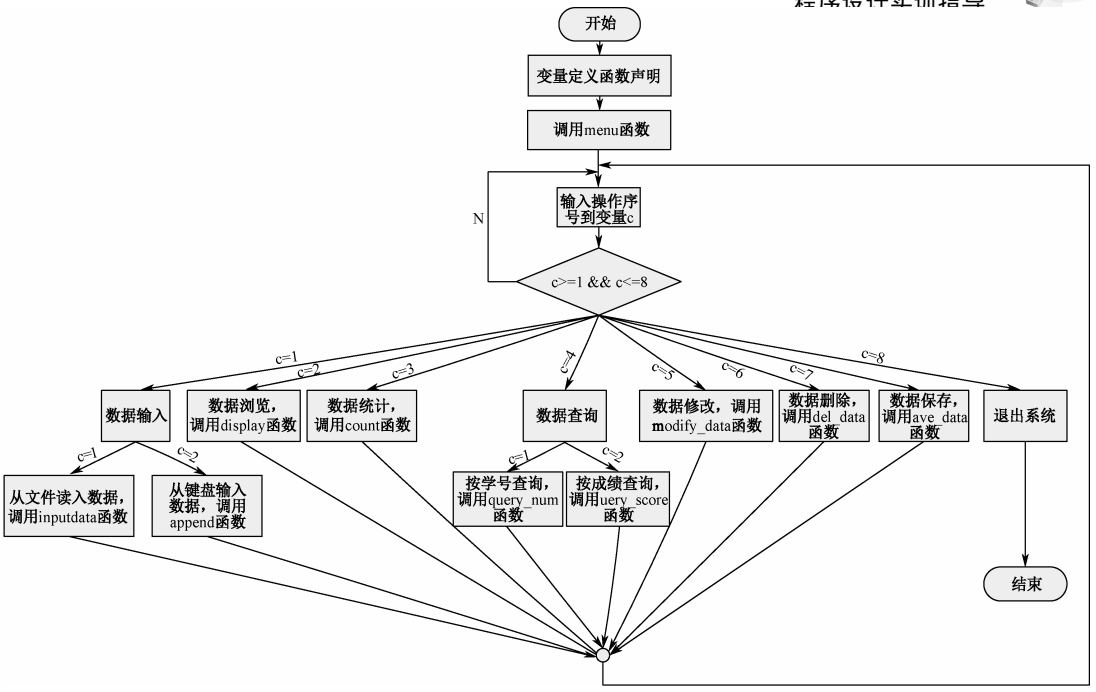


图 2-5　main()函数流程图

函数内部数据定义

```
int c;                                    /*存放输入的菜单序号*/
```

（2）inputdata()函数

inputdata()函数模块流程图如图 2-6 所示。

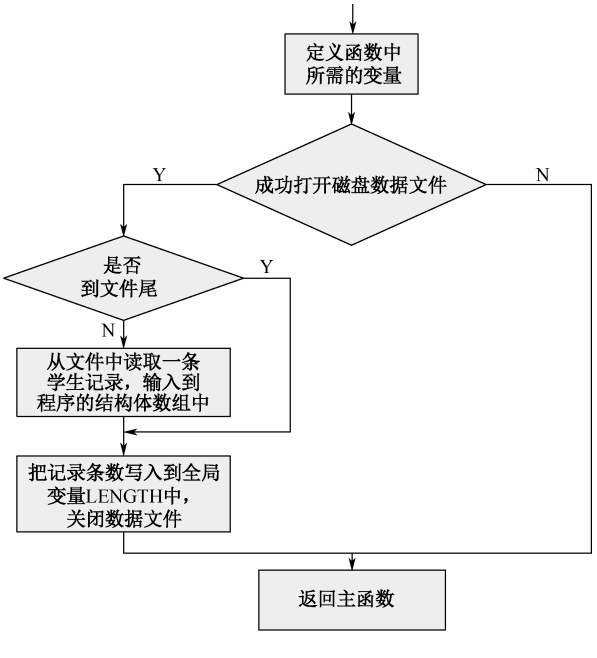


图 2-6　inputdata()函数流程图

函数内部数据定义：

```
int  i;                              /*数组下标*/
FILE *infp;                          /* 定义文件指针变量  */
```

（3）append()函数

append()函数模块流程图如图 2-7 所示。

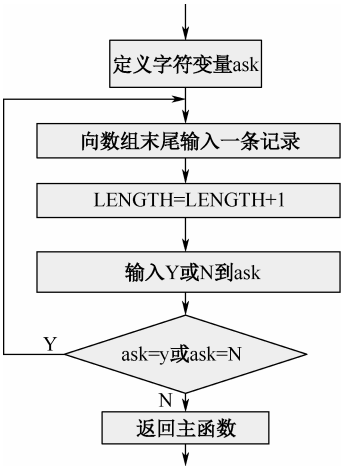


图 2-7　append()函数流程图

函数内部数据定义

```
char ask;
```

（4）display()函数

display()函数模块流程图如图 2-8 所示。

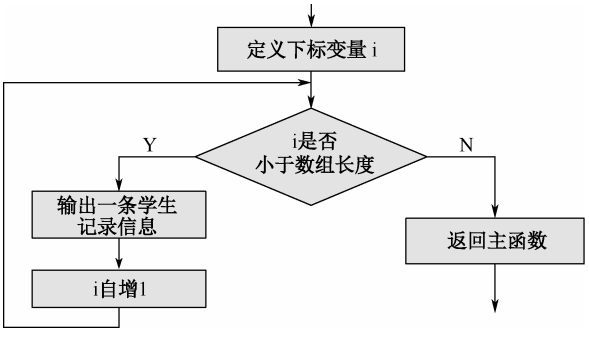


图 2-8　display()函数流程图

函数内部数据定义：

```
int  i;                              /*数组下标*/
```

（5）count()函数

count()函数模块流程图如图 2-9 所示。

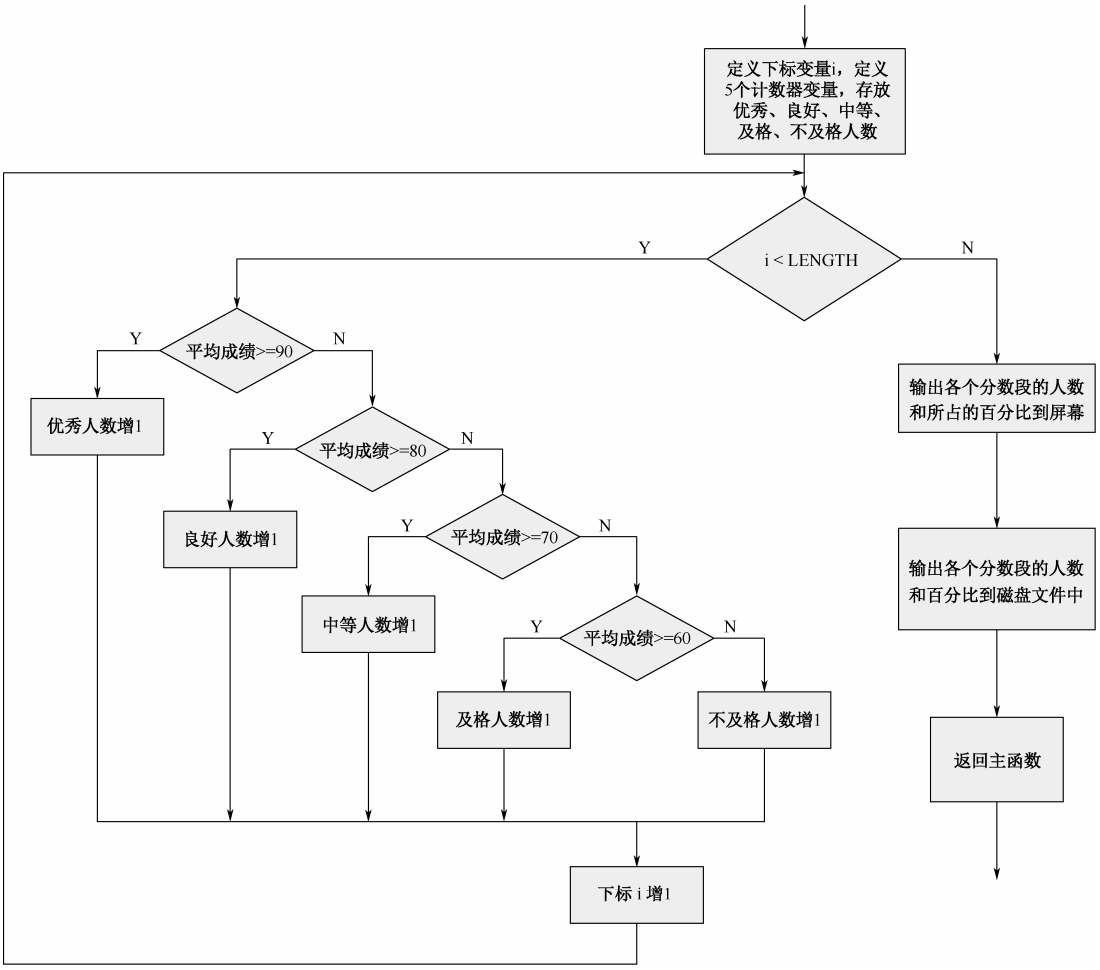


图 2-9　count()函数流程图

函数内部数据定义：

```
int i;                                  /* 定义数组下标*/
int excellent,good,medium,pass,fail;    /* 定义各类成绩计数器变量 */
FILE *outfp;
excellent=0;                            /* 优秀人数 */
good=0;                                 /* 良好人数 */
medium=0;                               /* 中等人数 */
pass=0;                                 /* 及格人数 */
fail=0;                                 /* 不及格人数 */
```

（6）query_num()函数

query_num()函数模块流程图如图 2-10 所示。

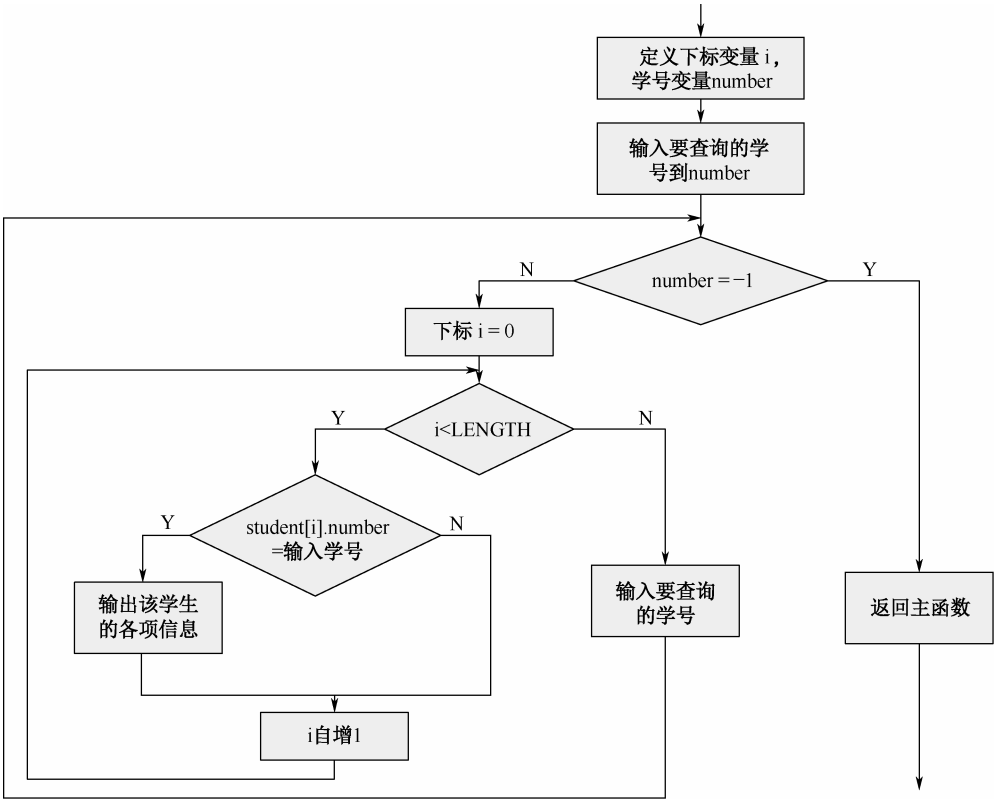


图 2-10　query_num()函数流程图

函数内部数据定义：

```
    int i,number;                              /* 定义数组下标和学号变量*/
```

（7）query_score()函数

query_score()函数模块流程图如图 2-11 所示。

函数内部数据定义：

```
    int i;                  /* 定义数组下标变量*/
    FILE *outfp;            /* 定义文件指针变量*/
```

（8）modify_data()函数

modify_data()函数模块流程图如图 2-12 所示。

函数内部数据定义：

```
    int i,number;           /* 定义数组下标变量和学号变量*/
```

（9）del_data()函数

del_data()函数模块流程图如图 2-13 所示。

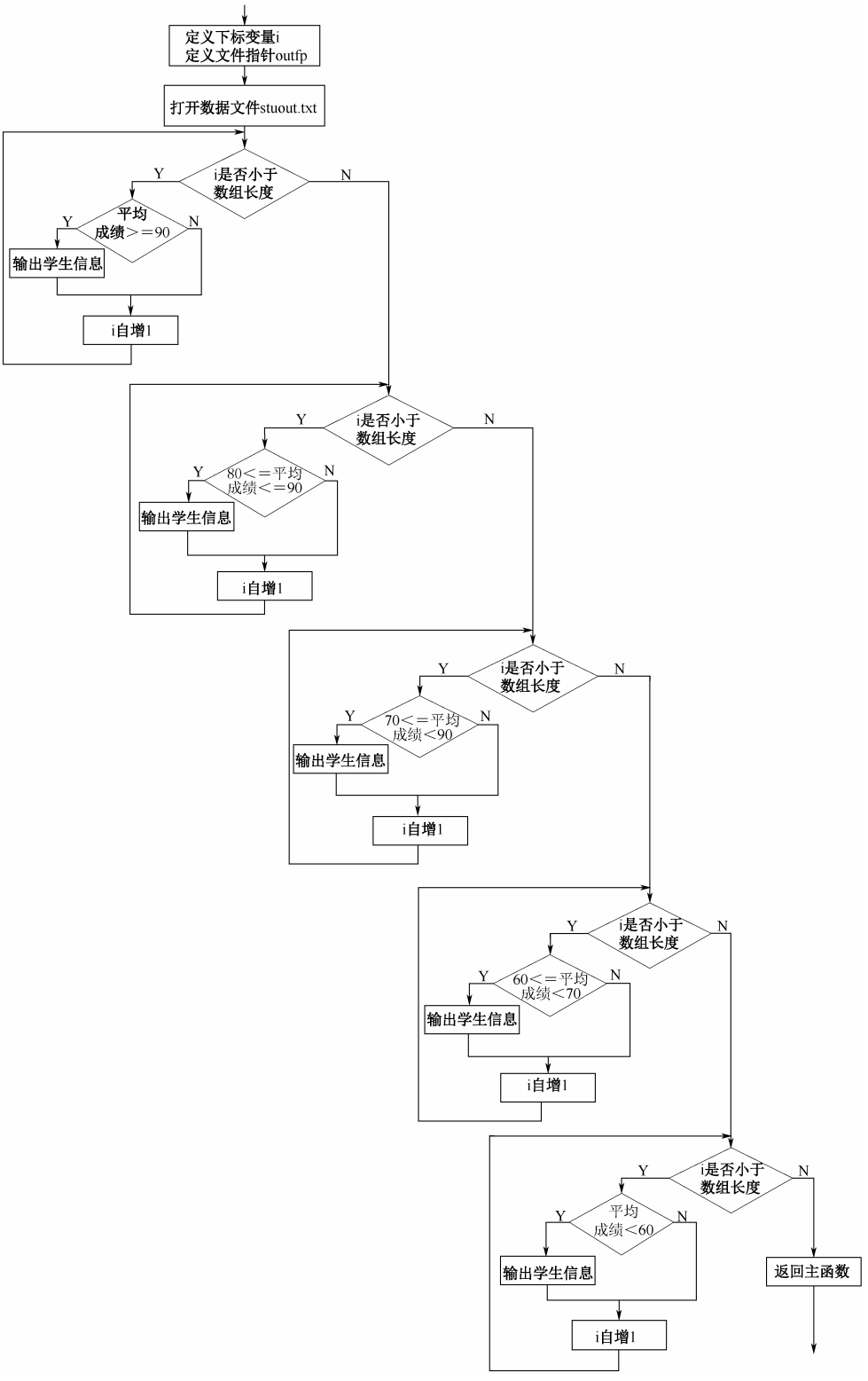


图 2-11　query_score()函数流程图

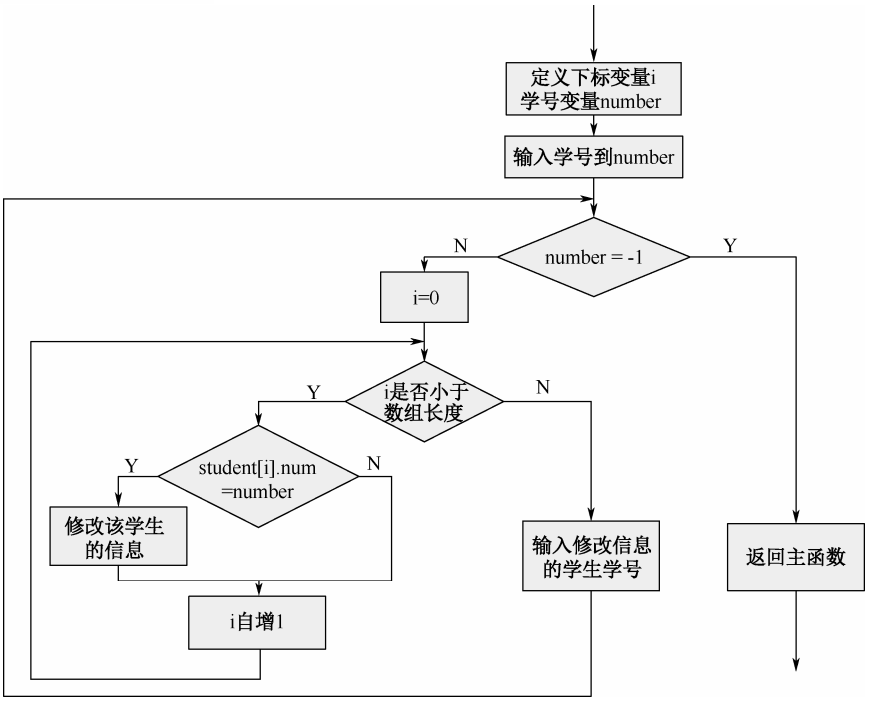


图 2-12　modify_data()函数流程图

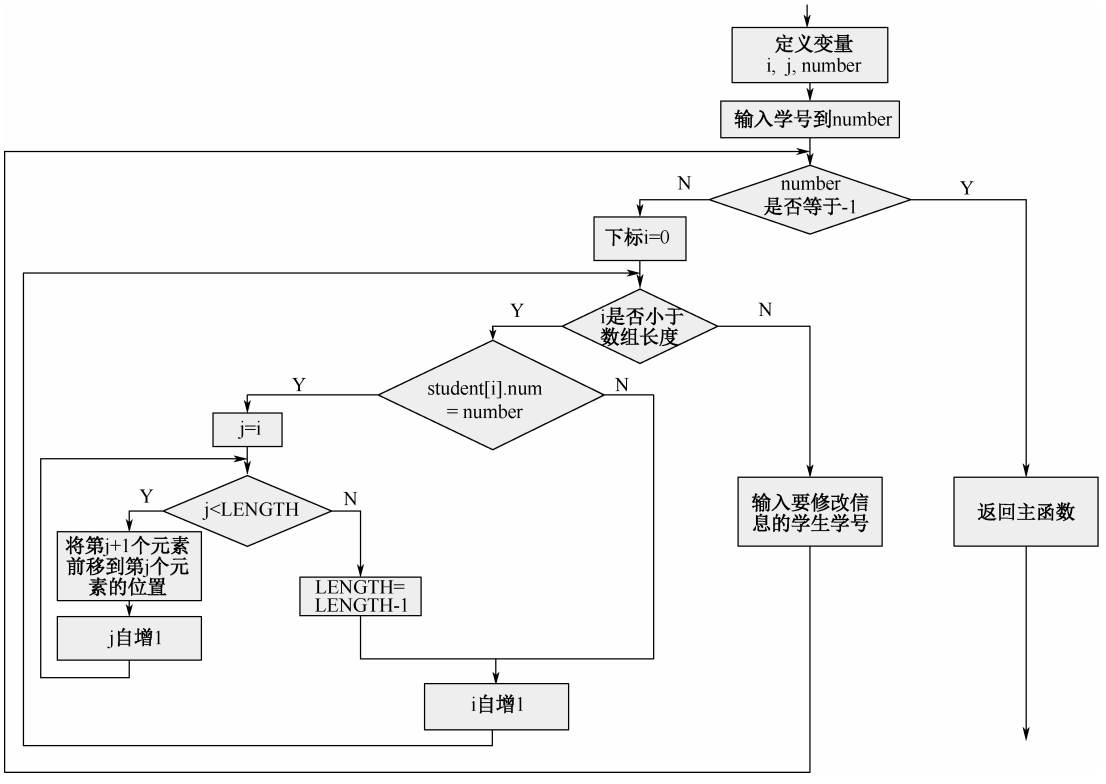


图 2-13　del_data()函数流程图

函数内部数据定义：

```
int i, number, j;              /*  定义下标变量i，j，学号变量number */
```

（10）save_data()函数

save_data()函数模块流程图如图 2-14 所示。

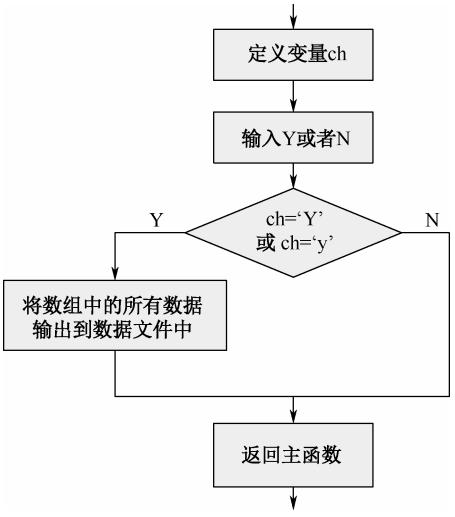


图 2-14　save_data()函数流程图

函数内部数据定义：

```
char ch;                /* 存放是否保存的标志信息 */
int i;                  /* 定义下标变量i */
```

4. 代码编写

（1）代码框架

```
#define N 100
#include"stdio.h"
#include"stdlib.h"
#include"conio.h"
#include"string.h"

struct  student
{
     int       num;                /*  学号  */
     char      name[20];           /*  姓名  */
     float     math;               /*  高等数学成绩  */
     float     physics;            /*  大学物理成绩  */
     float     english;            /*  英语成绩  */
     float     computer;           /*  计算机成绩  */
     float     sumscore;           /*  总成绩  */
     float     avescore;           /*  平均成绩  */
}student[N];
int LENGTH;                       /*数组中的实际信息记录条数*/
```

```
/* 函数功能：调用菜单函数，通过输入的菜单项序号，控制整个程序的运行 */
int main()
{

}

/* 函数功能：显示程序菜单项 */
void menu()
{

}

/* 函数功能：从磁盘文件中读取初始学生记录信息送入程序的结构体数组 */
int inputdata()
{

}

/* 函数功能：在数组末尾追加学生记录 */
void append()
{


}

/* 函数功能：浏览所有学生记录信息 */
void display()
{

}

/* 函数功能：统计优秀、良好、中等、及格、不及格人数及所占百分比 */
void count()
{


}

/* 函数功能：按学号查询学生记录信息 */
void query_num()
{

}

/* 函数功能：按成绩查询学生记录信息 */
void query_score()
{
```

```
}

/* 函数功能：按学号修改学生记录信息 */
void modify_data()
{

}

/* 函数功能：按学号删除学生记录信息 */
void del_data()
{

}

/* 函数功能：保存最新的学生数据记录到磁盘文件中 */
void save_data()
{

}
```

(2) 完整代码

```
#define N 100
#include"stdio.h"
#include"stdlib.h"
#include"conio.h"
#include"string.h"

struct  student
{
    int     num;                /* 学号 */
    char    name[20];           /* 姓名 */
    float   math;               /* 高等数学成绩 */
    float   physics;            /* 大学物理成绩 */
    float   english;            /* 英语成绩 */
    float   computer;           /* 计算机成绩 */
    float   sumscore;           /* 总成绩 */
    float   avescore;           /* 平均成绩 */
};
struct student student[N];      /* 定义结构体数组存放学生记录 */
int LENGTH;                     /* 数组中的实际信息记录条数 */


/* 函数功能：调用菜单函数，通过输入的菜单项序号，控制整个程序的运行 */
int main()
{
    int c;                      /* 存放输入的菜单项序号 */
```

```
FILE  *outfp;
void menu();
int  inputdata();
void sum_average();
void append();
void display();
void count();
void query_num();
void query_score();
void modify_data();
void del_data();
void save_data();
/* 在磁盘上指定位置创建一个结果数据文件，存放输出的数据 */
outfp=fopen("..\\材成141张晓雪\\stuout.txt","w");
fclose(outfp);       /* 暂时关闭文件 */
menu();
while(1)             /* 遇到非法输入时，重新输入 */
{

     printf("\n\t 请输入您要选择的操作序号，按回车键确认：");
     scanf("%d",&c);              /* 输入菜单项序号 */
     printf("\n");
     if(c>8||c<1)                 /* 判断输入的菜单项序号是否合法 */
          printf("------对不起，没 有 此 项 功 能 -----!\n");
     /* 通过输入的菜单项序号，调用相应的函数，执行对应的操作 */
     switch(c)
     {
     case 1:                      /* 数据输入 */
          {   int c;
          printf("\t***1.磁盘文件输入***\n");
          printf("\t***2.追加学生记录***\n\t");
          while(1)                /* 遇到非法输入时，重新输入 */
          {
             scanf("%d", &c);     /* 输入子菜单项序号 */

             if(c>2||c<1)         /* 判断输入的子菜单项序号是否合法 */
             printf("------对不起，请输入1或者2 -----!\n");
           else
             break;
           }

           switch(c)
           {
           case 1: inputdata();  break;      /* 从磁盘文件读入 */
           case 2: append(); break;          /* 追加记录 */
           }
           }  break;
```

```
        case 2:display();  break;                    /* 数据浏览 */

        case 3:count();break;                        /* 数据统计 */

        case 4:                                      /* 数据查询 */
        {
              int c;
              printf("\t***1.按学号查询***\n");
              printf("\t***2.按成绩查询***\n\t");
              while(1)
              {
                    scanf("%d", &c);
                    if(c>2||c<1)
                        printf("------对不起，请输入1或者2 -----!\n");
                    else
                        break;
              }
              switch(c)
              {
              case 1: query_num();     break;  /* 按学号查询 */
              case 2: query_score();   break;  /* 按成绩查询 */
              }
            };      break;

        case 5:modify_data();       break;       /* 数据修改 */
        case 6:del_data();          break;       /* 数据删除 */
        case 7:save_data();         break;       /* 数据保存 */
        case 8: exit(0);                         /* 退出系统 */
        }
    }
    getch();
    return 0;
}


/* 函数功能：显示程序菜单项 */
void menu()
{
     printf("\n\n\n\n");

     printf("\t*************************************************n");
     printf("\t*************************************************n");
     printf("\t\t    ※ ☆ 欢 迎 光 临  ☆ ※\n");
     printf("\n");
     printf("\t\t  ★★学 生 成 绩 管 理 系 统 ★★ \n");
     printf("\n");
```

```
        printf("\t\t\t   ★1.数据输入★ \n");
        printf("\n");
        printf("\t\t\t   ★2.数据浏览★ \n");
        printf("\n");
        printf("\t\t\t   ★3.数据统计★ \n");
        printf(" \n");
        printf("\t\t\t   ★4.数据查询★ \n");
        printf("\n");
        printf("\t\t\t   ★5.数据修改★ \n");
        printf("\n");
        printf("\t\t\t   ★6.数据删除★ \n");
        printf("\n");
        printf("\t\t\t   ★7.数据保存★ \n");
        printf("\n");
        printf("\t\t\t   ★8.退出系统★ \n");
        printf("\n\n");
        printf("\t**************************************************n");
        printf("\t**************************************************n");
}


/* 函数功能：从磁盘文件中读取初始学生记录信息送入程序的结构体数组 */
int inputdata()
{
        int i;              /* 定义下标变量i */
        FILE *infp;
        /* 是否成功打开数据文件*/
        if((infp=fopen("..\\材成141张晓雪\\stuin.txt","r"))==NULL)
        {
           printf("can not open file stuin.txt!");
           getch();
           return -1;
        }

        i=0;
        while(!feof(infp))       /* 文件指针未到文件尾，继续读入数据 */
        {
            fscanf(infp,"%d%s%f%f%f%f",&student[i].num,&student[i].name,

        &student[i].math,&student[i].physics,&student[i].english,
                &student[i].computer);
            i=i+1;
        }
        LENGTH=i;
        fclose(infp);
        sum_average();
        return 0;
```

```
}

/* 函数功能：计算每个学生的总成绩和平均成绩 */
void sum_average()
{
     int i;
     for(i=0;i<LENGTH;i++)
     {
         student[i].sumscore=student[i].math+student[i].physics+
             student[i].english+student[i].computer;
         student[i].avescore=student[i].sumscore/4;
     }
}

/* 函数功能：在数组末尾追加学生记录 */
void append()
{
     char ask;
     while(1)
     {
         printf("   请按顺序输入\n   学号\t   姓名\t\t
             高等数学\t大学物理\t英语\t计算机，数据之间用Tab键分隔\n   ");
         scanf("%d%s%f%f%f%f",&student[LENGTH].num,&student[LENGTH].
                 name,
             &student[LENGTH].math,&student[LENGTH].physics,
             &student[LENGTH].english,&student[LENGTH].computer);
         student[LENGTH].sumscore=student[LENGTH].math+student
                 [LENGTH].physics
                 +student[LENGTH].english+student[LENGTH].computer;
         student[LENGTH].avescore=student[LENGTH].sumscore/4;
             LENGTH=LENGTH+1;
             printf("   是否继续追加Y/y?");
             getchar();
             scanf("%c",&ask);
             if(!(ask=='y'||ask=='Y'))
                 break;
     }
}

/* 函数功能：浏览所有学生记录信息 */
void display()
{
     int i;
     printf("   学号\t   姓名\t\t高等数学\t大学物理
```

```
            \t英语\t计算机\t总分\t平均分\n");
        for(i=0;i<LENGTH;i++)
            printf("   %d\t%s\t%.2f\t\t%.2f\t\t%.2f\t%.2f\t%.2f\t%.2f\n",
                student[i].num,student[i].name,student[i].math,
                student[i].physics,
                student[i].english,student[i].computer,
                student[i].sumscore,
                student[i].avescore);
    }


    /* 函数功能：统计优秀、良好、中等、及格、不及格人数 */
    void count()
    {

        int i;                      /* 定义数组下标*/
        int excellent,good,medium,pass,fail;    /* 定义各类成绩计数器变量 */
        FILE *outfp;
        excellent=0;                /* 优秀人数 */
        good=0;                     /* 良好人数 */
        medium=0;                   /* 中等人数 */
        pass=0;                     /* 及格人数 */
        fail=0;                     /* 不及格人数 */
        for(i=0;i<LENGTH;i++)
        {
            if(student[i].avescore>=90)             /* 优秀条件 */
                excellent++;                        /* 优秀人数计数器 */
            else if(student[i].avescore>=80)        /* 良好条件 */
                good++;                             /* 良好人数计数器 */
            else if(student[i].avescore>=70)        /* 中等条件 */
                medium++;                           /* 中等人数计数器 */
            else if(student[i].avescore>=60)        /* 及格条件 */
                pass++;                             /* 及格人数计数器 */
            else
                fail++;                             /* 不及格人数计数器 */
        }
        printf("\t平均成绩各分数段的统计结果如下:\n");
        printf("\t平均成绩优秀人数为：%d\t\t，优秀率为：%d%%\n",excellent,
        excellent);
        printf("\t平均成绩良好人数为：%d\t\t，良好率为：%d%%\n",good,good);
        printf("\t平均成绩中等人数为：%d\t\t，中等率为：%d%%\n",medium,medium);
        printf("\t平均成绩及格人数为：%d\t\t，及格率为：%d%%\n",pass,pass);
        printf("\t平均成绩不及格人数为：%d\t\t，不及格率为：%d%%\n",fail,fail);
            outfp=fopen("..\\材成141张晓雪\\stuout.txt","a");
        fprintf(outfp,"平均成绩各分数段的统计结果如下:\n");
        fprintf(outfp,"平均成绩优秀人数为：%d\t\t，优秀率为：
                %d%%\n",excellent,excellent);
```

```
    fprintf(outfp,"平均成绩良好人数为：%d\t\t,良好率为：%d%%\n",good,good);
    fprintf(outfp,"平均成绩中等人数为：%d\t\t，中等率为：
            %d%%\n",medium,medium);
    fprintf(outfp,"平均成绩及格人数为：%d\t\t,及格率为：%d%%\n",pass,pass);
    fprintf(outfp,"平均成绩不及格人数为：%d\t\t，不及格率为：
           %d%%\n",fail,fail);
    fclose(outfp);
}


/* 函数功能：按学号查询学生记录信息 */
void query_num()
{
    int number,i;                    /* 定义下标变量i，学号变量number */
    printf("\t请输入要查询的学号,输入-1停止查询\n\t学号：");
    scanf("%d",&number);             /* 输入查找学号 */
    while(number!=-1)                /* 遇到非法输入时，重新输入 */
    {
        for(i=0;i<LENGTH;i++)
        {
            if(student[i].num ==number)
            {
                printf("\t学号\t姓名\t\t高等数学\t大学物理\t
                            英语\t计算机\t总成绩\t平均成绩\n");
                printf("\t%d\t%s\t%.2f\t\t%.2f\t\t%.2f\t%.2f\
                        t%.2f\t%.2f\n",
                    student[i].num,student[i].name,student[i].math,
                    student[i].physics,student[i].english,
                    student[i].computer,
                    student[i].sumscore,student[i].avescore);
            }
        }
            printf("\t请输入要查询的学号，输入-1停止查询\n\t学号：");
            scanf("%d",&number);
    }
}


/* 函数功能：按成绩查询学生记录信息 */
void query_score()
{
    int i;                           /* 定义下标变量i*/
    FILE *outfp;                     /* 定义文件指针变量outfp */
    outfp=fopen("..\\材成141张晓雪\\stuout.txt","a");
    /*计算每个学生的总成绩和平均成绩到屏幕和输出文件 */
    printf("\n\t各科目平均成绩为优秀的学生有：\n");
    printf("\t学号\t姓名\t\t高等数学\t大学物理\t英语\t计算机\t总分\
```

```
            t平均分\n");
    fprintf(outfp,"\n各科目平均成绩为优秀的学生有：\n");
    fprintf(outfp,"学号\t姓名\t\t高等数学\t大学物理\t英语\t计算机\
            t总分\t平均分\n");
    /*查找成绩为优秀的学生并输出到屏幕和输出文件中*/
    for(i=0;i<LENGTH;i++)
    {
        if(student[i].avescore>=90)
        {
            printf("\t%d\t%s\t%.2f\t\t%.2f\t\t%.2f\t%.2f\t%.2f\t%.
                    2f\n",student[i].num,
                student[i].name,student[i].math,student[i].
                    physics,student[i].english,
                student[i].computer,student[i].sumscore,
                    student[i].avescore);
            fprintf(outfp,"%d\t%s\t%.2f\t\t%.2f\t\t%.2f\t%.2f\t%
                    .2f\t%.2f\n",
                student[i].num,student[i].name,
                student[i].math,student[i].physics,
                student[i].english,student[i].computer,
                student[i].sumscore,student[i].avescore);
        }
    }

    printf("\n\t各科目平均成绩为良好的学生有：\n");
    printf("\t学号\t姓名\t\t高等数学\t大学物理\t英语\t计算机\t总分\
             t平均分\n");
    fprintf(outfp,"\n各科目平均成绩为良好的学生有：\n");
    fprintf(outfp,"学号\t姓名\t\t高等数学\t大学物理\t英语\t计算机\
             t总分\t平均分\n");
    /*查找成绩为良好的学生并输出到屏幕和输出文件中*/
    for(i=0;i<LENGTH;i++)
    {
        if(student[i].avescore<90 && student[i].avescore>=80)
        {
            printf("\t%d\t%s\t%.2f\t\t%.2f\t\t%.2f\t%.
                    2f\t%.2f\t%.2f\n",student[i].num,
                student[i].name,student[i].math,
                student[i].physics,student[i].english,
                student[i].computer,
                student[i].sumscore,student[i].avescore);
        fprintf(outfp,"%d\t%s\t%.2f\t\t%.2f\t\t%.
                    2f\t%.2f\t%.2f\t%.2f\n",
                student[i].num,student[i].name,
                student[i].math,student[i].physics,
                student[i].english,student[i].computer,
                student[i].sumscore,student[i].avescore);
```

```
    }
}

printf("\n\t各科目平均成绩为中等的学生有：\n");
printf("\t学号\t姓名\t\t高等数学\t大学物理\t英语\t计算机\t总分\t平均分\n");
fprintf(outfp,"\n各科目平均成绩为中等的学生有：\n");
fprintf(outfp,"学号\t姓名\t\t高等数学\t大学物理\t英语\t计算机\t总分\
                    t平均分\n");
/*查找成绩为中等的学生并输出到屏幕和输出文件中*/
for(i=0;i<LENGTH;i++)
{
    if(student[i].avescore<80 && student[i].avescore>=70)
    {
        printf("\t%d\t%s\t%.2f\t\t%.2f\t\t%.2f\t%.
                2f\t%.2f\t%.2f\n",student[i].num,
            student[i].name,student[i].math,
            student[i].physics,student[i].english,
            student[i].computer,student[i].sumscore,
            student[i].avescore);
        fprintf(outfp,"%d\t%s\t%.2f\t\t%.2f\t\t%.
                2f\t%.2f\t%.2f\t%.2f\n",
            student[i].num,student[i].name,
            student[i].math,student[i].physics,
            student[i].english,student[i].computer,
            student[i].sumscore,student[i].avescore);
    }
}

printf("\n\t各科目平均成绩为及格的学生有：\n");
printf("\t学号\t姓名\t\t高等数学\t大学物理\t英语\t计算机\t总分\t平均分\n");
fprintf(outfp,"\n各科目平均成绩为及格的学生有：\n");
fprintf(outfp,"学号\t姓名\t\t高等数学\t大学物理\t英语\t计算机\t总分\
                    t平均分\n");
/*查找成绩为及格的学生并输出到屏幕和输出文件中*/
for(i=0;i<LENGTH;i++)
{
    if(student[i].avescore<70 && student[i].avescore>=60)
    {
        printf("\t%d\t%s\t%.2f\t\t%.2f\t\t%.2f\t%.
                2f\t%.2f\t%.2f\n",student[i].num,
            student[i].name,student[i].math,
            student[i].physics,student[i].english,
            student[i].computer,student[i].sumscore,
            student[i].avescore);
        fprintf(outfp,"%d\t%s\t%.2f\t\t%.2f\t\t%.
                 2f\t%.2f\t%.2f\t%.2f\n",
            student[i].num,student[i].name,
```

```
                student[i].math,student[i].physics,
                student[i].english,student[i].computer,
                student[i].sumscore,student[i].avescore);
        }
    }

    printf("\n\t各科目平均成绩为不及格的学生有：\n");
    printf("\t学号\t姓名\t\t高等数学\t大学物理\t英语\t计算机\t总分\t平均分\n");
    fprintf(outfp,"\n各科目平均成绩为不及格的学生有：\n");
    fprintf(outfp,"学号\t姓名\t\t高等数学\t大学物理\t英语\t计算机\t总分\
                    t平均分\n");
    /*查找成绩为不及格的学生并输出到屏幕和输出文件中*/
    for(i=0;i<LENGTH;i++)
    {
        if(student[i].avescore<60)
        {
            printf("\t%d\t%s\t%.2f\t\t%.2f\t\t%.2f\t%.
                    2f\t%.2f\t%.2f\n",student[i].num,
                student[i].name,student[i].math,
                student[i].physics,student[i].english,
                student[i].computer,student[i].sumscore,
                student[i].avescore);
            fprintf(outfp,"%d\t%s\t%.2f\t\t%.2f\t\t%.
                     2f\t%.2f\t%.2f\t%.2f\n",
                student[i].num,student[i].name,
                student[i].math,student[i].physics,
                student[i].english,student[i].computer,
                student[i].sumscore,student[i].avescore);
        }
    }
    fclose(outfp);
}


/* 函数功能：按学号修改学生记录信息 */
void modify_data()
{
    int i,number;       /* 定义下标变量i，学号变量number */
    printf("\t请输入要修改数据的学号，输入-1停止修改\n\t");
    scanf("%d",&number);
     /* 若输入的学号为-1，则停止修改记录并返回主函数 */
    while(number!=-1)
    {
        for(i=0;i<LENGTH;i++)
        {
            /*如果第i个元素的学号等于输入学号，则修改第i条记录的其他数据 */
            if(student[i].num==number)
```

```
                {
                    printf("\t原记录信息如下\n");
                    printf("\t%d\t%s\t%.2f\t\t%.2f\t\t%.
                            2f\t%.2f\t%.2f\t%.2f\n",
                        student[i].num,student[i].name,
                        student[i].math,student[i].physics,
                        student[i].english,student[i].computer,
                        student[i].sumscore,student[i].avescore);
                    printf("\n\t【请重新输入该条记录信息，学号不能修改，
                                数据之间以Tab键分隔】\n\n");
                    printf("\t学号\t姓名\t\t高等数学\t大学物理\t英语
                            \t计算机\n\t%d\t",student[i].num);
                    scanf("%s",&student[i].name);printf("\t\t");
                    scanf("%f",&student[i].math);printf("\t");
                    scanf("%f",&student[i].physics);printf("\t");
                    scanf("%f",&student[i].english);printf("\t");
                    scanf("%f",&student[i].computer);
                    student[i].sumscore=student[i].
                    math+student[i].physics
                        +student[i].english+student[i].computer;
                student[i].avescore=student[i].sumscore/4;
            }
        }
        printf("\n");
        printf("\t请输入要修改数据的学号，输入-1停止修改\n\t学号：");
        scanf("%d",&number);
    }
}


/* 函数功能：按学号删除学生记录信息 */
void del_data()
{
    int i,number,j;          /* 定义下标变量i、j，学号变量number */
    printf("\t输入学号，可以删除学号、姓名及各科成绩\n\t学号：");
    scanf("%d",&number);
    while(number!=-1)        /* 若输入的学号为-1,则停止删除记录并返回主函数*/
    {
        for(i=0;i<LENGTH;i++)
        {
            if(student[i].num==number)
            {
            /* 将数组中的元素依次前移，覆盖被删除的记录，达到删除的目的 */
                for(j=i;j<LENGTH;j++)
                {
                    student[j]=student[j+1];
                }
```

```
                    LENGTH=LENGTH-1;
                }

            }
            printf("\t请输入要删除的学号，输入-1停止删除\n\t学号：");
            scanf("%d",&number);
        }
    }
    /* 函数功能：保存最新的学生数据记录到磁盘文件 */
    void save_data()
    {
        char ch;              /* 存放是否保存的标志信息 */
        int i;                /* 定义下标变量i */
        FILE *fp;             /* 定义文件指针 */
        printf("\t当前的所有数据信息是否保存到磁盘文件stuin_new.txt,Y/N?:");
        getchar();
        scanf("%c",&ch);
        /* 输入为y或者Y，将当前数组中的最新信息保存到磁盘文件stuin_new.txt中 */
        if(ch=='Y'||ch=='y')
        {
            /* 创建并打开新的数据文件 */
            fp=fopen("..\\材成141张晓雪\\stuin_new.txt","w");
            for(i=0;i<LENGTH;i++)       /* 向磁盘文件写入数据 */
            {
                fprintf(fp,"%d\t%s\t%.2f\t%.2f\t%.2f\t%.
                        2f\t%.2f\t%.2f\n",
                    student[i].num,student[i].name,
                    student[i].math,student[i].physics,
                    student[i].english,student[i].computer,
                    student[i].sumscore,student[i].avescore);
            }
            fclose(fp);             /* 关闭文件 */
            printf("\t数据已保存至指定的磁盘文件stuin_new.txt\n");
        }
        else
            printf("\t您的最新数据未保存至指定的磁盘文件stuin_new.txt\n");
    }
```

5. 代码调试

代码的调试采用逐个函数调试的方法，根据详细设计方案，首先写出代码框架，然后逐个完善每一个函数的具体代码。调试从主函数 main()开始，调用菜单函数 menu()，调试成功后再从主函数开始，逐个调试实现输入、追加、浏览、统计、查询、修改、删除、保存等功能的各个函数。函数都调试成功后，再进行整体功能的测试，编制一个较大的数据文件，依次测试各系统功能是否能够实现。调试过程中的一些具体问题和调试方法的细节参见第 4 章。

2.2.3 程序运行结果

1. 测试数据

编制一个学生成绩管理系统测试数据集，如表 2-1 所示，将表中数据存入一个文本文件 stuin.txt，存放在磁盘上，作为初始数据文件，导入程序系统，测试各模块功能。

表 2-1 学生成绩管理系统测试数据表

学　号	姓　名	高等数学	大学物理	英　语	计算机
1001	stuname1	89	56	78	96
1002	stuname2	55	45	67	88
1003	stuname3	60	87	89	78
1004	stuname4	54	67	78	98
1005	stuname5	53	76	87	99
1006	stuname6	38	78	34	67
1007	stuname7	55	67	87	45
1008	stuname8	49	99	87	98
1009	stuname9	53	45	78	98
1010	stuname10	12	78	98	54
1011	stuname11	53	89	67	59
1012	stuname12	94	87	78	69
1013	stuname13	49	67	96	87
1014	stuname14	28	45	67	78
1015	stuname15	79	67	78	89
1016	stuname16	39	78	82	81
1017	stuname17	96	71	76	82
1018	stuname18	35	76	87	98
1019	stuname19	31	34	65	86
1020	stuname20	51	76	78	98
1021	stuname21	82	76	87	99
1022	stuname22	53	45	65	76
1023	stuname23	58	63	66	67
1024	stuname24	10	84	88	86
1025	stuname25	57	83	82	84
1026	stuname26	58	89	84	81
1027	stuname27	76	71	76	82
1028	stuname28	54	63	66	67
1029	stuname29	60	89	81	82
1030	stuname30	17	71	72	73
1031	stuname31	39	81	83	85
1032	stuname32	73	71	78	79

续表

学　号	姓　名	高等数学	大学物理	英　语	计　算　机
1033	stuname33	74	77	72	73
1034	stuname34	10	74	79	82
1035	stuname35	65	87	89	99
1036	stuname36	64	89	81	82
1037	stuname37	19	78	71	79
1038	stuname38	51	71	72	73
1039	stuname39	6	45	44	65
1040	stuname40	10	76	87	23
1041	stuname41	21	76	65	83
1042	stuname42	92	98	97	99
1043	stuname43	54	34	65	76
1044	stuname44	62	24	65	76
1045	stuname45	25	65	87	88
1046	stuname46	66	65	77	88
1047	stuname47	73	76	87	99
1048	stuname48	88	67	87	45
1049	stuname49	79	76	87	99
1050	stuname50	98	98	99	56

2. 运行结果

1）菜单界面如图 2-15 所示。

图 2-15　菜单界面

2）数据输入界面如图 2-16 所示。

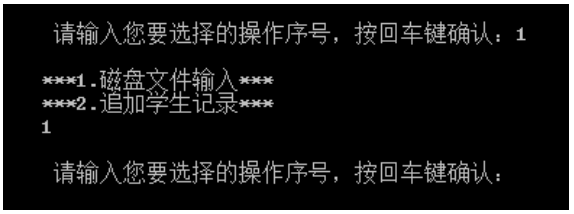


图 2-16　数据输入界面

3）数据浏览界面如图 2-17 所示。

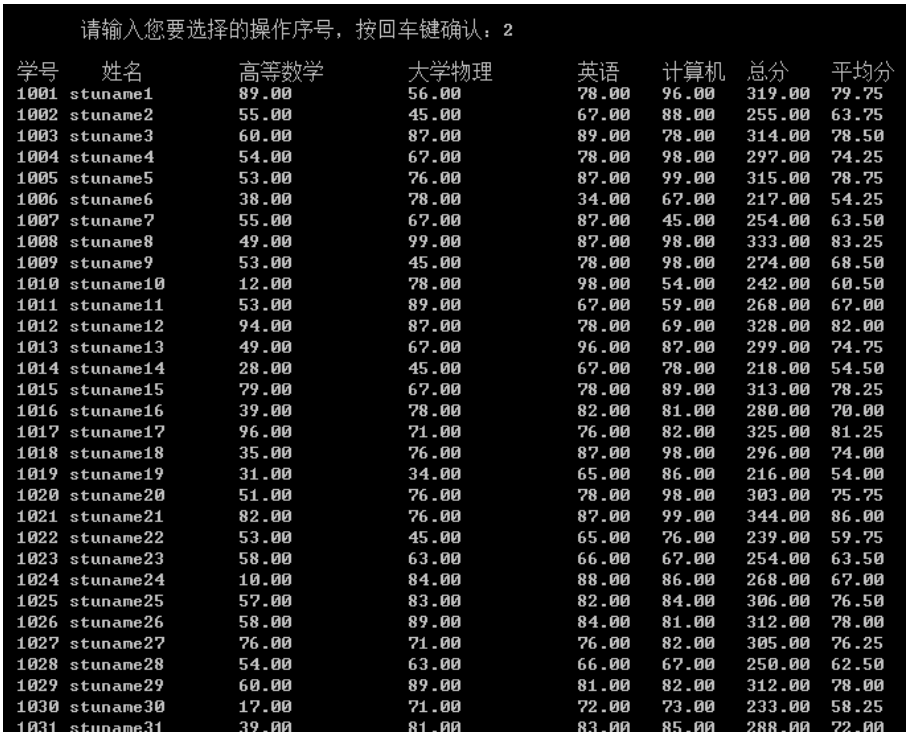


图 2-17　数据浏览界面

4）数据追加界面如图 2-18 所示。

图 2-18　数据追加界面

5）数据统计界面如图 2-19 所示。

```
请输入您要选择的操作序号，按回车键确认：3

平均成绩各分数段的统计结果如下：
平均成绩优秀人数为：2          ，优秀率为：2%
平均成绩良好人数为：8          ，良好率为：8%
平均成绩中等人数为：19         ，中等率为：19%
平均成绩及格人数为：13         ，及格率为：13%
平均成绩不及格人数为：9        ，不及格率为：9%

请输入您要选择的操作序号，按回车键确认：
```

图 2-19　数据统计界面

6）数据按学号查询界面如图 2-20 所示。

```
请输入您要选择的操作序号，按回车键确认：4

***1.按学号查询***
***2.按成绩查询***
1
请输入要查询的学号,输入-1停止查询
学号：1040
学号    姓名          高等数学        大学物理        英语    计算机  总分    平均分
1040    stuname40     10.00           76.00           87.00   23.00   196.00  49.00
请输入要查询的学号,输入-1停止查询
学号：1045
学号    姓名          高等数学        大学物理        英语    计算机  总分    平均分
1045    stuname45     25.00           65.00           87.00   88.00   265.00  66.25
请输入要查询的学号,输入-1停止查询
学号：-1

请输入您要选择的操作序号，按回车键确认：
```

图 2-20　数据按学号查询界面

7）数据按成绩查询界面如图 2-21 所示。

```
请输入您要选择的操作序号，按回车键确认：4

***1.按学号查询***
***2.按成绩查询***
2

各科目平均成绩为优秀的学生有：
学号    姓名          高等数学        大学物理        英语    计算机  总分    平均分
1042    stuname42     92.00           98.00           97.00   99.00   386.00  96.50

各科目平均成绩为良好的学生有：
学号    姓名          高等数学        大学物理        英语    计算机  总分    平均分
1008    stuname8      49.00           99.00           87.00   98.00   333.00  83.25
1012    stuname12     94.00           87.00           78.00   69.00   328.00  82.00
1017    stuname17     96.00           71.00           76.00   82.00   325.00  81.25
1021    stuname21     82.00           76.00           87.00   99.00   344.00  86.00
1035    stuname35     65.00           87.00           89.00   99.00   340.00  85.00
1047    stuname47     73.00           76.00           87.00   99.00   335.00  83.75
1049    stuname49     79.00           76.00           87.00   99.00   341.00  85.25
1050    stuname50     98.00           98.00           99.00   56.00   351.00  87.75
```

图 2-21　数据按成绩查询界面

```
各科目平均成绩为中等的学生有:
学号    姓名          高等数学        大学物理        英语    计算机  总分    平均分
1001    stuname1      89.00           56.00           78.00   96.00   319.00  79.75
1003    stuname3      60.00           87.00           89.00   78.00   314.00  78.50
1004    stuname4      54.00           67.00           78.00   98.00   297.00  74.25
1005    stuname5      53.00           76.00           87.00   99.00   315.00  78.75
1013    stuname13     49.00           67.00           96.00   87.00   299.00  74.75
1015    stuname15     79.00           67.00           78.00   89.00   313.00  78.25
1016    stuname16     39.00           78.00           82.00   81.00   280.00  70.00
1018    stuname18     35.00           76.00           87.00   98.00   296.00  74.00
1020    stuname20     51.00           76.00           78.00   98.00   303.00  75.75
1025    stuname25     57.00           83.00           82.00   84.00   306.00  76.50
1026    stuname26     58.00           89.00           84.00   81.00   312.00  78.00
1027    stuname27     76.00           71.00           76.00   82.00   305.00  76.25
1029    stuname29     60.00           89.00           81.00   82.00   312.00  78.00
```

图 2-21　数据按成绩查询界面（续）

8）数据修改界面如图 2-22 所示。

```
 请输入您要选择的操作序号，按回车键确认：5

请输入要修改数据的学号,输入-1停止修改
1020
原记录信息如下
1020    stuname20       51.00           76.00           78.00   98.00   303.00  75.75

【请重新输入该条记录信息，学号不能修改，数据之间以Tab键分隔】

学号    姓名            高等数学        大学物理        英语    计算机
1020    stuname20       60              76              90      70

请输入要修改数据的学号,输入-1停止修改
学号：1045
原记录信息如下
1045    stuname45       25.00           65.00           87.00   88.00   265.00  66.25

【请重新输入该条记录信息，学号不能修改，数据之间以Tab键分隔】

学号    姓名            高等数学        大学物理        英语    计算机
1045    stuname45       70              89              90      85

请输入要修改数据的学号,输入-1停止修改
学号：-1

 请输入您要选择的操作序号，按回车键确认：
```

图 2-22　数据修改界面

9）数据删除界面如图 2-23 所示。

```
 请输入您要选择的操作序号，按回车键确认：6

输入学号，可以删除学号、姓名及各科成绩
学号：1001
请输入要删除的学号,输入-1停止删除
学号：
```

图 2-23　数据删除界面

10）数据保存界面如图 2-24 所示。

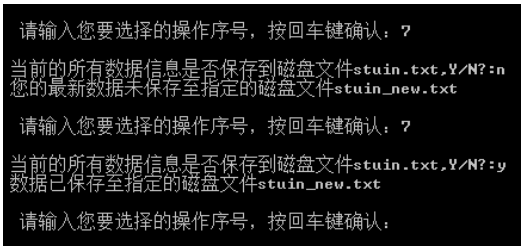


图 2-24　数据保存界面

11）退出系统界面如图 2-25 所示。

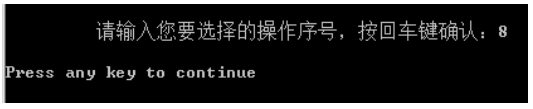


图 2-25　退出系统界面

3. 结果数据文件

学生成绩管理系统程序运行结束后，会在磁盘的指定位置自动生成一个结果数据文件 stuout.txt，如图 2-26 所示。

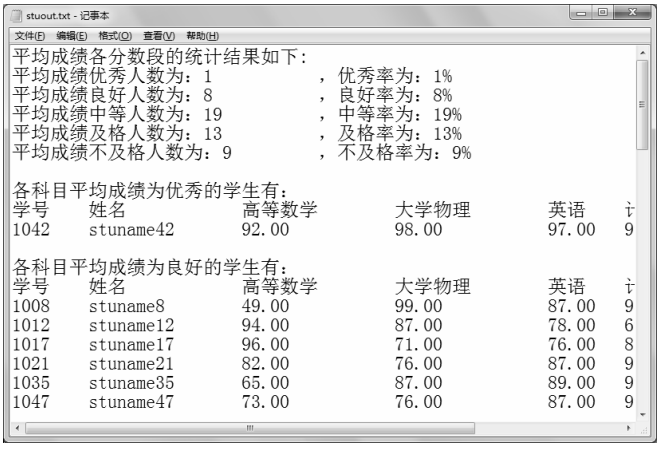


图 2-26　自动生成的数据

2.3　通讯录信息管理系统

综合运用所学程序设计语言的知识，设计一个通讯录信息管理系统，实现对通讯录中信息记录的输入、显示、追加、查询、修改、删除、保存等功能。

2.3.1　设计要求

1. 功能要求

（1）输入通讯录记录信息

通过一个数据文件向程序中输入记录信息。

(2) 显示通讯录记录信息

按姓名排序显示；

按所在城市排序显示；

按所在单位排序显示；

按自然顺序显示。

其中，自然顺序显示是指按通讯录输入数据时各条记录的先后顺序显示通讯录中已有的记录信息。

(3) 追加通讯录记录信息

通过键盘向程序追加记录信息，一次可以输入一条，也可以输入多条。一条记录信息是指通讯录中一个人员的完整信息。

(4) 查询通讯录记录信息

按姓名查询电话号码及其他信息；

按电话号码查询姓名及其他信息；

按性别查询电话号码及其他信息。

(5) 修改通讯录记录信息

按指定的姓名修改通讯录中的电话号码。

(6) 删除通讯录记录信息

按指定的姓名删除通讯录中对应记录，删除时以记录为单位，可一次删除一条记录，也可一次删除多条记录。

(7) 保存通讯录记录信息

将当前通讯录信息保存到指定的磁盘文件中。

(8) 退出通讯录信息管理系统

操作完成后，退出系统。

2. 技术要求

1) 每条通讯录记录包括如下信息：序号、姓名、性别、年龄、电话号码、所在城市、所在单位、住址。

2) 以菜单形式实现功能选择控制。

3) 对于通讯录中的记录输入、显示、查询、修改、删除等功能，要求编写独立的函数予以实现，其所属的各项功能尽量由独立的函数实现。

4) 退出通讯录信息管理系统之前，以磁盘文件的形式存储通讯录，存储位置、文件名、文件格式由设计者确定。

2.3.2 设计过程

1. 问题分析

通讯录信息管理系统需要处理的数据是通讯录信息记录，每条记录中涉及序号、姓名、性别、年龄、电话号码、所在城市、所在单位、住址等信息，其数据类型各不相同，既有

数值类型，又有字符类型，并且每条记录的各项信息是一个不可分割的整体，所以要考虑使用结构体数据类型存储一条信息记录，以结构体数组存储所有信息记录。对于要实现的所有功能要求，就是对数组的输入、输出、插入、查询、删除等有关数组的基本操作。

2. 总体设计

(1) 功能设计

依据上面的问题分析，可以将这个系统分解为：输入记录、显示记录、追加记录、查询记录、修改记录、删除记录、保存记录、退出系统共八大模块，如图 2-27 所示。

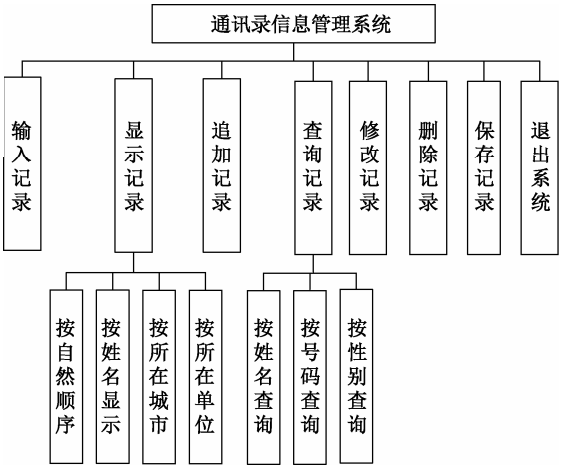


图 2-27　通讯录信息管理系统功能图

(2) 数据设计

根据问题分析，进行数据设计，定义一个包含序号、姓名、性别、年龄、电话号码、所在城市、所在单位、住址共八个成员的结构体类型，作为通讯录信息记录的存储类型；定义一个结构体类型的数组存放若干通讯录信息记录；定义一个符号常量 N 作为数组的长度；定义全局变量 COUNT 存放通讯录中当前实际存放的信息记录条数。

```
#define N  100;  /* N宏定义为通讯录记录条数   */
```

定义一个学生信息的结构体类型：

```
struct record
{
     int    num;            /*  序号  */
     char   name[20];       /*  姓名  */
     char   sex[3];         /*  性别  */
     int    age;            /*  年龄  */
     char   phone[13];      /*  电话号码  */
     char   city[20];       /*  所在城市  */
     char   units[30];      /*  所在单位  */
     char   address[20];    /*  住址  */
};
struct record addlist[N];  /*定义数组存放信息记录*/
int COUNT;                 /*当前实际记录条数*/
```

（3）函数设计

根据功能设计，为通讯录信息管理系统设计如下函数实现各模块的功能。

main()：函数的原型是 int main()，是通讯录信息管理系统的主函数，控制各功能函数模块的调用，实现整个程序的各项功能。

menu()：函数的原型是 void menu()，用于实现程序功能的主菜单显示。

menu_outfile()：函数的原型是 void menu_outfile()，用于实现将程序功能的主菜单输出到结果数据文件中。

load()：函数的原型是 int load()，用于实现将磁盘文件中的数据记录导入到程序中，实现数据信息的文件输入。

output()：函数的原型是 void output(struct record array[],int n)，用于实现数组中记录信息的显示输出，将记录信息显示在屏幕上，同时写入输出结果文件。

display()：函数的原型是 void display()，用于实现四种显示方式的菜单输出和显示方法的选择的主控模块。

display_init()：函数的原型是 void display_init()，用于使通讯录中的数据记录信息按输入时的原始自然顺序显示在屏幕上，同时写入输出结果数据文件。

display_name()：函数的原型是 void display_name()，用于使通讯录中的数据记录信息按姓名排序显示在屏幕上，同时写入输出结果数据文件。

display_city()：函数的原型是 void display_city()，用于使通讯录中的数据记录信息按所在城市排序显示在屏幕上，同时写入输出结果数据文件。

display_unit()：函数的原型是 void display_unit()，用于使通讯录中的数据记录信息按所在单位排序显示在屏幕上，同时写入输出结果数据文件。

append()：函数的原型是 void append()，用于向通讯录中追加信息记录，可以一次追加一条记录，也可以追加多条记录。

query()：函数的原型是 void query()，用于实现查询功能的主控模块。

query_name()：函数的原型是 void query_name()，用于按姓名查询电话号码，输入姓名即可查询到该姓名对应的电话号码。

query_tele()：函数的原型是 void query_tele()，用于按电话号码查询姓名，输入电话号码即可查询该电话号码对应的姓名。

query_sex()：函数的原型是 void query_sex()，用于按性别查询通讯录信息。

modify_record()：函数的原型是 void modify_record()，用于按姓名修改通讯录中的电话号码，输入姓名即可修改该姓名对应信息记录的电话号码。

del_record()：函数的原型是 void del_record()，用于按姓名删除通讯录中的信息记录，输入姓名即可将该姓名对应的信息记录从通讯录中删除。

save_record()：函数的原型是 void save_record()，用于将当前通讯录中的所有记录信息保存到一个新创建的磁盘数据文件中。

（4）函数调用关系

以上函数的相互调用关系如图 2-28 所示。

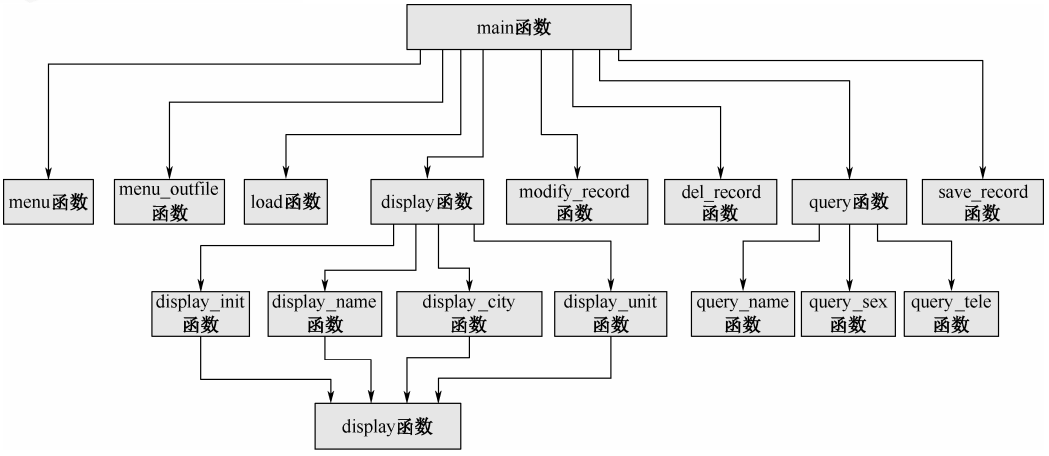


图 2-28　通讯录信息管理系统函数调用关系图

3. 详细设计

（1）main()函数

main()函数模块流程图如图 2-29 所示。

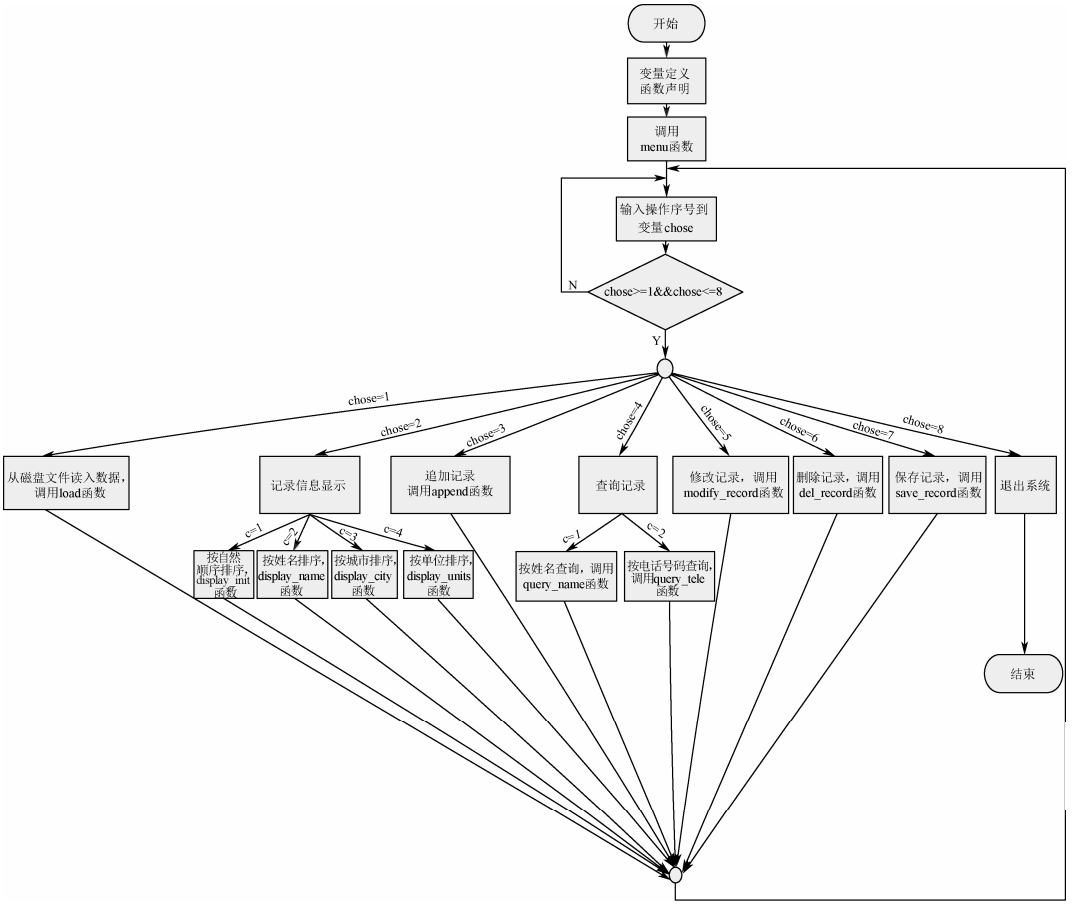


图 2-29　main()函数流程图

函数内部数据定义：

```
    int chose;              /*定义变量存放输入的菜单项*/
    FILE *outfp;            /* 定义文件指针 */
```

（2）load()函数

load()函数模块流程图如图 2-30 所示。

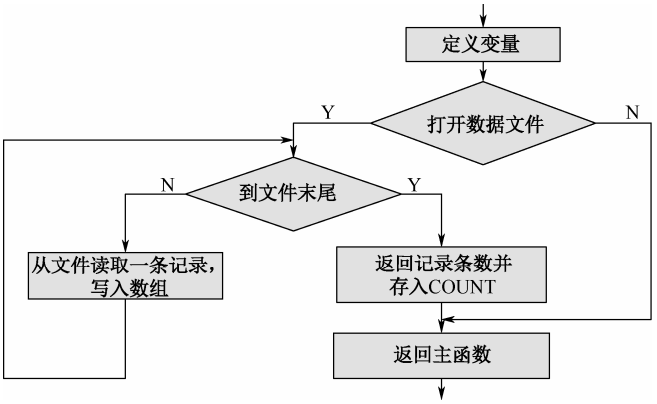


图 2-30　load()函数流程图

函数内部数据定义：

```
   int i;                   /* 定义下标变量i */
   FILE *infp;              /*定义文件指针，指向打开的数据文件*/
```

（3）output()函数

output()函数模块流程图如图 2-31 所示。

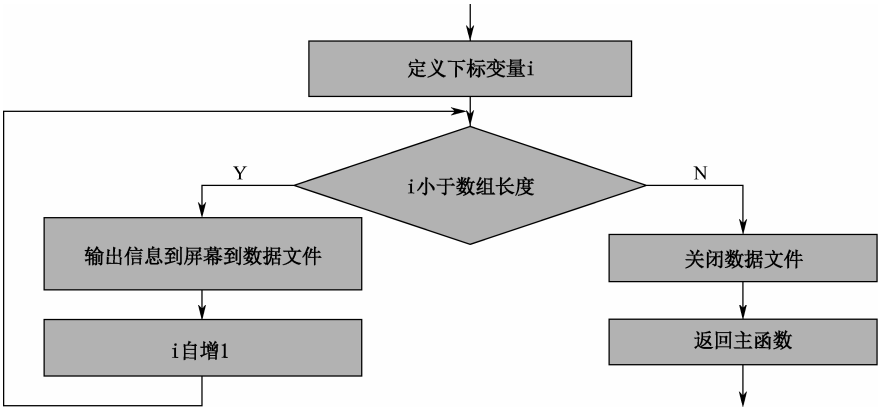


图 2-31　output()函数流程图

函数内部数据定义：

```
   int i;                   /*定义下标变量*/
   FILE *outfp;             /*定义文件指针*/
```

（4）display_name()函数

display_name()函数模块流程图如图 2-32 所示。

函数内部数据定义：

```
int min, i, j;                          /*定义下标变量和最小值下标变量*/
struct record addlist1[N], temp;        /*定义结构体数组和交换时要借助的变量*/
```

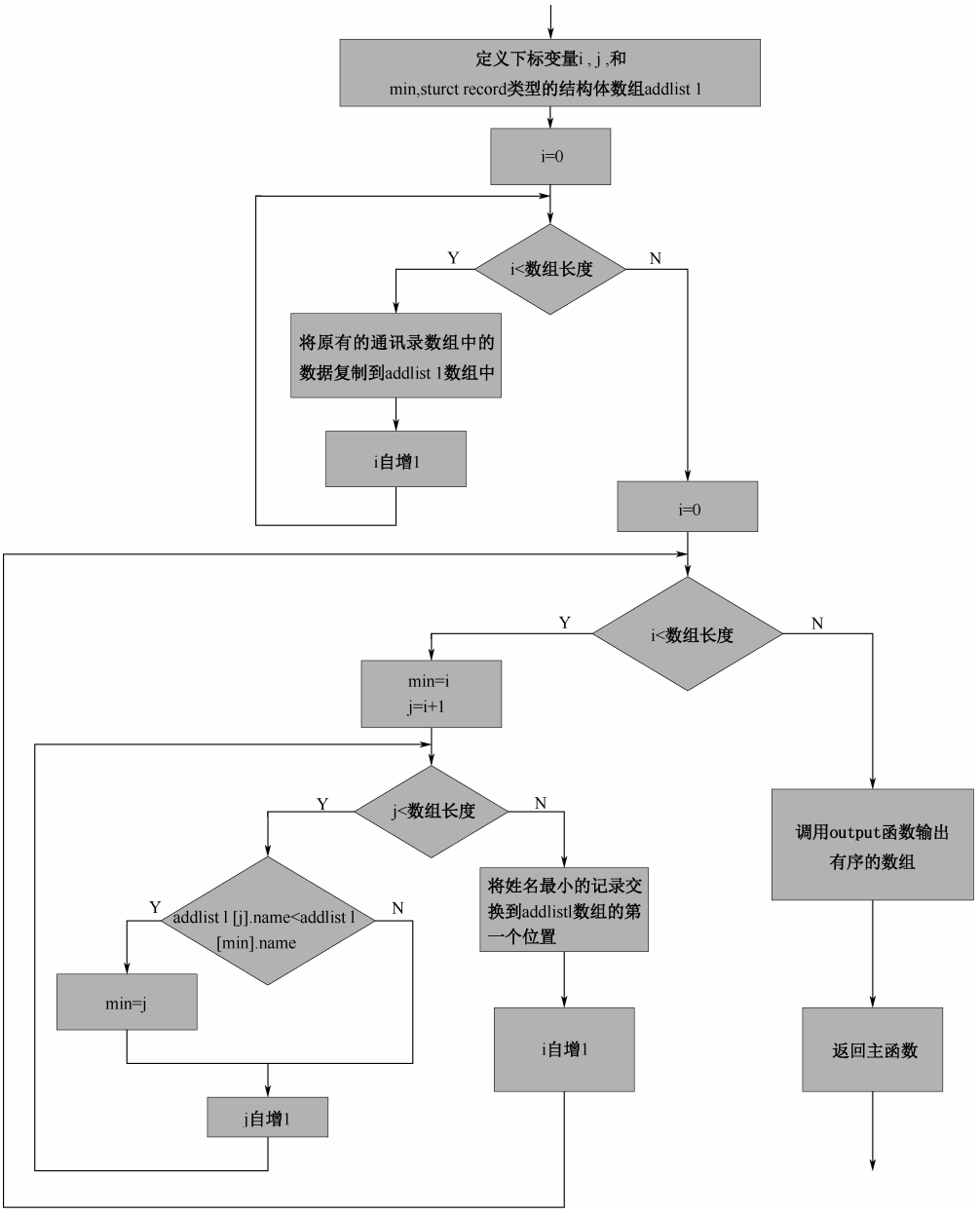


图 2-32　display_name()函数流程图

（5）append()函数

append()函数模块流程图如图 2-33 所示。

函数内部数据定义：

```
int i;              /*定义下标变量*/
char ask;           /*定义是否继续的询问变量*/
```

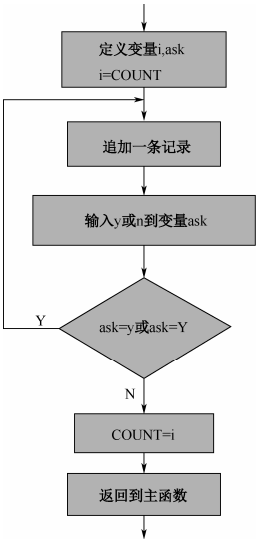


图 2-33　append()函数流程图

（6）query_name()函数

query_name()函数模块流程图如图 2-34 所示。

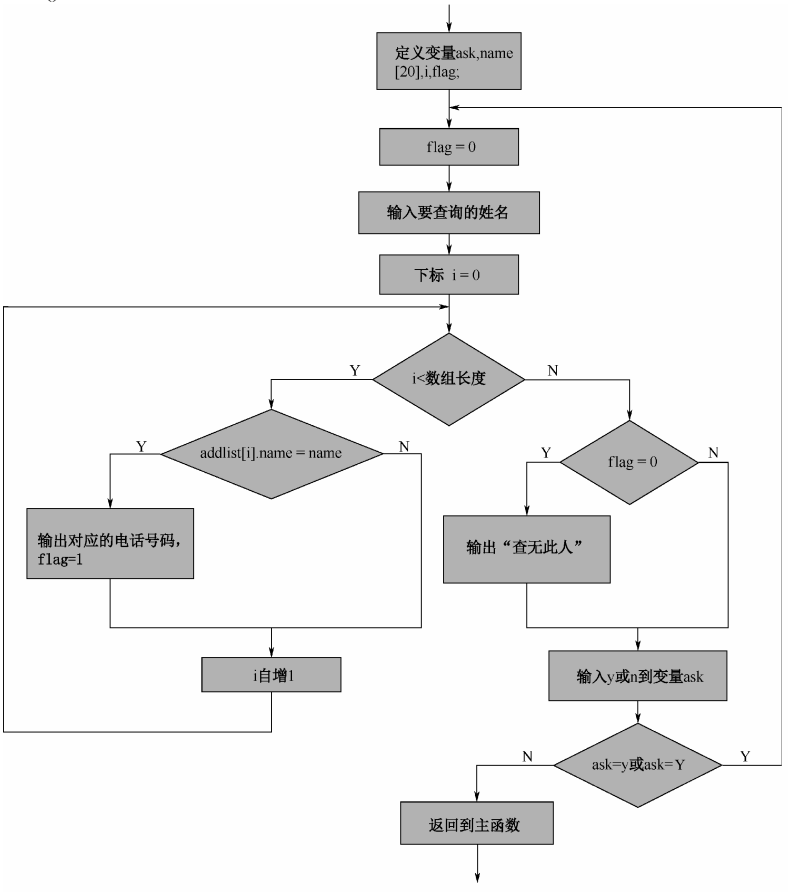


图 2-34　query_name()函数流程图

函数内部数据定义

```
char     ask, name[20];
int  i, flag;
FILE *outfp;
```

（7）modify_record()函数

modify_record()函数模块流程图如图 2-35 所示。

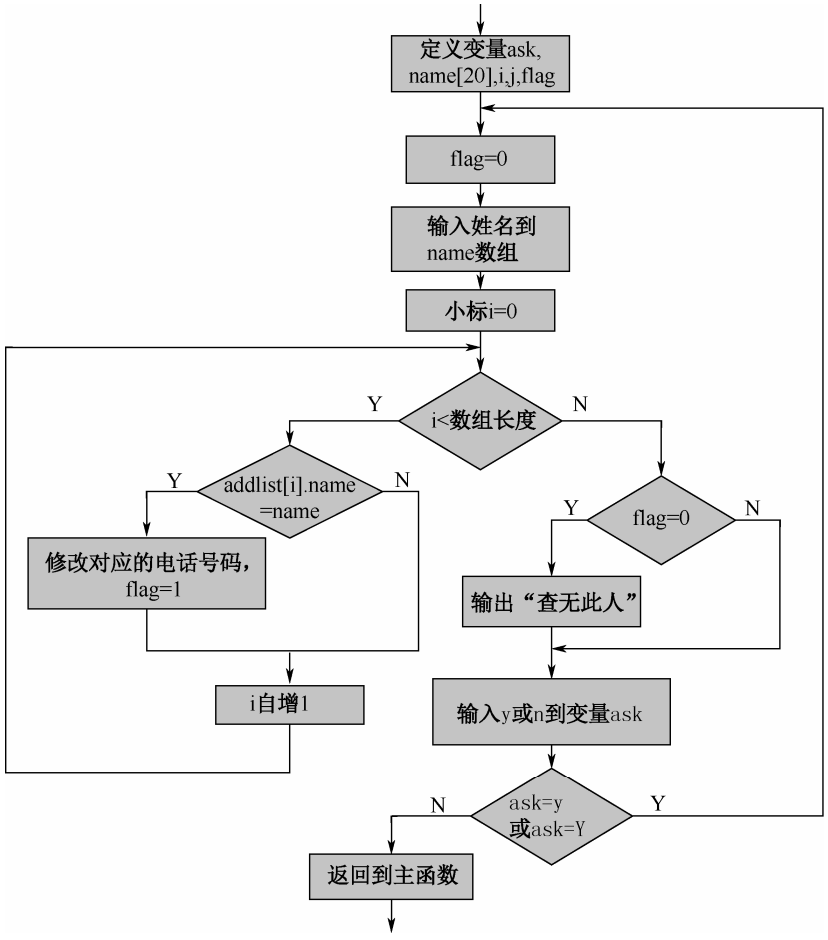


图 2-35　modify_record()函数流程图

函数内部数据定义：

```
char ask, name[20];  /*定义变量存放是否保存的标志信息，定义数组存放输入的姓名*/
int i, flag;         /*定义下标变量和标致变量*/
```

（8）del_record()

del_record()函数模块流程图如图 2-36 所示。

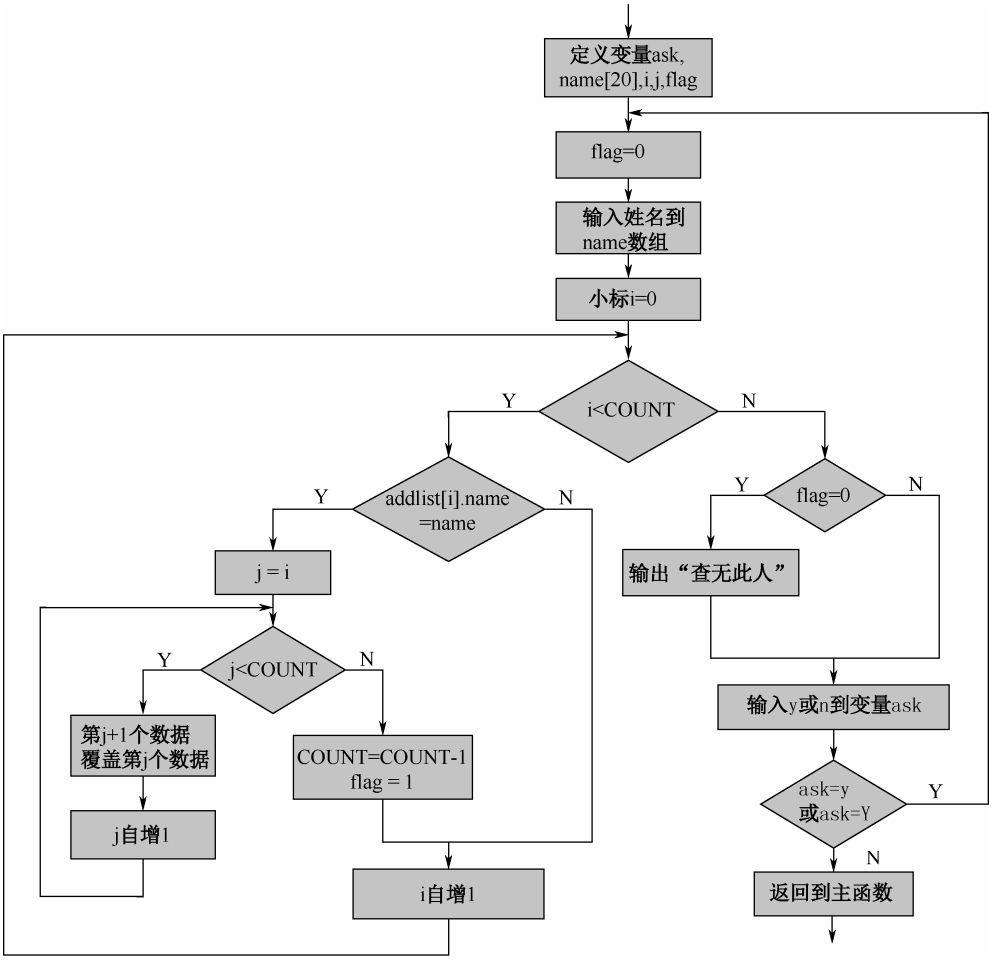


图 2-36　del_record()函数流程图

函数内部数据定义：

```
char ask, name[20];  /*定义变量存放是否保存的标志信息，定义数组存放输入的姓名*/
int i, j, flag;      /*定义下标变量和标致变量*/
```

4. 代码编写

（1）代码框架

```
#define N 30          /*通讯录的最大容量*/
#include"stdio.h"
#include"stdlib.h"
#include"conio.h"
#include"string.h"

struct record
{
     int   num;              /*  序号  */
     char  name[20];         /*  姓名  */
```

```
    int   age;              /*  年龄  */
    char  phone[13];        /*  电话号码  */
    char  city[20];         /*  所在城市  */
    char  units[30];        /*  所在单位  */
    char  address[20];      /*  住址  */
};
struct record addlist[N];   /*定义数组存放信息记录*/
int COUNT;                  /*当前实际记录条数*/


/* 函数功能：调用菜单函数，通过输入的菜单项序号，控制整个程序的运行 */
int main()
{

}

/* 函数功能：显示程序菜单项 */
void menu()
{

}

/* 函数功能：在输出结果文件中显示程序菜单项 */
void menu_outfile()
{

}

/* 函数功能：从磁盘文件中读取初始学生记录信息，送入程序的结构体数组 */
int load()
{

}

/*函数功能：实现数组中记录信息的显示输出，将记录信息显示在屏幕上，
          同时写入输出结果文件*/
void output(struct record array[],int n)
{

}

/*函数功能：显示功能主控模块*
void display()
{
```

```
}

/*函数功能：将通讯录中的数据记录信息按输入时的原始自然顺序显示在屏幕上，
          同时写入输出结果文件*/
void display_init()
{


}

/*函数功能：将通讯录中的数据记录信息按姓名排序显示在屏幕上，同时写入输出结果文件*/
void display_name()
{

}


/*函数功能：将通讯录中的数据记录信息按所在城市排序显示在屏幕上，
          同时写入输出结果文件*/
void display_city()
{

}

/*函数功能：将通讯录中的数据记录信息按所在单位排序显示在屏幕上，
          同时写入输出结果文件*/
void display_unit()
{

}

/*函数功能：向通讯录中追加信息记录，一次可以追加一条记录，也可以追加多条记录*/
void append()
{


}

/*函数功能：查询功能主控模块*
void  query()
{

}

/*函数功能：按姓名查询电话号码，输入姓名，查询该姓名对应的电话号码*/
void query_name()
{
```

```
}

/*函数功能：按电话号码查询姓名，输入电话号码，查询该电话号码对应的姓名*/
void query_tele()
{
}

/*函数功能：按性别查询通讯录中的记录*/
 void query_sex()
 {

 }
/*函数功能：按姓名修改通讯录中的电话号码*/
void modify_record()
{



}

/*函数功能：删除通讯录中的信息记录*/
void del_record()
{

}

/*函数功能：将当前通讯录中的所有记录信息保存到一个新创建的磁盘文件中*/
void save_record()
{

}
```

(2) 完整代码

```
#define N 30              /*通讯录的最大容量*/
#include"stdio.h"
#include"stdlib.h"
#include"conio.h"
#include"string.h"

struct record
{
     int    num;              /*序号*/
     char   name[20];         /*姓名*/
     char   sex[3];           /*性别*/
     int    age;              /*年龄*/
     char   phone[13];        /*电话号码*/
     char   city[20];         /*所在城市*/
```

```
    char   units[30];        /*所在单位*/
    char   address[20];      /*住址*/
};
struct record addlist[N];    /*定义数组存放信息记录*/
int COUNT;                   /*当前实际记录条数*/

/* 函数功能：调用菜单函数，通过输入的菜单项序号，控制整个程序的运行 */
int main()
{
    int chose;               /*定义变量存放输入的菜单选项*/
    FILE *outfp;
    void menu();
    void menu_outfile();
    int  load();
    void output(struct record array[],int n);
    void display();
    void display_init();
    void display_name();
    void display_city();
    void display_unit();
    void append();
    void query();
    void query_name();
    void query_tele();
    void query_sex();
    void modify_record();
    void del_record();
    void save_record();

    /* 在磁盘的指定位置创建一个结果数据文件，存放输出数据 */
    outfp=fopen("..\\材成141李兵\\addlistout.txt","w");
    fclose(outfp);                    /* 暂时关闭文件 */
    menu();
    menu_outfile();
    while(1)                          /* 遇到非法输入时，重新输入 */
    {
        printf("\n\t 请输入您要选择的操作序号，按回车键确认：");
        scanf("%d",&chose);           /* 输入菜单项序号 */
        printf("\n");
        if(chose>8||chose<1)          /* 判断输入的菜单项序号是否合法 */
            printf("------对不起，没 有 此 项 功 能 -----!\n");
        /* 通过输入的菜单项序号，调用相应的函数，执行对应的操作 */
        switch(chose)
        {
        case 1:   COUNT=load();     break;      /* 输入记录 */
        case 2:   display();            break;      /* 显示记录 */
        case 3:  append();              break;      /* 追加记录 */
```

```
            case 4:  query();                break;      /* 查询记录 */
            case 5:    modify_record(); break;           /* 修改记录 */
            case 6:    del_record();         break;      /* 删除记录 */
            case 7:    save_record();        break;      /* 保存记录*/
            case 8:    exit(0);                          /* 退出系统 */
            }
        }
        getch();
        return 0;
}

/* 函数功能：显示程序菜单项 */
void menu()
{
        printf("\n\n\n\n");

        printf("\t*************************************************n");
        printf("\t*************************************************n");
        printf("\t\t    ※ ☆ 欢 迎 光 临  ☆ ※\n");
        printf("\n");
        printf("\t      ★★通 讯 录 信 息 管 理 系 统 ★★ \n");
        printf("\n");
        printf("\t\t\t   ★1.输入记录★ \n");
        printf("\n");
        printf("\t\t\t   ★2.显示记录★ \n");
        printf("\n");
        printf("\t\t\t   ★3.追加记录★ \n");
        printf(" \n");
        printf("\t\t\t   ★4.查询记录★ \n");
        printf("\n");
        printf("\t\t\t   ★5.修改记录★ \n");
        printf("\n");
        printf("\t\t\t   ★6.删除记录★ \n");
        printf("\n");
        printf("\t\t\t   ★7.保存记录★ \n");
        printf("\n");
        printf("\t\t\t   ★8.退出系统★ \n");
        printf("\n\n");
        printf("\t*************************************************n");
        printf("\t执行其他操作前，请先执行操作1，载入数据\n");
        printf("\t*************************************************n");
}

/* 函数功能：在输出结果文件中显示程序菜单项 */
void menu_outfile()
{
        FILE *outfp;
```

```
    outfp=fopen("..\\材成141李兵\\addlistout.txt","a");
    fprintf(outfp,"\n\n\n\n");
    fprintf(outfp,"\t*****************************************n");
    fprintf(outfp,"\t*****************************************n");
    fprintf(outfp,"\t\t    ※ ☆ 欢 迎 光 临  ☆ ※\n");
    fprintf(outfp,"\n");
    fprintf(outfp,"\t      ★★通 讯 录 信 息 管 理 系 统 ★★ \n");
    fprintf(outfp,"\n");
    fprintf(outfp,"\t\t\t  ★1.输入记录★ \n");
    fprintf(outfp,"\n");
    fprintf(outfp,"\t\t\t  ★2.显示记录★ \n");
    fprintf(outfp,"\n");
    fprintf(outfp,"\t\t\t  ★3.追加记录★ \n");
    fprintf(outfp," \n");
    fprintf(outfp,"\t\t\t  ★4.查询记录★ \n");
    fprintf(outfp,"\n");
    fprintf(outfp,"\t\t\t  ★5.修改记录★ \n");
    fprintf(outfp,"\n");
    fprintf(outfp,"\t\t\t  ★6.删除记录★ \n");
    fprintf(outfp,"\n");
    fprintf(outfp,"\t\t\t  ★7.保存记录★ \n");
    fprintf(outfp,"\n");
    fprintf(outfp,"\t\t\t  ★8.退出系统★ \n");
    fprintf(outfp,"\n\n");
    fprintf(outfp,"\t*****************************************n");
    fprintf(outfp,"\t执行其他操作前，请先执行操作1，载入数据\n");
    fprintf(outfp,"\t*****************************************n");
    fclose(outfp);      /* 暂时关闭文件 */
}

/* 函数功能：从磁盘文件中读取初始学生记录信息，送入程序的结构体数组 */
int load()
{
    int i;              /*定义下标变量i */
    FILE *infp;         /*定义文件指针，指向打开的数据文件*/
    /* 是否成功打开数据文件*/
    if((infp=fopen("..\\材成141李兵\\list.txt","r"))==NULL)
    {
        printf("can not open file stuin.txt!");
        getch();
        return -1;
    }
    i=0;
    while(!feof(infp))  /*文件指针未到文件尾，继续读入数据 */
    {
        fscanf(infp,"%d%s%s%d%s%s%s%s",
            &addlist[i].num,addlist[i].name,
```

```
                addlist[i].sex,&addlist[i].age,
                addlist[i].phone,addlist[i].city,
                addlist[i].units,addlist[i].address);
            i=i+1;
        }
        fclose(infp);
        return i;
    }

    /*函数功能：实现数组中记录信息的显示输出*/
    void output(struct record array[],int n)
    {
        int i;                      /*定义下标变量*/
        FILE *outfp;                /*定义文件指针*/
        /*以追加的方式打开结果数据文件*/
        outfp=fopen("..\\材成141李兵\\addlistout.txt","a");
        printf("\t序号\t姓名\t性别\t年龄\t电话号码\t所在城市\t所在单位\
                t住址\n");
        fprintf(outfp,"\t序号\t姓名\t年龄\t电话号码\t所在城市\t所在单位\
                t住址\n");
        for(i=0;i<COUNT;i++)    /*通过下标的变化，依次输出记录*/
        {
            printf("\t%d\t%s\t%s\t%d\t%s\t",array[i].num,
                array[i].name,array[i].sex,
                array[i].age,array[i].phone);
            printf("%s\t\t%s\t\t%s\n",array[i].city,
                array[i].units,array[i].address);
            fprintf(outfp,"\t%d\t%s\t%s\t%s\t%d\t%s\t",
                array[i].num,array[i].name,array[i].sex,
                array[i].sex,array[i].age,array[i].phone);
            fprintf(outfp,"%s\t\t%s\t\t%s\n",array[i].city,
                array[i].units,array[i].address);
        }
        fclose(outfp);
    }

    /*函数功能：显示主控模块*/
    void display()
    {
        int c;
        printf("\t\t***1.按自然顺序查询***\n\t");
        printf("\t***2.按姓名排序显示***\n\t");
        printf("\t***3.按城市排序显示***\n\t");
        printf("\t***4.按单位排序显示***\n\t");
        while(1)
        {
            printf("\n\t 请输入您要选择的操作序号1—4,按回车键确认：");
```

```
            scanf("%d", &c);
            if(c>4||c<1)
            {
                printf("------对不起，请输入1—4 -----!\n");
                continue;
            }
            else
            break;
            }
            switch(c)
            {
                case 1:display_init(); break;
                case 2:display_name(); break;
                case 3:display_city(); break;
                case 4:display_unit(); break;      /* 显示记录信息 */
            }
    }

    /*函数功能：将通讯录中的数据记录信息按输入时的原始自然顺序显示出来*/
    void display_init()
    {
         FILE *outfp;
         outfp=fopen("..\\材成141李兵\\addlistout.txt","a");
         printf("\n\t按输入的自然顺序显示通讯录\n");
         fprintf(outfp,"\n\t按输入的自然顺序显示通讯录\n");
         fclose(outfp);
         output(addlist,COUNT);

    }

    /*函数功能：使通讯录中的数据记录信息按姓名排序显示*/
    void display_name()
    {
         int min,i,j;                        /*定义下标变量和最小值下标变量*/
         struct record addlist1[N],temp;/*定义结构体数组和交换时要借助的变量*/
         FILE *outfp;
         outfp=fopen("..\\材成141李兵\\addlistout.txt","a");
         printf("\n\t按姓名排序显示通讯录\n");
         fprintf(outfp,"\n\t按姓名排序显示通讯录\n");
         fclose(outfp);
         for(i=0;i<COUNT;i++)                /*将记录复制到临时数组中*/
         {
             addlist1[i]=addlist[i];
         }
         /*采用选择法排序，使临时数组中的信息按姓名排序*/
         for(i=0;i<COUNT;i++)
         {
```

```
            min=i;
            for(j=i+1;j<COUNT;j++)
            {
                if(strcmp(addlist1[j].name,addlist1[min].name)==-1)
                {
                    min=j;
                }
            }
            temp=addlist1[i];
            addlist1[i]=addlist1[min];
            addlist1[min]=temp;
        }
        output(addlist1,COUNT);     /*调用输出函数输出排好序的记录信息*/
    }


    /*函数功能：通讯录中的数据记录信息按所在城市排序显示*/
    void display_city()
    {
        int min,i,j;                        /*定义下标变量和最小值下标变量*/
        struct record addlist1[N],temp; /*定义结构体数组和交换时要借助的变量*/
        FILE *outfp;
        outfp=fopen("..\\材成141李兵\\addlistout.txt","a");
        printf("\n\t按所在城市排序显示通讯录\n");
        fprintf(outfp,"\n\t按所在城市排序显示通讯录\n");
        fclose(outfp);
        for(i=0;i<COUNT;i++)                /*将记录复制到临时数组中*/
        {
            addlist1[i]=addlist[i];
        }
        /*采用选择法排序，使临时数组中的信息按所在城市排序*/
        for(i=0;i<COUNT;i++)
        {
            min=i;
            for(j=i+1;j<COUNT;j++)
            {
                if(strcmp(addlist1[j].city,addlist1[min].city)==-1)
                {
                    min=j;
                }
            }
            temp=addlist1[i];
            addlist1[i]=addlist1[min];
            addlist1[min]=temp;
        }
        output(addlist1,COUNT);     /*调用输出函数输出排好序的记录信息*/
    }
```

```
/*函数功能：使通讯录中的数据记录信息按所在单位排序显示*/
void display_unit()
{
     int min,i,j;
     struct record addlist1[N],temp;
     FILE *outfp;
     outfp=fopen("..\\材成141李兵\\addlistout.txt","a");
     printf("\n\t按所在单位排序显示通讯录\n");
     fprintf(outfp,"\n\t按所在单位排序显示通讯录\n");
     fclose(outfp);
     for(i=0;i<COUNT;i++)
     {
         addlist1[i]=addlist[i];
     }
     for(i=0;i<COUNT;i++)
     {
         min=i;
         for(j=i+1;j<COUNT;j++)
         {
             if(strcmp(addlist1[j].units,addlist1[min].units)==-1)
             {
                 min=j;
             }
         }
         temp=addlist1[i];
         addlist1[i]=addlist1[min];
         addlist1[min]=temp;
     }
     output(addlist1,COUNT);
}

/*函数功能：向通讯录中追加信息记录，一次可追加一条或多条记录*/
void append()
{
     int i;          /*定义下标变量*/
     char ask;       /*定义是否继续的询问变量*/
     i=COUNT;
     while(1)        /*用循环实现一次追加一条或多条记录*/
     {
         printf("\t请依次输入，用Tab键分隔数据\n\t");
         printf("序号\t姓名\t性别\t年龄\t电话号码\t所在城市\t所在单位
                 \t住址\n\t");
         scanf("%d%s%s%d%s%s%s%s",&addlist[i].num,addlist[i].name,
                   addlist[i].sex,
              &addlist[i].age,addlist[i].phone,addlist[i].city,
               addlist[i].units,
```

```
                addlist[i].address);
            i++;
            printf("继续录入记录吗？(Y/N)");
            getchar();
            scanf("%c",&ask);
            if(ask=='n'||ask=='N')
                break;

        }
        COUNT=i;
}

/*函数功能：查询模块*/
void query()
{
        int c;
        printf("\t***1.按姓名查询***\n");
        printf("\t***2.按电话号码***\n");
        printf("\t***3.按性别查询***\n\t");
        while(1)
        {
            scanf("%d", &c);
            if(c>3||c<1)
                printf("------对不起，请输入1或者2 -----!\n");
            else
                break;
        }
        switch(c)
        {
            case 1: query_name();   break;      /* 按姓名查询通讯录 */
            case 2: query_tele();   break;      /* 按电话号码查询通讯录 */
            case 3: query_sex();    break;      /* 按性别查询通讯录 */
        }

}

/*函数功能：按姓名查询电话号码及其他信息*/
void query_name()
{
        char ask,name[20];
        int i,flag;
        FILE *outfp;
        outfp=fopen("..\\材成141李兵\\addlistout.txt","a");
        printf("\n\t输入姓名，查询通讯录\n");
        fprintf(outfp,"\n\t输入姓名，查询通讯录\n");
        /*采用循环实现一次查询一条或者多条记录功能*/
        while(1)
```

```
        {
            flag=0;
            printf("\n\t请输入\n\t姓名:");
            fprintf(outfp,"\n\t请输入\n\t姓名:");
            scanf("%s",name);
            fprintf(outfp,"%s",name);   /*将查询的姓名输出到结果文件中*/
            printf("\t序号\t姓名\t性别\t年龄\t电话号码\t所在城市\
                    t所在单位\t住址\n");
            fprintf(outfp,"\t序号\t姓名\t性别\t年龄\t电话号码\t所在城市\
                        t所在单位\t住址\n");
            for(i=0;i<COUNT;i++)        /*通过下标的变化检查整个数组*/
                if(strcmp(addlist[i].name,name)==0)
                {
                    /*找到查询姓名，输出对应信息*/
                    printf("\t%d\t%s\t%s\t%d\t%s\t",addlist[i].num,
                        addlist[i].name,addlist[i].sex,
                        addlist[i].age,addlist[i].phone);
                    printf("%s\t\t%s\t\t%s\n",addlist[i].city,
                        addlist[i].units,addlist[i].address);
                    fprintf(outfp,"\t%d\t%s\t%s\t%s\t%d\t%s\t",
                        addlist[i].num,addlist[i].name,addlist[i].sex,
                        addlist[i].sex,addlist[i].age,addlist[i].phone);
                    fprintf(outfp,"%s\t\t%s\t\t%s\n",addlist[i].city,
                        addlist[i].units,addlist[i].address);
                    flag=1;
                }
                if(flag==0)   /*当未找到要查询的姓名时，输出提示信息*/
                {
                    printf("\t查无此人\n\n");
                fprintf(outfp,"\t查无此人\n\n");
            }
            printf("\t继续查询吗？(Y/N)");
            fprintf(outfp,"\t继续查询吗？(Y/N)");
            getchar();
            scanf("%c",&ask);
            fprintf(outfp,"%c",ask);
            if(ask=='n'||ask=='N')
                break;
        }
        fclose(outfp);
    }

    /*函数功能：按电话号码查询姓名及其他信息*/
    void query_tele()
    {
        char ask,phone[13];
        int i,flag;
```

```
FILE *outfp;
outfp=fopen("..\\材成141李兵\\addlistout.txt","a");
printf("\n\t输入电话号码，查询通讯录\n");
fprintf(outfp,"\n\t输入电话号码，查询通讯录\n");
/*采用循环实现一次查询一条或者多条记录功能*/
while(1)
{
    flag=0;
    printf("\t请输入\n\t电话号码:");
    fprintf(outfp,"\t请输入\n\t电话号码:");
    scanf("%s",phone);
    fprintf(outfp,"%s",phone);
    printf("\t序号\t姓名\t性别\t年龄\t电话号码\t所在城市\
            t所在单位\t住址\n");
    fprintf(outfp,"\t序号\t姓名\t性别\t年龄\t电话号码\
                    t所在城市\t所在单位\t住址\n");
    for(i=0;i<COUNT;i++)        /*通过下标的变化检查整个数组*/
        if(strcmp(addlist[i].phone,phone)==0)
        {
            /*找到查询号码*/
                printf("\t%d\t%s\t%s\t%d\t%s\t",addlist[i].num,
                     addlist[i].name,addlist[i].sex,
                     addlist[i].age,addlist[i].phone);
                printf("%s\t\t%s\t\t%s\n",addlist[i].city,
                     addlist[i].units,addlist[i].address);
                fprintf(outfp,"\t%d\t%s\t%s\t%s\t%d\t%s\t",
                     addlist[i].num,addlist[i].name,
                     addlist[i].sex,addlist[i].sex,addlist[i]
                               .age,
                     addlist[i].phone);
            fprintf(outfp,"%s\t\t%s\t\t%s\n",addlist[i].city,
                        addlist[i].units,addlist[i].address);
            flag=1;
        }
    if(flag==0)     /*当未找到查询号码，输出提示信息*/
    {
        printf("\t查无此号码\n\n");
        fprintf(outfp,"\t查无此号码\n\n");
    }
    printf("\t继续查询吗？(Y/N)");
    fprintf(outfp,"\t继续查询吗？(Y/N)");
    getchar();
    scanf("%c",&ask);
    fprintf(outfp,"%c",ask);
    if(ask=='n'||ask=='N')
        break;
```

```
        }
        fclose(outfp);
    }

    /*函数功能：按性别查询通讯录中的记录*/
     void query_sex()
     {
          char ask,sex[3];
          int i,flag;
          FILE *outfp;
          outfp=fopen("..\\材成141李兵\\addlistout.txt","a");
          printf("\n\t输入性别，查询通讯录\n");
          fprintf(outfp,"\n\t输入性别，查询通讯录\n");
          /*采用循环实现一次查询一条或者多条记录功能*/
          while(1)
          {
              flag=0;
              printf("\t请输入\n\t性别:");
              fprintf(outfp,"\t请输入\n\t性别:");
             scanf("%s",sex);
             fprintf(outfp,"%s",sex);
             printf("\t序号\t姓名\t性别\t年龄\t电话号码\t所在城市\t所在单位\
                       t住址\n");
             fprintf(outfp,"\t序号\t姓名\t性别\t年龄\t电话号码\t所在城市\
                            t所在单位\t住址\n");
             for(i=0;i<COUNT;i++)        /*通过下标的变化检查整个数组*/
                 if(strcmp(addlist[i].sex,sex)==0)
                 {
                     printf("\t%d\t%s\t%s\t%d\t%s\t",addlist[i].num,
                         addlist[i].name,addlist[i].sex,
                         addlist[i].age,addlist[i].phone);
                     printf("%s\t\t%s\t\t%s\n",addlist[i].city,
                         addlist[i].units,addlist[i].address);
                     fprintf(outfp,"\t%d\t%s\t%s\t%s\t%d\t%s\t",
                         addlist[i].num,addlist[i].name,addlist[i].sex,
                         addlist[i].sex,addlist[i].age,
                         addlist[i].phone);
                     fprintf(outfp,"%s\t\t%s\t\t%s\n",addlist[i].city,
                        addlist[i].units,addlist[i].address);
                     flag=1;
                 }
             if(flag==0)     /*未找到此性别时，输出提示信息*/
             {
                 printf("\t查无此性别\n\n");
                 fprintf(outfp,"\t查无此性别\n\n");
             }
```

```
            printf("\t继续查询吗？(Y/N)");
            fprintf(outfp,"\t继续查询吗？(Y/N)");
            getchar();
            scanf("%c",&ask);
            fprintf(outfp,"%c",ask);
            if(ask=='n'||ask=='N')
                break;
        }
        fclose(outfp);

    }
/* 函数功能：按姓名修改通讯录中的电话号码 */
void modify_record()
{
/*定义变量，存放是否保存的标志信息；定义数组，存放输入的姓名*/
        char ask,name[20];
        int i,flag;          /*定义下标变量和标志变量*/
        printf("\n\t姓名，修改对应的电话号码\n");
        while(1)             /*采用循环实现一次修改一条或者多条记录功能*/
        {
            flag=0;
            printf("\t请输入\n\t姓名:");
            scanf("%s",name);

            for(i=0;i<COUNT;i++)                    /*检查整个数组*/
                if(strcmp(addlist[i].name,name)==0)
                                                    /*找到输入姓名对应的记录*/
                {
                    printf("\t原电话号码为:%s\t",addlist[i].phone);
                    printf("请输入新号码：");
                    scanf("%s",addlist[i].phone);  /*修改找到记录的电话号码*/
                    flag=1;
                }
            if(flag==0)          /*当未找到输入的姓名时，输出提示信息*/
            {
                printf("\t查无此人\n\n");
            }
            printf("\t继续修改吗？(Y/N)");
            getchar();
            scanf("%c",&ask);
            if(ask=='n'||ask=='N')
                break;
        }
}

/*函数功能：删除通讯录中的信息记录*/
```

```
void del_record()
{
     char ask,name[20];  /*定义变量存放是否保存的标志信息，
                           定义数组存放输入的姓名 */
     int i,j,flag;       /*定义下标变量和标志变量*/
     printf("\n\t姓名，删除对应的通讯录信息记录\n");
     while (1)    /*采用循环实现一次删除一条或者多条记录功能*/
     {
         flag=0;
         printf("\t请输入\n\t姓名:");
         scanf("%s",name);

         for(i=0;i<COUNT;i++)            /*检查整个数组*/
             if(strcmp(addlist[i].name,name)==0)
                                         /*找到输入姓名对应的记录*/
             {
                 for(j=i;j<COUNT;j++)    /*在数组中删除该记录*/
                 {
                     addlist[j]=addlist[j+1];     /*用后一条记录覆盖此记录*/
                 }
                 COUNT=COUNT-1;
                 flag=1;
             }
         if(flag==0)                     /*在数组中未找到输入的姓名*/
         {
             printf("\t查无此人\n\n");
         }
         printf("\t继续删除吗？(Y/N)");
         getchar();
         scanf("%c",&ask);
         if(ask=='n'||ask=='N')
             break;
     }
}

/*函数功能：将当前通讯录中的信息保存到一个新创建的磁盘文件中*/
void save_record()
{
     char ask;           /* 定义变量，存放是否保存的标志信息 */
     int i;              /* 定义下标变量i */
     FILE *fp;           /* 定义文件指针 */
     printf("\n\t当前的所有数据信息是否保存到磁盘文件list_new.txt,Y/N?:");
     getchar();
     scanf("%c",&ask);
 /* 输入为y或者Y，将当前数组中的最新信息保存到磁盘文件stuin_new.txt中 */
     if(ask=='Y'||ask=='y')
```

```
        {
    /* 创建并打开新的数据文件 */
            fp=fopen("..\\材成141李兵\\list_new.txt","w");
            for(i=0;i<COUNT;i++)                    /* 向磁盘文件写入数据 */
            {
                fprintf(fp,"%d\t%s\t%d\t%s\t",addlist[i].num,
                        addlist[i].name,
                        addlist[i].age,addlist[i].phone);
                fprintf(fp,"%s\t\t%s\t\t%s\n",addlist[i].city,
                        addlist[i].units, addlist[i].address);
            }
            fclose(fp);                 /* 关闭文件 */
            printf("\t数据已保存至指定的磁盘文件stuin_new.txt\n");
        }
        else
            printf("\t您的最新数据未保存至指定的磁盘文件stuin_new.txt\n");
    }
```

5. 代码调试

代码的调试采用逐个函数调试的方法，根据详细设计方案，首先写出代码框架，然后逐个完善每一个函数的具体代码。调试从主函数 main()开始，调用菜单函数 menu()，调试成功后再从主函数开始，逐个调试实现输入、显示、追加、查询、修改、删除、保存等功能的各个函数。函数都调试成功后，再进行整体功能的测试，编制一个较大的数据文件，依次测试各系统功能是否能够实现。调试过程中的一些具体问题和调试方法的细节参见第 4 章。

2.3.3 程序运行结果

1. 测试数据

编制一个通讯录信息管理系统的测试数据集，如表 2-2 所示，将表中数据保存为一个文本文件 list.txt 并存放在磁盘中，作为初始数据文件，导入程序系统，测试各模块的功能。

表 2-2 通讯录信息管理系统测试数据表

序 号	姓 名	性 别	年 龄	电话号码	所在城市	所在单位	住 址
1	李兵妞	女	20	13303791102	洛阳	小学	青岛路
2	林田野	男	26	13300090425	北京	中学	礼士路
3	刘兵	男	24	13603790426	洛阳	公安局	开元路
4	马鹏鹏	男	34	13803790427	洛阳	工商局	关林路
5	王欢欢	女	21	13903790428	洛阳	邮局	牡丹路
6	张小段	男	22	13303710429	郑州	铁路	二马路
7	边加斌	男	34	18703710430	郑州	小学	正义街
8	陈琳琳	女	25	13303710431	郑州	中学	文化路
9	孙展翅	男	20	13603710432	郑州	公安局	农业路

续表

序　号	姓　名	性　别	年　龄	电话号码	所在城市	所在单位	住　址
10	徐佳	女	21	15507550433	南京	工商局	中山路
11	朱康	男	24	13303790434	郑州	邮局	花园路
12	邓博	男	45	13303790435	郑州	铁路	管城街
13	郭雅莉	女	40	13303790436	郑州	小学	航海路
14	李思思	女	45	13703790437	上海	中学	南京路
15	马净文	男	43	13300290438	上海	公安局	仁爱路
16	张建伟	男	21	13303790439	上海	工商局	和平路
17	常元哲	男	22	13303790440	广州	邮局	北京路
18	林飞	男	34	13303490441	天津	铁路	复兴路
19	常哲玮	男	25	13303790442	洛阳	小学	中州路
20	高鹏	男	19	15803790443	南京	中学	二马路

2. 运行结果

1）菜单界面如图 2-37 所示。

图 2-37　菜单界面

2）按自然顺序显示通讯录如图 2-38 所示。

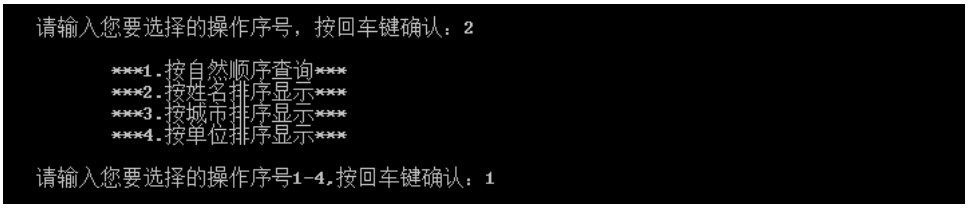
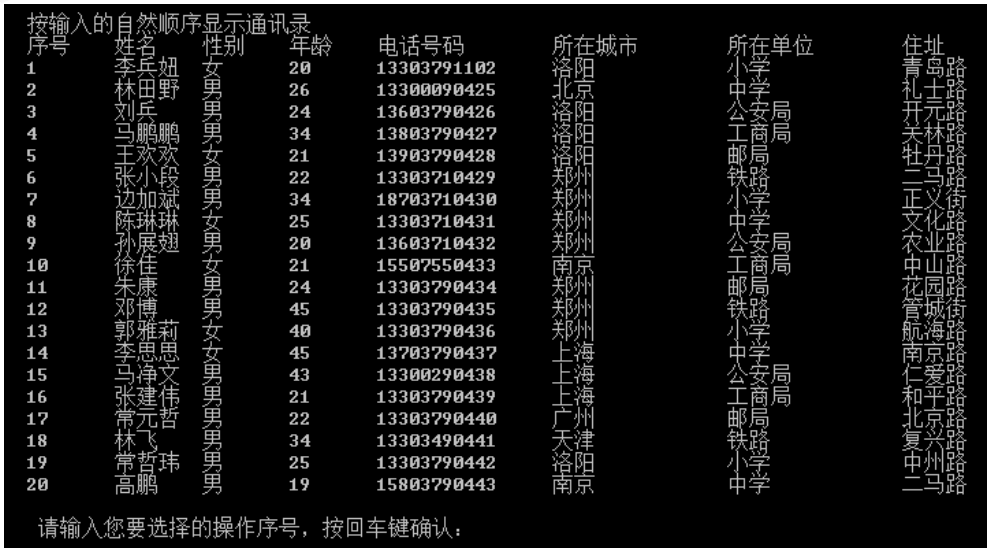


图 2-38　按自然顺序显示通讯录



按输入的自然顺序显示通讯录

序号	姓名	性别	年龄	电话号码	所在城市	所在单位	住址
1	李乐妞	女	20	13303791102	洛阳	小学	青岛路
2	林田野	男	26	13300090425	北京	中学	礼士路
3	刘兵	男	24	13603790426	洛阳	公安局	开元路
4	马鹏鹏	男	34	13803790427	洛阳	工商局	关林路
5	王欢欢	女	21	13903790428	洛阳	邮局	牡丹路
6	张小段	男	22	13303710429	郑州	铁路	二马路
7	边加斌	男	34	18703710430	郑州	小学	正义街
8	陈琳琳	女	25	13303710431	郑州	中学	文化路
9	孙展翅	男	20	13603710432	郑州	公安局	农业路
10	徐佳	女	21	15507550433	南京	工商局	中山路
11	朱康	男	24	13303790434	郑州	邮局	花园路
12	邓博	男	45	13303790435	郑州	铁路	管城街
13	郭雅莉	女	40	13303790436	郑州	小学	航海路
14	李思思	女	45	13703790437	上海	中学	南京路
15	马净文	男	43	13300290438	上海	公安局	仁爱路
16	张建伟	男	21	13303790439	上海	工商局	和平路
17	常元哲	男	22	13303790440	广州	邮局	北京路
18	林飞	男	34	13303490441	天津	铁路	复兴路
19	常哲玮	男	25	13303790442	洛阳	小学	中州路
20	高鹏	男	19	15803790443	南京	中学	二马路

请输入您要选择的操作序号，按回车键确认：

图 2-38　按自然顺序显示通讯录（续）

3）按姓名排序显示通讯录如图 2-39 所示。

请输入您要选择的操作序号，按回车键确认：2

1.按自然顺序查询
2.按姓名排序显示
3.按城市排序显示
4.按单位排序显示

请输入您要选择的操作序号1-4.按回车键确认：2

按姓名排序显示通讯录

序号	姓名	性别	年龄	电话号码	所在城市	所在单位	住址
7	边加斌	男	34	18703710430	郑州	小学	正义街
17	常元哲	男	22	13303790440	广州	邮局	北京路
19	常哲玮	男	25	13303790442	洛阳	小学	中州路
8	陈琳琳	女	25	13303710431	郑州	中学	文化路
12	邓博	男	45	13303790435	郑州	铁路	管城街
20	高鹏	男	19	15803790443	南京	中学	二马路
13	郭雅莉	女	40	13303790436	郑州	小学	航海路
1	李乐妞	女	20	13303791102	洛阳	小学	青岛路
14	李思思	女	45	13703790437	上海	中学	南京路
18	林飞	男	34	13303490441	天津	铁路	复兴路
2	林田野	男	26	13300090425	北京	中学	礼士路
3	刘兵	男	24	13603790426	洛阳	公安局	开元路
15	马净文	男	43	13300290438	上海	公安局	仁爱路
4	马鹏鹏	男	34	13803790427	洛阳	工商局	关林路
9	孙展翅	男	20	13603710432	郑州	公安局	农业路
5	王欢欢	女	21	13903790428	洛阳	邮局	牡丹路
10	徐佳	女	21	15507550433	南京	工商局	中山路
16	张建伟	男	21	13303790439	上海	工商局	和平路
6	张小段	男	22	13303710429	郑州	铁路	二马路
11	朱康	男	24	13303790434	郑州	邮局	花园路

请输入您要选择的操作序号，按回车键确认：

图 2-39　按姓名顺序显示通讯录

4）按所在城市排序显示通讯录如图 2-40 所示。

```
请输入您要选择的操作序号，按回车键确认：2

       ***1.按自然顺序查询***
       ***2.按姓名排序显示***
       ***3.按城市排序显示***
       ***4.按单位排序显示***

请输入您要选择的操作序号1-4,按回车键确认：3

按所在城市排序显示通讯录
序号    姓名    性别    年龄    电话号码        所在城市        所在单位        住址
2       林田野  男      26      13300090425     北京            中学            礼士路
17      常元哲  男      22      13303790440     广州            邮局            北京路
3       刘兵    男      24      13603790426     洛阳            公安局          开元路
4       马鹏鹏  男      34      13803790427     洛阳            工商局          关林路
5       王欢欢  女      21      13903790428     洛阳            邮局            牡丹路
1       李兵妞  女      20      13303791102     洛阳            小学            青岛路
19      常哲玮  男      25      13303790442     洛阳            小学            中州路
10      徐佳    女      21      15507550433     南京            工商局          中山路
20      高鹏    男      19      15803790443     南京            中学            二马路
14      李思思  女      45      13703790437     上海            中学            南京路
15      马淨文  男      43      13300290438     上海            公安局          仁爱路
16      张建伟  男      21      13303790439     上海            工商局          和平路
18      林飞    男      34      13303490441     天津            铁路            复兴路
8       陈琳琳  女      25      13303710431     郑州            中学            文化路
11      朱康    男      24      13303790434     郑州            邮局            花园路
12      邓博    男      45      13303790435     郑州            铁路            管城街
6       张小段  男      22      13303710429     郑州            铁路            二马路
13      郭雅莉  女      40      13303790436     郑州            小学            航海路
7       边加斌  男      34      18703710430     郑州            小学            正义街
9       孙展翅  男      20      13603710432     郑州            公安局          农业路

请输入您要选择的操作序号，按回车键确认：
```

图 2-40　按所在城市排序

5）按所在单位排序显示通讯录如图 2-41 所示。

```
请输入您要选择的操作序号，按回车键确认：2

       ***1.按自然顺序查询***
       ***2.按姓名排序显示***
       ***3.按城市排序显示***
       ***4.按单位排序显示***

请输入您要选择的操作序号1-4,按回车键确认：4

按所在单位排序显示通讯录
序号    姓名    性别    年龄    电话号码        所在城市        所在单位        住址
4       马鹏鹏  男      34      13803790427     洛阳            工商局          关林路
10      徐佳    女      21      15507550433     南京            工商局          中山路
16      张建伟  男      21      13303790439     上海            工商局          和平路
9       孙展翅  男      20      13603710432     郑州            公安局          农业路
15      马淨文  男      43      13300290438     上海            公安局          仁爱路
3       刘兵    男      24      13603790426     洛阳            公安局          开元路
12      邓博    男      45      13303790435     郑州            铁路            管城街
6       张小段  男      22      13303710429     郑州            铁路            二马路
18      林飞    男      34      13303490441     天津            铁路            复兴路
7       边加斌  男      34      18703710430     郑州            小学            正义街
13      郭雅莉  女      40      13303790436     郑州            小学            航海路
1       李兵妞  女      20      13303791102     洛阳            小学            青岛路
19      常哲玮  男      25      13303790442     洛阳            小学            中州路
5       王欢欢  女      21      13903790428     洛阳            邮局            牡丹路
17      常元哲  男      22      13303790440     广州            邮局            北京路
11      朱康    男      24      13303790434     郑州            邮局            花园路
14      李思思  女      45      13703790437     上海            中学            南京路
2       林田野  男      26      13300090425     北京            中学            礼士路
8       陈琳琳  女      25      13303710431     郑州            中学            文化路
20      高鹏    男      19      15803790443     南京            中学            二马路

请输入您要选择的操作序号，按回车键确认：
```

图 2-41　按所在单位排序

6）追加记录界面如图 2-42 所示。

```
 请输入您要选择的操作序号，按回车键确认：3

请依次输入,用Tab键分隔数据
序号    姓名    性别    年龄    电话号码        所在城市        所在单位        住址
21      张琳    女      22      15003752233     杭州            园林局          北京路
继续录入记录吗？(Y/N)y

请依次输入,用Tab键分隔数据
序号    姓名    性别    年龄    电话号码        所在城市        所在单位        住址
22      王明明  男      25      13807058978     深圳            教育局          南京路
继续录入记录吗？(Y/N)n

 请输入您要选择的操作序号，按回车键确认：
```

图 2-42　追加记录界面

7）按姓名查询通讯录如图 2-43 所示。

```
 请输入您要选择的操作序号，按回车键确认：4

***1.按姓名查询***
***2.按电话号码***
***3.按性别查询***
1

输入姓名，查询通讯录

请输入
姓名:张小段
序号    姓名    性别    年龄    电话号码        所在城市        所在单位        住址
6       张小段  男      22      13303710429     郑州            铁路            二马路
继续查询吗？(Y/N)y

请输入
姓名:朱康
序号    姓名    性别    年龄    电话号码        所在城市        所在单位        住址
11      朱康    男      24      13303790434     郑州            邮局            花园路
继续查询吗？(Y/N)n

 请输入您要选择的操作序号，按回车键确认：
```

图 2-43　按姓名查询通讯录

8）按电话号码查询通讯录如图 2-44 所示。

```
 请输入您要选择的操作序号，按回车键确认：4

***1.按姓名查询***
***2.按电话号码***
***3.按性别查询***
2

输入电话号码，查询通讯录
请输入
电话号码:13303790435
序号    姓名    性别    年龄    电话号码        所在城市        所在单位        住址
12      邓博    男      45      13303790435     郑州            铁路            管城街
继续查询吗？(Y/N)y
请输入
电话号码:13303791102
序号    姓名    性别    年龄    电话号码        所在城市        所在单位        住址
1       李兵妞  女      20      13303791102     洛阳            小学            青岛路
继续查询吗？(Y/N)y
请输入
电话号码:13078056677
序号    姓名    性别    年龄    电话号码        所在城市        所在单位        住址
查无此号码

继续查询吗？(Y/N)n

 请输入您要选择的操作序号，按回车键确认：
```

图 2-44　按电话号码查询通讯录

9）按性别查询通讯录如图 2-45 所示。

```
 请输入您要选择的操作序号，按回车键确认：4

***1.按姓名查询***
***2.按电话号码***
***3.按性别查询***
3

输入性别，查询通讯录
请输入
性别:男
序号    姓名    性别    年龄    电话号码        所在城市    所在单位    住址
2       林田野  男      26      13300090425     北京        中学        礼士路
3       刘兵    男      24      13603790426     洛阳        公安局      开元路
4       马鹏鹏  男      34      13803790427     洛阳        工商局      关林路
6       张小段  男      22      13303710429     郑州        铁路        二马路
7       边加斌  男      34      18703710430     郑州        小学        正义街
9       孙展翅  男      20      13603710432     郑州        公安局      农业路
11      朱康    男      24      13303790434     郑州        邮局        花园路
12      邓博    男      45      13303790435     郑州        铁路        管城街
15      马净文  男      43      13300290438     上海        公安局      仁爱路
16      张建伟  男      21      13303790439     上海        工商局      和平路
17      常元哲  男      22      13303790440     广州        邮局        北京路
18      林飞    男      34      13303490441     天津        铁路        复兴路
19      常哲玮  男      25      13303790442     洛阳        小学        中州路
20      高鹏    男      19      15803790443     南京        中学        二马路
继续查询吗？(Y/N)y
请输入
性别:女
序号    姓名    性别    年龄    电话号码        所在城市    所在单位    住址
1       李兵妞  女      20      13303791102     洛阳        小学        青岛路
5       王欢欢  女      21      13903790428     洛阳        邮局        牡丹路
8       陈琳琳  女      25      13303710431     郑州        中学        文化路
10      徐佳    女      21      15507550433     南京        工商局      中山路
13      郭雅莉  女      40      13303790436     郑州        小学        航海路
14      李思思  女      45      13703790437     上海        中学        南京路
21      张琳    女      21      15032450099     南阳        市政府      北京路
22      王珊珊  女      23      15869905656     海口        旅游局      梅林路
继续查询吗？(Y/N)n

 请输入您要选择的操作序号，按回车键确认：
```

图 2-45　按性别查询通讯录

10）按姓名修改电话号码界面如图 2-46 所示。

```
 请输入您要选择的操作序号，按回车键确认：5


姓名，修改对应的电话号码
请输入
姓名:林飞
原电话号码为:13303490441          请输入新号码：18783450708
继续修改吗？(Y/N)y
请输入
姓名:李婷婷
查无此人

继续修改吗？(Y/N)n

 请输入您要选择的操作序号，按回车键确认：
```

图 2-46　按姓名修改电话号码

11）删除信息界面如图 2-47 所示。

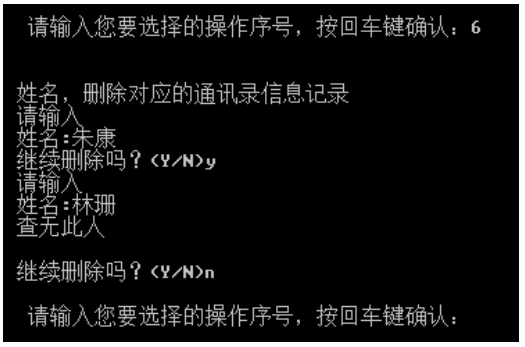


图 2-47　删除信息界面

12）保存数据界面如图 2-48 所示。

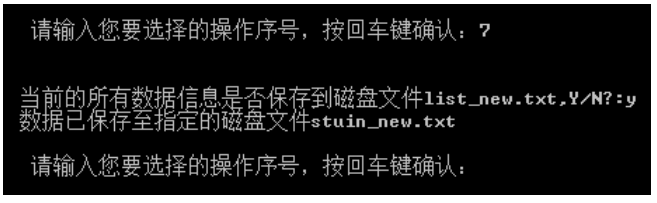


图 2-48　保存数据界面

3. 结果数据文件

通讯录信息管理系统程序运行结束后，会在磁盘上指定位置自动生成一个结果数据文件 addlistout.txt，将程序的运行结果写入该文件保存，如图 2-49 所示。

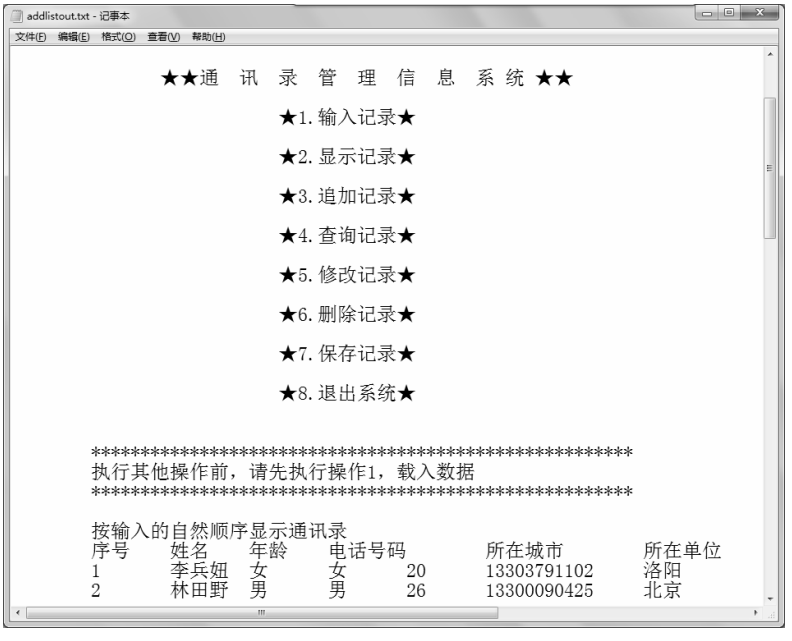


图 2-49　自动生成的数据

第 3 章　综合程序设计实训

本章结合前两章的内容，给出了一些典型的程序设计题目，通过对这些题目的分析设计，进一步强化读者对综合程序设计的基本思想和方法的掌握，训练灵活运用课程知识解决实际问题的能力。各实训题目针对系统功能描述给出了问题分析、总体设计、源代码参考框架等，测试结果均在 Visual C++ 6.0 环境下运行产生，读者可根据给出的设计要求和内容，完善详细设计、代码编写、代码调试等步骤，撰写程序设计报告。

3.1　字符串处理问题

字符串处理是程序设计课程中一类重要的问题，在实际中也有广泛应用，如信息查询、信息修改、字典管理等。本实验的内容涉及函数、数组、文件等方面的知识，旨在训练读者综合程序设计的能力，掌握利用数组或指针处理字符串的方法及文件操作，构建综合程序设计的思路及框架。

3.1.1　功能描述与要求

1. 功能描述

设计一个字符串处理系统，利用字符数组来处理字符串信息，要求实现如下系统功能：

（1）字符串输入

从键盘输入若干以字母和空格组成的字符串，存储在字符数组中；也可先建立一个数据文件，在文件中录入字符串，再通过格式化读写函数，将文件中的字符串读入字符数组。

（2）删除指定字符

删除每一个字符串中的指定字符，将删除指定字符后的字符串输出至屏幕上。

（3）字符串排序

将每一个字符串中的字符按 ASCII 码从小到大的顺序进行排列，例如，某一字符串为“kxDsAip”，排序后的字符串为“ADikpsx”，将排好序的字符串输出至屏幕上。

（4）字符替换

将字符串中的每个字母替换为字母表中其后的第 5 个字母，如遇大写字母，则先将其替换为小写，再替换为字母表中其后的第 5 个字母，空格不替换，将经过字符替换后的字符串输出至屏幕上。

（5）字符统计

统计指定字符在每个字符串中出现的次数。从键盘输入指定字符，在每个字符串中查找，统计其出现的次数，将次数输出至屏幕上。

（6）子字符串统计

统计某一指定子字符串在每个字符串中出现的次数。从键盘输入指定子字符串，在每个字符串中查找，统计其出现的次数，将次数输出至屏幕上。

（7）字符串反转

将字符数组中的每个字符串进行反转（逆序存放），例如，某一字符串为“abcde”，反转后的字符串为“edcba”，将反转后的字符串输出至屏幕上。

（8）保存信息

建立一个数据文件，如 file.txt。实现上述各功能时，分别将结果写入数据文件 file.txt 进行保存。

（9）系统主界面

进入字符串处理系统时，输出系统主界面，在主界面中显示系统各功能的名称及编号，用户根据需要选择执行相应的功能模块。

结合以上内容编写源程序，并写出本实验的综合程序设计报告。

2. 要求

（1）源程序编写要求

根据系统功能描述，采用模块化程序设计方法进行程序设计，要求程序结构清晰。字符串的输入、删除、排序、替换、查找和反转功能，分别用函数实现，在主函数中通过调用这些函数，完成系统功能的要求。代码书写要规范，有简要的注释，给出数据和函数说明。

（2）设计报告撰写要求

设计报告内容包括题目内容和要求、总体设计、详细设计、源代码、调试过程中的问题、总结等。

总体设计：对程序的整体设计思路进行描述，画出字符串处理系统的总体功能模块图，说明系统使用的主要数据结构，给出实现字符串输入、删除字符、字符串排序、字符替换、子字符串查找、字符串反转等模块的函数设计。

详细设计：分析实现各函数功能的算法，画出函数的算法流程图。

调试过程中的问题：记录程序编写和调试过程中遇到的各种问题，以及解决这些问题的途径和方法。

总结：回顾整个综合程序设计的过程，对学习到的设计方法和思路进行总结，写出个人体会。

设计报告的撰写参见第 4 章。

3.1.2 问题分析

字符串的处理可使用字符数组，本实验内容涉及多个字符串，可使用二维字符数组来处理，具体操作过程也可配合字符指针完成。

字符串的排序主要根据字符的 ASCII 码进行，可使用选择排序、冒泡排序等多种排序方法。字符的替换，通过改变字符的 ASCII 码值，实现对字符串的简单加密。统计子字符串出现的次数相对于统计单个字符要复杂一些，需要将存放子字符串的字符数组元素与存

放原字符串的字符数组元素进行比较，使用双重循环实现。字符串的反转可以在原数组中进行，反转过程中注意需要交换的两个字符的下标，以及循环次数的控制。最终，所建立的数据文件中包含初始的字符串数据，以及所有字符串处理模块的结果数据。

3.1.3 总体设计

1. 功能模块设计

根据系统功能描述和问题分析，可将系统功能划分为若干模块，如图 3-1 所示。

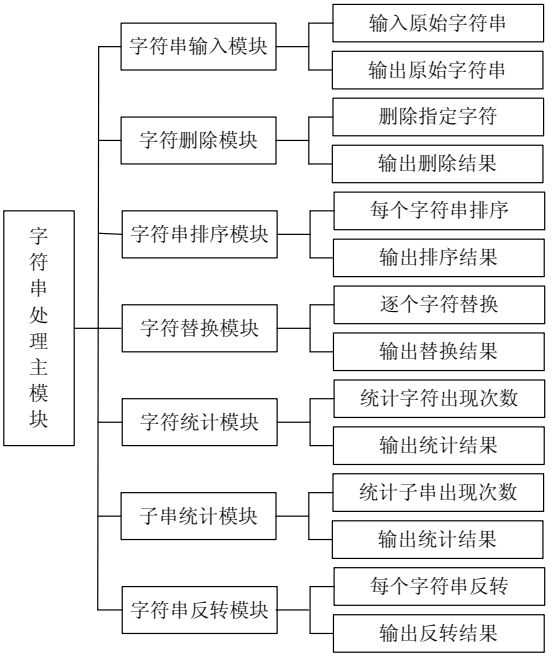


图 3-1 字符串处理系统功能模块图

进入字符串处理系统，首先显示系统主界面，列出八个菜单选项：字符串输入、字符删除、字符串排序、字符替换、字符统计、子字符串统计、字符串反转和退出程序，各选项编号为 1～7，0 为退出程序选项编号。用户根据主菜单中显示的功能模块名称，输入编号，选择执行相应的功能。用户必须先选择字符串输入模块，才可以执行其他模块。所选模块对应的操作执行完后，返回系统主界面，用户可以继续选择其他操作，直至选择编号 0（退出系统）。

2. 数据结构设计

系统涉及多个字符串的处理，通常定义二维字符数组来存放多个字符串。根据功能描述，需要输入若干字符串，可将二维数组定义如下。

```
#define M 5
#define N 30
char str[M][N];
```

若使用指针处理字符串，则可在主函数中定义指针数组：

```
    char *str[M];
```

将指针数组作为函数参数传递给各子模块，在子模块中对数组元素（即各字符串）进行处理。

3. 函数设计

（1）str_in()

字符串输入函数。函数参考原型为

void str_in(char str[M][N])。

str_in()函数用于输入待处理的原始字符串，将字符串存储在二维字符数组中。注意，使用 gets()函数输入的字符串中可以包含空格，而使用 scanf()函数则不能接收字符串中的空格。调用 str_out()函数将原始字符串输出至屏幕上，并写入数据文件。

（2）char_del()

字符删除函数。函数参考原型为

void char_del(char str[M][N])。

char_del()函数用于实现将每个字符串中的指定字符删除功能。从键盘输入要删除的字符，对每个字符串中的字符逐个进行判断，删除指定的字符。调用 str_out()函数将删除指定字符后的所有字符串输出至屏幕上，并追加写入数据文件。

（3）str_sort()

字符串排序函数。函数参考原型为

void str_sort(char str[M][N])。

str_sort()函数用于对二维字符数组的每一行数据，即每个字符串，分别进行排序，按照字符的 ASCII 码从小到大排列。可采用选择排序、冒泡排序等方法完成排序，调用 str_out()函数将排序后的各字符串输出至屏幕上，并追加写入数据文件。

（4）char_replace()

字符替换函数。函数参考原型为

void char_replace(char str[M][N])。

char_replace()函数用于实现对每个字符串的简单加密，即字符替换功能。将每个字符串中的字母进行替换，若遇到大写字母，则将其转换为相应的小写字母后再替换。注意，判断字符是否为空格，若为空格，则不进行替换。调用 str_out()函数将经过字母替换后的各字符串输出至屏幕上，并追加写入数据文件。

（5）char_stati()

字符统计函数。函数参考原型为

void char_stati(char str[M][N])。

char_stati()函数用于统计各字符串中指定字符的数量。从键盘输入要统计的字符，为每个字符串设定一个计数器，用于存放各字符串中指定字符出现的次数。遍历数组 str 的每个元素与指定字符进行比较，若找到一个指定字符，则相应的计数器加 1。调用 stati_out()函数将统计得到的各字符串中指定字符出现的次数输出至屏幕上，并追加写入数据文件。

（6）substr_stati()

子字符串统计函数。函数参考原型为

void substr_stati(char str[M][N])。

substr_stati()函数用于统计各字符串中指定子字符串的数量。从键盘输入要统计的子字

符串，将其存放在字符数组中。同样为每个字符串设定一个计数器，用于存放各字符串中指定子字符串出现的次数。将指定子字符串与原字符串进行比较，若找到一个指定子字符串，相应的计数器加 1。调用 stati_out()函数将统计得到的每个字符串中指定子字符串出现的次数输出至屏幕上，并追加写入数据文件。

（7）str_reverse()

字符串反转函数。函数参考原型为

void str_reverse(char str[M][N])。

str_reverse()函数实现字符串的逆序存放，即字符串反转功能。逆序存放的过程可在原数组中进行，将字符串的首字符和尾字符交换，再将字符串的第二个字符与倒数第二个字符交换，以此类推，经过 n/2 次（n 为原字符串的长度）交换，即可实现字符串的反转。调用 str_out()函数将反转后的字符串输出至屏幕上，并追加写入数据文件。

（8）str_out()

字符串输出函数。函数参考原型为

void str_out(char str[M][N], FILE *fp)。

str_out()函数用于实现将字符数组中的字符串输出至屏幕，并写入 fp 指向的数据文件功能。

（9）stati_out()

统计结果输出函数。函数参考原型为

void stati_out(int sum[M], FILE *fp)。

stati_out()函数用于将各字符串的统计结果（如每个字符串中所含指定字符的个数）输出至屏幕上，并写入 fp 指向的数据文件。

（10）menu()

显示系统主界面函数。函数参考原型为

void menu()。

menu()函数用于显示系统主菜单。用户进入字符串处理系统后，屏幕显示系统主菜单，在菜单中显示系统名称、各功能模块名称及编号。用户根据功能模块名称，输入某一编号，选择执行相应的功能模块。

（11）main()

字符串处理系统主函数。主函数实现数据定义、模块调用等功能，完成整个系统的功能控制。调用 menu()函数显示系统主菜单，提示用户输入选择的功能模块编号，根据编号执行用户选择的功能模块，所选模块功能完成后返回主菜单，由用户继续选择功能模块。若用户输入的编号超出范围，则输出提示信息。

3.1.4 源代码参考框架

1. 预处理

```
#include "stdio.h"
#include "string.h"
#define M 5
#define N 30
…
```

2. 子模块定义

```
void menu()
{
     /* 输出字符串处理系统名称;
        输出功能模块名称; */
}
void str_out(char str[M][N], FILE *fp)
{
     /* 变量定义;
        将数组str中的字符串输出至屏幕上;
        将数组str中的字符串输出至fp指向的数据文件中; */
}
void str_in(char str[M][N])
{
     /* 变量定义;
        输入M个字符串;
        打开数据文件，若文件不存在，则创建文件;
        调用str_out()函数，输出字符串;
        关闭数据文件; */
}
void stati_out(int sum[M], FILE *fp)
{
     /* 变量定义;
        输出每个字符串的统计结果至屏幕上 ;
        输出每个字符串的统计结果至fp指向的数据文件中; */
}
void char_del(char str[M][N])
{
     /* 变量定义;
        以追加方式打开数据文件;
        提示用户输入待删除的字符;
        删除每个字符串中的指定字符;
        调用str_out()函数，输出删除指定字符后的字符串;
        关闭数据文件; */
}
void str_sort(char str[M][N])
{
     /* 变量定义;
        以追加方式打开数据文件;
        对数组str中的每个字符串分别进行排序;
        调用str_out()函数，输出排序后的字符串;
        关闭数据文件; */
}
void char_replace(char str[M][N])
{
     /* 变量定义;
```

```
            以追加方式打开数据文件;
            每个字符串中的大写字母转换为对应的小写字母;
            每个字符串中的字母替换为字母表中其后的第5个字母;
            调用str_out()函数，输出字母替换后的字符串;
            关闭数据文件;  */
    }
    void char_stati(char str[M][N])
    {
        /* 变量定义;
            以追加方式打开数据文件;
            提示用户输入需要统计的字符;
            统计每个字符串中指定字符的个数;
            调用stati_out()函数，输出统计结果;
            关闭数据文件;  */
    }
    void substr_stati(char str[M][N])
    {
        /* 变量定义;
            以追加方式打开数据文件;
            提示用户输入需要统计的子字符串;
            统计每个字符串中指定子字符串的个数;
            调用stati_out()函数，输出统计结果;
            关闭数据文件;  */
    }
    void str_reverse(char str[M][N])
    {
        /* 变量定义;
            以追加方式打开数据文件;
            将每个字符串逆序存放;
            调用str_out()函数，输出反转后的字符串;
            关闭数据文件;  */
    }
```

3. 主函数

```
    int main()
    {
        /* 变量定义;
            调用menu()函数，显示系统主菜单;
            提示用户输入编号;
            根据编号调用函数，执行相应的功能模块;  */
    }
```

3.1.5 功能测试

为方便测试输入，以 5 个含有字母和空格的字符串为例，将其作为初始信息输入，进行功能测试。

1）输出系统主菜单，如图 3-2 所示。

```
                    the string processing system

*******************************  MENU  *******************************

     1.  input string                 2.  delete character
     3.  string sort                  4.  character replace
     5.  character statistic          6.  substring statistic
     7.  string inversion             0.  quit program

*************************************************************************

Please enter the number(0-7):
```

图 3-2　系统主菜单

2）用户选择编号“1”，从键盘输入原始字符串，如图 3-3 所示。

```
Please enter the number(0-7): 1

Please input 5 strings:
HHH aaa pppppp YYY
NEW new NEW
yyeeaaRR
totoTOtoto tttooo
You youyou
```

图 3-3　输入原始字符串

原始字符串被输出，返回系统主菜单，继续选择，如图 3-4 所示。

```
The original strings:
HHH aaa pppppp YYY
NEW new NEW
yyeeaaRR
totoTOtoto tttooo
You youyou



                    the string processing system

*******************************  MENU  *******************************

     1.  input string                 2.  delete character
     3.  string sort                  4.  character replace
     5.  character statistic          6.  substring statistic
     7.  string inversion             0.  quit program

*************************************************************************

Please enter the number(0-7):
```

图 3-4　返回系统主菜单

3）用户选择编号“2”，输入待删除的字符“e”，输出删除指定字符后的字符串，如图 3-5 所示。

```
Please enter the number(0-7): 2

Please input the character to be deleted: e
e is to be deleted!
After deleting the character, strings are:
HHH aaa pppppp YYY
NEW nw NEW
yyaaRR
totoTOtoto tttooo
You youyou
```

图 3-5　删除指定字符

返回系统主菜单，继续选择。

4）用户选择编号“3”，排序后的字符串如图 3-6 所示。

```
Please enter the number(0-7): 3

The sorted strings:
   HHHYYYaaappppppp
  EENNWWenw
RRaaeeyy
 OTooooooottttttt
 Yooouuuyy
```

图 3-6　字符串排序

返回系统主菜单，继续选择。

5）用户选择编号“4”，完成替换后的字符串如图 3-7 所示。

```
Please enter the number(0-7): 4

The replaced strings:
mmm fff uuuuuu ddd
sjb sjb sjb
ddjjffww
ytytytytyt yyyttt
dtz dtzdtz
```

图 3-7　替换字符串

返回系统主菜单，继续选择。

6）用户选择编号“5”，输入待查字符为“a”，统计结果如图 3-8 所示。

```
Please enter the number(0-7): 5

Please input the character to be counted: a
The number of a in the 5 strings are:
string1: 3
string2: 0
string3: 2
string4: 0
string5: 0
```

图 3-8　统计字符出现次数

返回系统主菜单，继续选择。

7）用户选择编号“6”，输入需要统计的子字符串为“to”，统计结果如图 3-9 所示。

```
Please enter the number(0-7): 6

Please input the substring: to
The number of to in the 5 strings are:
string1: 0
string2: 0
string3: 0
string4: 5
string5: 0
```

图 3-9　统计子字符串出现次数

返回系统主菜单，继续选择。

8）用户选择编号“7”，完成反转后的字符串如图 3-10 所示。

```
Please enter the number(0-7): 7

The reversed strings:
YYY pppppp aaa HHH
WEN wen WEN
RRaaeeyy
ooottt ototOTotot
uoyuoy uoY
```

图 3-10　反转字符串

返回系统主菜单，继续选择。

9）在没有输入原始字符串的情况下，选择“0”和“1”以外的编号，会给出提示信息，提示用户需进行输入字符串的操作，如图 3-11 所示。

```
                    the string processing system

*******************************  MENU  *******************************

     1.  input string                  2.  delete character
     3.  string sort                   4.  character replace
     5.  character statistic           6.  substring statistic
     7.  string inversion              0.  quit program

*****************************************************************************

Please enter the number(0-7): 3

Please input strings first!
```

图 3-11　提示输入原始信息

返回系统主菜单，继续选择。

10）若用户选择编号“0”，则退出系统。若用户输入编号“0”～“7”以外的整数，则给出输入错误的提示信息。

功能测试时，不必按菜单项编号顺序进行测试，可随机选择编号，执行程序各个功能模块，测试系统各功能模块执行结果是否正确。也可使用各种符号组成的字符串进行测试，不断修改程序，使其更加完善。

3.1.6　拓展思考

本实验内容仅涉及几种常用的字符串操作，在实际应用中，字符串操作还有多种方式。可以对系统功能进行补充，如扩展以下字符串处理功能。

1）在字符串中插入一个指定的字符串。

2）删除字符串中指定的子字符串。

3）在字符串中查找并替换指定的子字符串。

4）统计字符串中单词的数量。

5）统计字符串中各单词的长度。

6）多个字符串排序。

7）字符串加密。

对多个字符串的处理可以使用二维数组也可使用指针数组，本实验在处理字符串时以二维数组为例。对于多个字符串的处理，通常各字符串的长度是不同的，而二维数组每行的列数相同，至少要大于最长的字符串长度。当处理的字符串较多时，会造成较大的内存空间浪费。而使用指针数组来处理多个不同长度的字符串时，能避免空间浪费，且在对多个字符串进行排序时，移动指针比移动字符数组的代价要小。读者可使用指针数组完成实

验中字符串的处理。

3.2 选票问题

选票问题是实际应用中常见的一类问题，如各种选举、选拔、评比活动。此类问题需要对相互关联的一组不同类型的数据进行处理，结构体数组是批量组织复合型数据的一种有效方式。本实验的内容涉及函数、结构体数组、文件等方面的知识，旨在使读者掌握使用结构体类型组织数据的方式，掌握文件操作，构建综合程序设计的思路及框架，提高综合程序设计能力。

3.2.1 功能描述与要求

1. 功能描述

设计一个选票信息处理系统，从 10 名优秀运动员中评选出 3 名超级运动员。要求实现如下系统功能。

（1）输入运动员信息及选票信息

运动员信息包括运动员编号、姓名、运动员得分和运动员得票。运动员按 1、2、3…顺序编号。选票信息包括选票编号、所投运动员编号、选票有效标志。选票同样按 1、2、3…顺序编号。每张选票可投 3 个不同的运动员编号；对应位置的运动员编号可以有空缺，但必须用 0 表示；若编号超出规定的范围，或编号出现重复，则选票无效。从键盘或数据文件输入各运动员信息和选票信息。

（2）统计运动员得分

选票中所列运动员顺序不同，则得分不同，选票中第 1 位运动员至第 3 位运动员所得分数依次为 3 分、2 分、1 分。根据选票信息中的运动员编号及顺序，统计运动员得分，记入相应的得分数据域；统计运动员得票数，记入相应的得票域。

（3）输出运动员及选票信息

输出所有运动员信息及选票信息至屏幕上。建立一个数据文件，将结构体数组中的运动员信息和选票信息写入数据文件。

（4）查询运动员信息

按运动员编号或姓名查询运动员信息。从键盘输入待查询运动员的编号或姓名，若找到该运动员，则输出其编号、姓名、得分及得票信息；若未找到，则输出提示信息，将查询结果输出至屏幕上。

（5）查询选票信息

从键盘输入待查询选票的编号，若找到该选票，则输出选票信息；若未找到，则输出提示信息，将查询结果输出至屏幕上。

（6）评选超级运动员

根据各运动员信息中的得分域，评选出得分最高的三名运动员为超级运动员。若运动员得分相同，则得票多者在前，如果得分与票数都相同，则编号小的在前。输出超级运动

员排名，格式如下。

Rank　　Number　　Name　　Score　　Votes

将超级运动员信息输出至屏幕上，并追加写入数据文件。

（7）系统主界面。

进入选票信息处理系统时，输出系统主界面。在主界面中显示系统各功能的名称及编号，用户根据需要选择执行相应的功能模块。

结合以上内容编写源程序，并写出本实验的综合程序设计报告。

2. 要求

（1）源程序编写要求

根据系统功能描述，采用模块化程序设计方法进行程序设计，要求程序结构清晰。运动员及选票信息的输入、输出、查询、统计和评选等功能，分别用函数实现，在主函数中通过调用这些函数，完成系统功能的要求。代码书写要规范，有简要的注释，给出数据和函数说明。

（2）设计报告撰写要求

设计报告内容包括题目内容和要求、总体设计、详细设计、源代码、调试过程中的问题、总结等。

总体设计：对程序的整体设计思路进行描述，画出选票信息处理系统的总体功能模块图；说明系统使用的主要数据结构；给出实现运动员信息的输入输出、选票信息的输入输出、查询运动员信息、查询选票信息、统计运动员得分和评选超级运动员等模块的函数设计。

详细设计：分析实现各函数功能的算法，画出函数的算法流程图。

调试过程中的问题：记录程序编写和调试过程中遇到的各种问题，以及解决这些问题的途径和方法。

总结：回顾整个综合程序设计的过程，对学习到的设计方法和思路进行总结，写出个人体会。

设计报告的撰写参见第 4 章。

3.2.2　问题分析

运动员和选票信息由多个数据项组成，存储在结构体变量中，实验内容涉及多个运动员及选票记录，使用结构体数组来处理，数组下标应与运动员编号和选票编号对应，便于后续的选票统计处理。

初始的运动员信息中得分和得票数应输入 0，待选票统计结束后，将统计结果赋值给相应的数据域。统计选票的过程包括计算运动员得分、统计运动员得票数及标记选票是否有效。在选票统计完成后，输出完整的运动员信息和选票信息。

超级运动员的评选，若根据运动员得分对结构体数组进行排序，则需要调整各运动员记录的顺序，记录较多时，占用空间较大。为避免多次进行整条记录信息的交换，可考虑采用记录下标的方法找出三位超级运动员，并将其信息输出至屏幕和数据文件中。最终，数据文件中包括所有运动员和选票的完整信息，以及评选出的超级运动员信息。

3.2.3 总体设计

1. 功能模块设计

根据系统功能描述和问题分析，可将系统功能划分为若干模块，如图 3-12 所示。

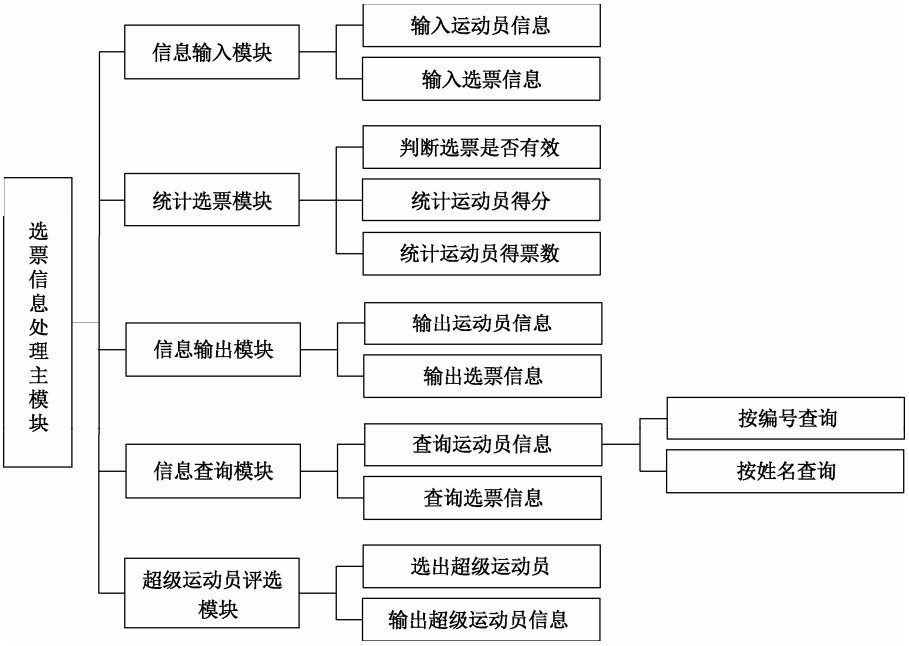


图 3-12　选票信息处理系统功能模块图

进入选票信息处理系统，首先显示系统主界面，菜单列出七个选项：输入运动员和选票信息、输出所有运动员信息、输出所有选票信息、运动员信息查询、选票信息查询、输出超级运动员、退出程序。各选项编号为 1～6，0 为退出程序选项编号。用户根据主菜单中显示的功能模块名称，输入编号，选择执行相应的功能。用户必须先选择输入运动员和选票信息模块，才可以执行其他模块。所选模块对应的操作执行完后，返回系统主界面，用户可以继续选择其他操作，直至选择编号 0（退出系统）。

2. 数据结构设计

实验内容涉及多条记录的处理，每条记录含有不同类型的数据项，定义结构体数组来存放这些记录。根据功能描述，可将存放运动员信息的结构体 player 和存放选票信息的结构体 vote 分别定义如下。

```
#define M 10
#define N 30
struct player
{
     int num;
     char name[20];
     int score;
     int votes;
```

```
};
struct vote
{
     int num;
     int top[3];
     int valid;
};
```

其中结构体 player 中各数据域的含义如下。

num：运动员编号。

name[20]：运动员姓名。

score：运动员得分。

votes：运动员得票数。

结构体 vote 中各数据域的含义如下。

num：选票编号。

top[3]：选票所选的三名运动员的编号。

valid：选票有效标志（valid 为 1，则选票有效；valid 为 0，则选票无效）。

结构体数组既可以与结构体类型一起在程序开始处定义，作为全局变量使用；还可以在主函数中定义，调用各函数时，作为函数参数传递给各子模块。

```
struct player ply[M];
struct vote vot[N];
```

3. 函数设计

（1）player_in()

运动员信息输入函数。函数参考原型为

void player_in(struct player ply[M])。

player_in()函数用于从键盘或数据文件输入运动员信息，将其存储在结构体数组中。若从键盘输入数据，则可给出提示信息，提示用户输入各运动员的编号、姓名、得分、得票数等信息。

（2）vote_in()

选票信息输入函数。函数参考原型为

vote_in(struct vote vot[N])。

vote_in()函数实现从键盘或数据文件输入选票信息，将其存储在结构体数组中。若从键盘输入数据，则可给出提示信息，提示用户输入各选票的编号、选票所投三位运动员的编号等信息。

（3）in_valid()

选票有效性判断函数。函数参考原型为

int in_valid(int top[3])。

in_valid()函数用于判断选票是否有效。根据评选规则判断，若选票有效，则函数返回值为 1；若选票无效，则函数返回值为 0。函数形参数组为 top，用于接收选票所投三位运动员的编号。

（4）scoring()

统计选票函数。函数参考原型为

void scoring(struct player ply[M], struct vote vot[N])。

scoring()函数用于实现运动员计分功能。首先调用 in_valid()函数，判断选票是否有效，若选票无效，则处理下一张选票；若选票有效，则根据选票中所投三位运动员的顺序，为运动员计分，并统计运动员得票数。

（5）orig_to_file()

信息写入文件函数。函数参考原型为

void orig_to_file(struct player ply[M], struct vote vot[N])。

orig_to_file()函数用于将运动员信息和选票信息写入数据文件。此时写入的运动员信息已包括统计选票后计算出的得分和得票数，选票信息包括判断后的选票有效标志。建立数据文件，将上述信息逐条写入数据文件。

（6）load_mod()

信息输入模块函数。函数参考原型为

void load_mod(struct player ply[M], struct vote vot[N])。

load_mod()函数通过调用 player_in ()、vote_in()和 scoring()函数，将运动员信息输入、选票信息输入和统计选票等功能组合成模块，供主函数调用。

（7）output_player()

运动员信息输出函数。函数参考原型为

void output_player(struct player ply[M])。

output_player()函数用于将所有运动员的信息逐条输出至屏幕上，包括编号、姓名和计算后的得分及得票数。

（8）output_vote()

选票信息输出函数。函数参考原型为

void output_vote(struct vote vot[N])。

output_vote()函数用于将所有选票的信息逐条输出至屏幕上，包括编号、所选三位运动员及经过判断后的选票有效标志。

（9）search_num_player()。

按编号查询运动员函数。函数参考原型为

void search_num_player(struct player ply[M])。

search_num_player()函数用于实现按照编号查询某位运动员信息的功能。从键盘输入所查询运动员的编号，进行逐条记录查找，若找到该记录，则输出运动员信息至屏幕上；若未找到，则输出提示信息。

（10）search_name_player()

按姓名查询运动员函数。函数参考原型为

void search_name_player(struct player ply[M])。

search_name_player()函数用于实现按照姓名查询某位运动员信息的功能。从键盘输入所查询运动员的姓名，进行逐条记录查找，若找到该记录，则输出运动员信息至屏幕上；若未找到，则输出提示信息。

（11）search_player()

查询运动员信息函数。函数参考原型为

void search_player(struct player ply[M])。

search_player()函数用于实现运动员信息的查询。可分为按编号查询和按姓名查询两种方式，若用户选择按编号查询，则调用 search_num_player()函数完成查询；若用户选择按姓名查询，则调用 search_name_player()函数完成查询。

（12）search_num_vote()

按编号查询选票函数。函数参考原型为

void search_num_vote(struct vote vot[N])。

search_num_vote()函数用于实现按照编号查询某张选票信息的功能。从键盘输入所查询选票的编号，进行逐条记录查找，若找到该记录，则输出选票信息至屏幕上；若未找到，则输出提示信息。

（13）super_player ()

评选超级运动员函数。函数参考原型为

void super_player(struct player ply[M])。

super_player ()函数用于实现三名超级运动员的评选及输出。函数定义三个变量，分别存放三名超级运动员编号的下标，采用比较运动员得分域、替换下标的方法找出三位超级运动员，并将其信息输出。

（14）menu()

显示系统主界面函数。函数参考原型为

void menu()。

menu()函数用于显示系统主菜单。用户进入选票信息处理系统后，屏幕显示系统主菜单，在菜单中显示系统名称，各功能模块名称及编号。用户根据功能模块名称，输入某一编号，选择执行相应的功能模块。

（15）main()

选票信息处理系统主函数。主函数实现数据定义、模块调用等功能，完成整个系统的功能控制。调用 menu()函数显示系统主菜单，提示用户输入选择的功能模块编号，根据编号执行用户选择的功能模块，所选模块功能完成后返回主菜单，由用户继续选择功能模块。若用户输入的编号超出范围，则输出提示信息。

3.2.4 源代码参考框架

1. 预处理

```
#define M 10
#define N 30
#include "stdio.h"
/*结构体定义*/
 …
```

2. 子模块定义

```
void menu()
{
       /* 输出选票系统名称;
          输出功能模块名称;  */
}
void player_in(struct player ply[M])
```

```
{
        /* 变量定义;
           提示用户输入M位运动员的信息;
           逐条逐项输入每位运动员的信息; */
}
void vote_in(struct vote vot[N])
{
        /*变量定义;
           提示用户输入N张选票的信息;
           逐条逐项输入每张选票的信息; */

}
int in_valid(int top[3])
{
        /*变量定义;
           初始化标志变量为1;
           若所选三位运动员编号均为0，则标志变量置为0;
           若所选编号重复，则标志变量置为0;
           若所选编号超出规定范围，则标志变量置为0;
           返回标志变量; */
}
void scoring(struct player ply[M],struct vote vot[N])
{
        /*变量定义;
           判断每张选票是否有效;
           若有效，则统计运动员得分及得票数;
           选票所选第一位运动员编号若不是0，则该运动员得分加3，得票数加1;
           选票所选第二位运动员编号若不是0，则该运动员得分加2，得票数加1;
           选票所选第三位运动员编号若不是0，则该运动员得分加1，得票数加1; */
}
void orig_to_file(struct player ply[M],struct vote vot[N])
{
        /*变量定义;
           建立数据文件;
           逐条写入所有运动员信息;
           逐条写入所有选票信息;
           关闭数据文件; */
}
void load_mod(struct player ply[M],struct vote vot[N])
{
        /*调用player_in()函数完成运动员信息输入;
           调用scoring()函数完成选票统计;
           调用orig_to_file()函数将信息写入数据文件; */
}
void output_player(struct player ply[M])
{
        /*变量定义;
           逐条输出运动员信息; */
}
void output_vote(struct vote vot[N])
{
```

```
        /*变量定义;
          逐条输出选票信息;  */
}
void search_num_player(struct player ply[M])
{
        /*变量定义;
          提示用户输入运动员编号;
          若找到，则输出该运动员信息;
          若未找到，则输出提示信息;  */
}
void search_name_player(struct player ply[M])
{
        /*变量定义;
          提示用户输入运动员姓名;
          若找到，则输出该运动员信息;
          若未找到，则输出提示信息;  */
}
void search_player(struct player ply[M])
{
        /*变量定义;
          提示用户选择按编号查询还是按姓名查询;
          若选择编号，则调用search_num_player()函数进行查询;
          若选择姓名，则调用search_name_player()函数进行查询;  */
}
void search_num_vote(struct vote vot[N])
{
        /*变量定义;
          提示用户输入选票编号;
          若找到，则输出该选票信息;
          若未找到，则输出提示信息;  */
}
void super_player(struct player ply[M])
{
        /*变量定义;
          为三位超级运动员所在下标变量a，b，c赋初值;
          逐个运动员的得分与a、b、c对应运动员的得分比较;
          若下标为i的运动员得分大于下标为a的运动员得分，
          则将b赋给c，a赋给b，i赋给a;
          若下标为i的运动员得分等于下标为a的运动员得分，
          则比较二者的得票数，继续判断;
          若下标为i的运动员得分小于下标为a的运动员得分，
          则比较i与b的情况，以此类推;
          输出超级运动员的信息至屏幕和数据文件中;  */
}
```

3. 主函数

```
int main()
{
        /*变量定义;
          调用menu()函数，显示系统主菜单;
          提示用户输入编号;
```

```
                根据编号调用函数，执行相应的功能模块；  */
        }
```

3.2.5 功能测试

为方便测试输入，以 5 名运动员和 10 张选票信息为例，如表 3-1 和表 3-2 所示，将其作为初始信息输入，进行功能测试。

表 3-1 运动员信息测试数据

Number	Name	Score	Votes
1	001	0	0
2	002	0	0
3	003	0	0
4	004	0	0
5	005	0	0

表 3-2 选票信息测试数据

Number	Player1	Player2	Player3
1	2	3	1
2	4	1	5
3	2	5	8
4	1	1	3
5	5	3	2
6	2	4	3
7	2	1	5
8	3	1	6
9	2	3	4
10	1	2	3

1）显示系统主菜单，如图 3-13 所示。

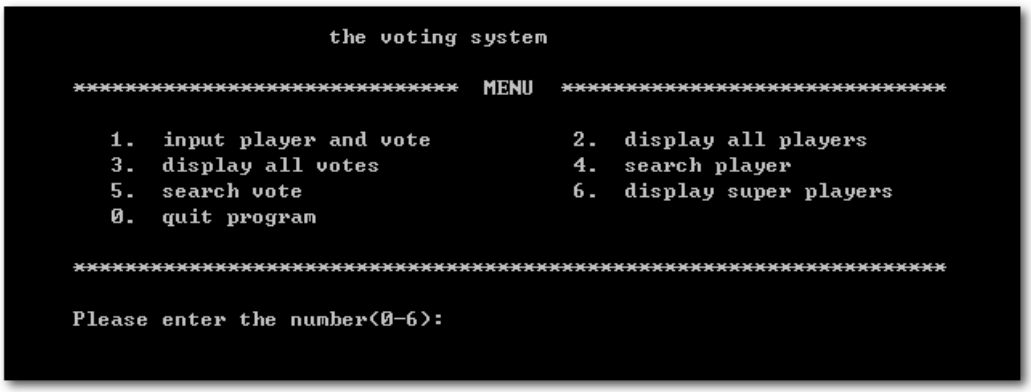


图 3-13 系统主菜单

2）用户选择编号“1”，根据提示从键盘逐项输入运动员信息，如图 3-14 所示。

```
Please enter the number(0-6): 1

Please input the information of 5 players:
player1's number: 1
player1's name: 001
player1's score: 0
player1's votes: 0
player2's number:
```

图 3-14　输入运动员信息

根据提示逐条输入选票信息，如图 3-15 所示。

```
Please input the information of 5 votes:
vote1's num: 1
first player number: 2
second player number: 3
third player number: 1
vote2's num:
```

图 3-15　输入选票信息

输入完成后，返回系统主菜单，继续选择。

3）用户选择编号“2”，输出所有运动员信息，如图 3-16 所示。

```
Please enter the number(0-6): 2

All players:
Number  Name    score   votes
1       001     8       4
2       002     15      6
3       003     8       5
4       004     6       3
5       005     5       3
```

图 3-16　输出所有运动员信息

返回系统主菜单，继续选择。

4）用户选择编号“3”，输出所有选票信息，如图 3-17 所示。

返回系统主菜单，继续选择。

5）用户选择编号“4”，按编号查询运动员信息，如图 3-18 所示。

```
Please enter the number(0-6): 3

All votes:
Number  player1 player2 player3 valid
1       2       3       1       1
2       4       1       5       1
3       2       5       8       0
4       1       1       3       0
5       5       3       2       1
6       2       4       3       1
7       2       1       5       1
8       3       1       6       0
9       2       3       4       1
10      1       2       3       1
```

图 3-17　输出所有选票信息

```
Please enter the number(0-6): 4

Search for num(1) or name(2)? enter the number: 1

Please input the num: 2
Search result:
Number  Name    score   votes
2       002     15      6
```

图 3-18　按编号查询运动员信息

用户选择编号“4”，按姓名查询运动员信息，如图 3-19 所示。

```
Please enter the number(0-6): 4

Search for num(1) or name(2)? enter the number: 2

Please input the name: 002
Search result:
Number  Name    score   votes
2       002     15      6
```

图 3-19　按姓名查询运动员信息

返回系统主菜单，继续选择。

6）用户选择编号“5”，按编号查询选票信息，如图 3-20 所示。

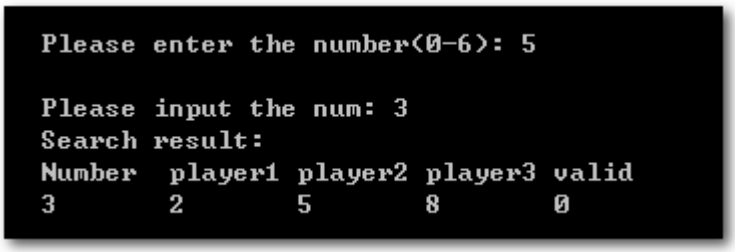


图 3-20　按编号查询选票信息

返回系统主菜单，继续选择。

7）用户选择编号“6”，输出超级运动员，如图 3-21 所示。

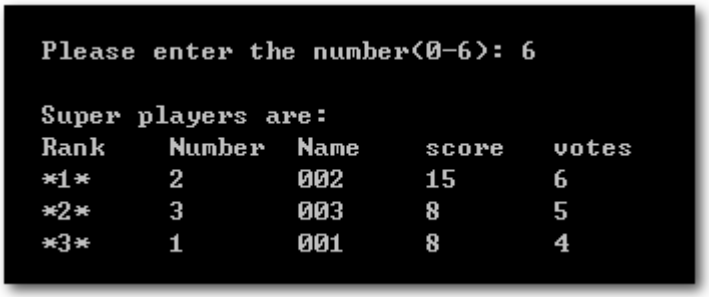


图 3-21　输出超级运动员

返回系统主菜单，继续选择。

8）在没有输入原始运动员和选票信息的情况下，选择“0”和“1”以外的编号，则给出提示信息，提示用户先进行输入原始信息的操作，如图 3-22 所示。

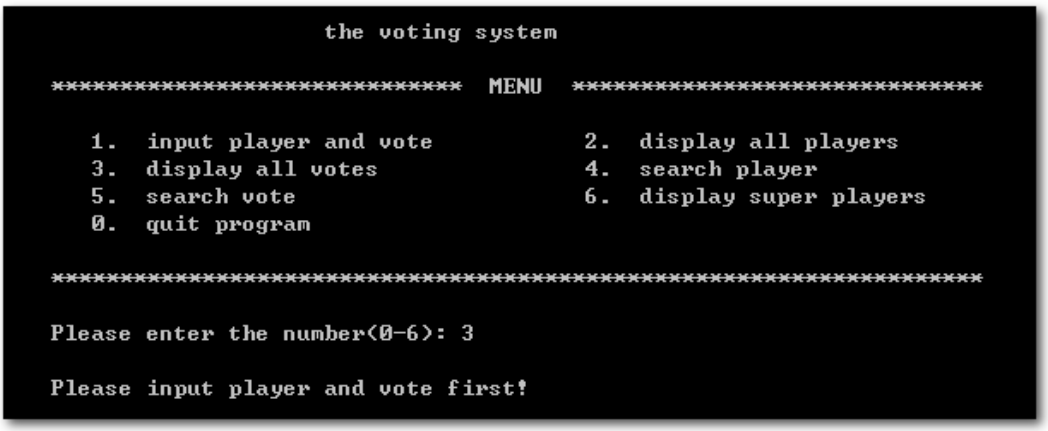


图 3-22　提示先输入原始信息

返回系统主菜单，继续选择。

9）若用户选择编号“0”，则退出系统。若用户输入编号“0”～“7”以外的整数，则

给出输入错误的提示。

功能测试时，不必按菜单项编号顺序进行测试，可随机选择编号，执行程序各个功能模块，测试系统各功能模块执行结果是否正确。使用多组不同的运动员和选票信息进行测试，不断修改程序，使其更加完善。

3.2.6 拓展思考

本实验是对运动员和选票信息的处理，在已有功能的基础上，可以对系统功能进行补充，如扩展以下功能。

1）评选人气最高的运动员，不计得分，只统计得票数，得票最多者获得人气最高运动员称号。

2）统计无效选票的数量，当无效选票超出一定比例时，判定评选无效，评选活动需重新进行，系统返回初始界面，用户重新输入选票。

3）出现两运动员得分及得票都相同的情况，为保证公平，允许系统对得分相同的运动员重新投票。

4）增加投票人获奖环节。记录投票人信息（编号、电话等），将最终超级运动员编号与选票中所选编号比对，统计选票的命中率，选出最佳参评人。

实际应用中，关于选票处理类的评选活动还有很多，如各大网站竞相推出的歌曲排行榜，根据歌曲的关注度（点击量、下载量和搜索量等）发布 top 10 歌曲排行榜等。

3.3 产品销售记录处理系统

产品销售记录处理是企业销售管理的重要环节，产品的销量统计、价格调整等都需要对产品销售记录进行相应处理。产品销售记录处理的规范化、系统化是企业提高管理效率的必要因素。本实验使用结构体数组来处理数据，涉及函数、结构体数组、文件等方面的知识，旨在使读者熟练掌握结构体数组的使用和文件操作，提高基本编程能力，构建综合程序设计的思路及框架。

3.3.1 功能描述与要求

1. 功能描述

设计一个产品销售记录处理系统，利用结构体数组来处理产品信息、销售员信息、销售记录信息，要求实现如下系统功能。

（1）输入销售信息

从键盘或数据文件输入产品信息、销售员信息和销售记录信息，将其存入结构体数组。产品的信息包括产品代码、产品名称、单价、销量和销售额。产品代码按 1、2、3…顺序编号。销售员信息包括销售员工号、姓名、销售额。销售员工号同样按 1、2、3…顺序编号。销售记录信息包括：销售记录编号、所售产品代码、销售员工号、销售量。销售记录

也按 1、2、3…顺序编号。

（2）统计销售记录

根据销售记录统计产品的总销量及销售额，统计每位销售员的销售额。

（3）输出销售信息

输出产品信息、销售员信息或销售记录信息至屏幕上。可选择输出其中某类信息，也可输出全部信息。

（4）查询销售信息

查询产品信息、销售员信息或销售记录信息。若选择查询产品信息，则可按代码或名称进行查询，从键盘输入待查询产品的代码或名称，若找到该产品，则输出其产品信息；若未找到，则输出提示信息。对销售员的查询，分为按工号或姓名两种方式。对销售记录的查询按照记录编号进行。将查询结果输出至屏幕上。

（5）销售信息排序

对产品信息和销售员信息进行排序。若对产品信息排序，则可选择按单价或按销售额进行排序。若对销售员信息排序，则按照销售额进行排序。将排序结果输出至屏幕上。

（6）保存信息

建立一个数据文件，将产品信息、销售员信息和销售记录信息，从结构体数组中写入数据文件进行保存。

（7）系统主界面

进入产品销售记录处理系统时，输出系统主界面。在主界面中显示系统各功能的名称及编号，用户根据需要选择执行相应的功能模块。

结合以上内容编写源程序，并写出本实验的综合程序设计报告。

2. 要求

（1）源程序编写要求

根据系统功能描述，采用模块化程序设计方法进行程序设计，要求程序结构清晰。产品、销售员和销售信息的输入、输出、查询、排序等功能，分别用函数实现，在主函数中通过调用这些函数，完成系统功能的要求。代码书写要规范，有简要的注释，给出数据和函数说明。

（2）设计报告撰写要求

设计报告内容包括题目内容和要求、总体设计、详细设计、源代码、调试过程中的问题、总结等。

总体设计：对程序的整体设计思路进行描述，画出产品销售记录处理系统的总体功能模块图；说明系统使用的主要数据结构；给出实现产品信息的输入/输出、销售员信息的输入/输出、销售记录信息的输入/输出、查询产品信息、查询销售员信息、查询销售记录信息、产品信息排序和销售员排序等模块的函数设计。

详细设计：分析实现各函数功能的算法，画出函数的算法流程图。

调试过程中的问题：记录程序编写和调试过程中遇到的各种问题，以及解决这些问题的途径和方法。

总结：回顾整个综合程序设计的过程，对学习到的设计方法和思路进行总结，写出个人体会。

设计报告的撰写参见第 4 章。

3.3.2 问题分析

产品、销售员和销售记录信息由多个数据项组成，存储在各自对应的结构体数组中。这三类信息之间相互存在联系，各自的数组下标应分别与产品代码、销售员工号和销售记录编号对应，便于统计销售记录时，对产品和销售员信息的关联处理。

初始的产品信息中，销量和销售额应输入 0；初始的销售员信息中，销售额也应输入 0。待销售记录统计结束后，将统计结果赋值给相应的数据域。统计销售记录的过程包括计算每种产品的总销量和销售额，每位销售员的销售额。产品信息、销售员信息和销售记录信息的输出应在统计销售记录后进行，此时输出的是完整的产品和销售员信息。对产品和销售员信息的排序可采用冒泡排序、选择排序等方法。保存信息后，完整的产品信息、销售员信息和销售记录信息被写入数据文件。

3.3.3 总体设计

1. 功能模块设计

根据系统功能描述和问题分析，可将系统功能划分为若干模块，如图 3-23 所示。

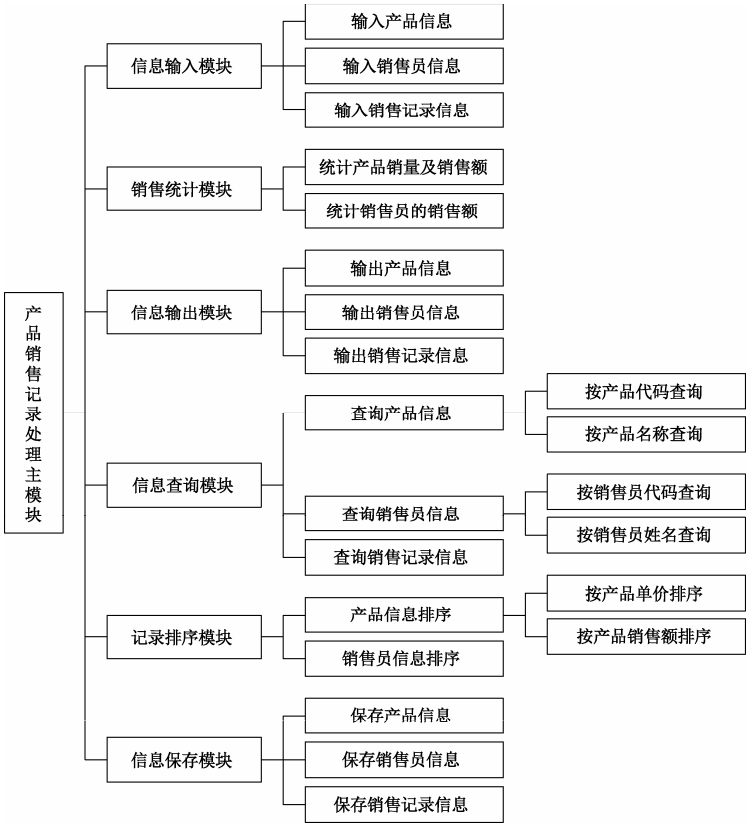


图 3-23　产品销售记录处理系统功能模块图

进入产品销售记录处理系统，首先显示系统主界面，菜单列出六个选项：输入原始销

售信息并统计销售记录、输出所有销售信息、销售信息查询、销售信息排序、保存销售信息、退出程序。各选项编号为 1～5，0 为退出程序选项编号。用户根据主菜单中显示的功能模块名称，输入编号，选择执行相应的功能。用户必须先选择输入原始销售信息并统计销售记录，才可以执行其他模块。所选模块对应的操作执行完后，返回系统主界面，用户可以继续选择其他操作，直至选择编号 0（退出系统）。

2. 数据结构设计

实验内容涉及多条记录的处理，每条记录含有不同类型的数据项，定义结构体数组来存放这些记录。根据功能描述，可将存放产品信息的结构体 product、存放销售员信息的结构体 s_person 和存放销售记录信息的结构体 sale 分别定义如下。

```
#define L 10
#define M 5
#define N 20
struct product
{
     int code;
     char name[20];
     int price;
     int count;
     int sum;
};
struct s_person
{
     int code;
     char name[20];
     int amount;
};
struct sale
{
     int num;
     int p_code;
     int s_code;
     int count;
};
```

其中结构体 product 中各数据域的含义如下。

code：产品代码。

name[20]：产品名称。

price：产品单价。

count：产品销量。

sum：产品销售额。

结构体 s_person 中各数据域的含义如下。

code：销售员工号。

name[20]：销售员姓名。

amount：销售员的销售额。

结构体 sale 中各数据域的含义如下。

num：销售记录编号。

p_code：所售产品代码。

s_code：本次销售的销售员工号。

count：本次销售产品的销量。

结构体数组可以与结构体类型一起在程序开始处定义，作为全局变量；也可以在主函数中定义，调用各函数时，作为函数参数传递给各子模块。

```
struct product prod[L];
struct s_person per[M];
struct sale sa[N];
```

3. 函数设计

（1）product_in()

产品信息输入函数。函数参考原型为

void product_in(struct product prod[L])

product_in ()函数用于从键盘或数据文件输入产品信息，将其存储在结构体数组中。若从键盘输入数据，则可以给出提示信息，提示用户输入产品的代码、名称、单价、销量和销售额等信息。

（2）person_in()

销售员信息输入函数。函数参考原型为

void person_in(struct s_person per[M])

person_in() 函数用于从键盘或数据文件输入销售员信息，将其存储在结构体数组中。若从键盘输入数据，则给出提示信息，提示用户输入销售员的工号、姓名和销售额等信息。

（3）sale_in()

销售记录输入函数。函数参考原型为

void sale_in(struct sale sa[N])

sale_in() 函数用于从键盘输入销售记录的信息，将其存储在结构体数组中。若从键盘输入数据，则给出提示信息，提示用户输入销售记录的编号、所售产品的代码、销售员工号和销量等信息。

（4）counting()

销售记录统计函数。函数参考原型为

void counting(struct product prod[L], struct s_person per[M], struct sale sa[N])

counting()函数实现对每条销售记录进行统计的功能。根据销售记录中的产品代码和销量，修改对应产品的销量，最终计算出产品的销售额；再根据销售记录中的销售员工号，修改该销售员的销售额。

（5）product_out()

产品信息输出函数。函数参考原型为

void product_out(struct product prod[L])

product_out()函数用于将所有产品信息逐条输出至屏幕上，包括代码、名称、单价、销量和销售额。

（6）person_out()

销售员信息输出函数。函数参考原型为

void person_out(struct s_person per[M])

person_out()函数用于将所有销售员的信息逐条输出至屏幕，包括工号、姓名和销售额。

（7）sale_out()

销售记录输出函数。函数参考原型为

void sale_out(struct sale sa[N])

sale_out()函数用于将所有销售记录的信息逐条输出至屏幕上，包括销售记录编号、所售产品代码、销售员工号和所售产品的销量。

（8）info_out()

信息输出函数。函数参考原型为

void info_out (struct product prod[L], struct s_person per[M], struct sale sa[N])

info_out()函数用于将产品信息、销售员信息、销售记录信息输出至屏幕上。根据用户的选择，调用 product_out()函数、person_out()函数或 sale_out()函数实现信息的输出。

（9）search_product()

产品信息查询函数。函数参考原型为

void search_product(struct product prod[L])

search_product()函数实现产品信息的查询。提示用户输入要查询产品的代码或名称，逐条记录进行查找，若找到该产品，则输出产品信息至屏幕上，若未找到，则输出提示信息。

（10）search_person()

销售员信息查询函数。函数参考原型为

void search_person(struct s_person per[M])

search_person()函数实现销售员信息的查询。提示用户输入要查询销售员的工号或姓名，逐条记录进行查找，若找到该销售员，则输出销售员信息至屏幕上，若未找到，则输出提示信息。

（11）search_num_sale()

销售记录查询函数。函数参考原型为

void search_num_sale(struct sale sa[N])

search_num_sale()函数实现按照编号查询某条销售记录信息的功能。提示用户输入要查询销售记录的编号，逐条记录进行查找，若找到该销售记录，则输出销售记录信息至屏幕上，若未找到，则输出提示信息。

（12）info_search()

信息查询函数。函数参考原型为

void info_search(struct product prod[L], struct s_person per[M], struct sale sa[N])

info_search()函数是销售信息查询模块，输出提示信息，用户根据提示信息选择查询产品信息、销售员信息或者销售记录信息。通过调用 search_product()函数、search_person()函数或 search_num_sale()函数实现用户选择的功能。

（13）sort_product()

产品信息排序函数。函数参考原型为

void sort_product(struct product prod[L])

sort_product()函数实现产品信息的排序，提示用户输入排序方式，选择按单价排序或按销售额排序，根据用户所选字段，使用一种排序方法（冒泡排序、选择排序等）对产品信息进行排序，排序结果输出至屏幕上。

（14）sort_amount_person()

销售员信息排序函数。函数参考原型为

void sort_amount_person(struct s_person per[M])

sort_amount_person()函数实现按销售员的销售额对销售员信息进行排序的功能，排序结果输出至屏幕上。

（15）record_sort()

信息排序函数。函数参考原型为

void record_sort(struct product prod[L], struct s_person per[M])

record_sort()函数是销售信息排序模块，输出提示信息，用户根据提示信息选择对产品信息排序或者对销售员信息排序。通过调用 sort_product()函数和 sort_amount_person()函数实现用户选择的功能。

（16）info_ to_file()

信息保存函数。函数参考原型为

void info_to_file(struct product prod[L], struct s_person per[M], struct sale sa[N])

info_ to_file()函数用于保存所有销售信息。建立一个数据文件，将完整的产品信息、销售员信息和销售记录信息写入数据文件进行保存。

（17）menu()

显示系统主界面函数。函数参考原型为

void menu()

menu()函数用于显示系统主菜单。用户进入产品销售记录处理系统后，屏幕显示系统主菜单，在菜单中显示系统名称，各功能模块名称及编号。用户根据功能模块名称，输入某一编号，选择执行相应的功能模块。

（18）main()

产品销售记录处理系统主函数。主函数实现数据定义、模块调用等功能，完成整个系统的功能控制。调用 menu()函数显示系统主菜单，提示用户输入选择的功能模块编号，根据编号执行用户选择的功能模块，所选模块功能完成后返回主菜单，由用户继续选择功能模块。若用户输入的编号超出范围，则输出提示信息。

3.3.4 源代码参考框架

1. 预处理

```
#define L 10
#define M 5
#define N 20
#include "stdio.h"
 …
```

2. 子模块定义

```
void menu()
{
     /* 输出产品销售系统的名称;
        输出功能模块名称; */
}
void product_in(struct product prod[L])
{
     /* 变量定义;
        提示用户输入L种产品的信息;
        逐条逐项输入每种产品的信息; */
}
void person_in(struct s_person per[M])
{
     /* 变量定义;
        提示用户输入M位销售员的信息;
        逐条逐项输入每位销售员的信息; */
}
void sale_in(struct sale sa[N])
{
     /* 变量定义;
        提示用户输入N条销售记录的信息;
        逐条逐项输入每条销售记录的信息; */
}
void counting(struct product prod[L],struct s_person per[M],
struct sale sa[N])
{
     /* 变量定义;
        根据销售记录，计算对应产品的销量;
        根据销售记录，计算对应销售员的销售额;
        根据销售记录，计算产品的销售额; */
}
void product_out(struct product prod[L])
{
     /* 变量定义;
        逐条输出产品信息; */
}
void person_out(struct s_person per[M])
{
     /* 变量定义;
        逐条输出销售员信息; */
}
void sale_out(struct sale sa[N])
{
     /* 变量定义;
        逐条输出产品销售记录信息; */
}
void submenu_out()
{
```

```
    /* 输出子菜单名称;
       输出功能模块名称及编号; */
}
void info_out(struct product prod[L],struct s_person per[M],
struct sale sa[N])
{
    /* 变量定义;
       调用submenu_out ()函数输出子菜单;
       提示用户选择菜单项的编号;
       若用户选择输出产品，则调用product_out()函数输出产品信息;
       若用户选择输出销售员，则调用person_out()函数输出销售员信息;
       若用户选择输出销售记录，则调用sale_out()函数输出销售记录信息; */
}
void search_product(struct product prod[L])
{
    /* 变量定义;
       提示用户选择按代码查询还是按名称查询;
       若选择按代码查询，提示用户输入代码，
       若找到则输出信息，若未找到则输出提示;
       若选择按名称查询，提示用户输入名称，
       若找到则输出信息，若未找到则输出提示; */
}
void search_person(struct s_person per[M])
{
    /* 变量定义;
       提示用户选择按工号查询还是按姓名查询;
       若选择按工号查询，则提示用户输入工号，
       若找到则输出信息，若未找到则输出提示;
       若选择按姓名查询，则提示用户输入姓名，
       若找到则输出信息，若未找到则输出提示; */
}
void search_num_sale(struct sale sa[N])
{
    /* 变量定义;
       提示用户输入销售记录编号;
       若找到，则输出该销售记录信息;
       若未找到，则输出提示信息; */
}
void submenu_search()
{
    /* 输出子菜单名称;
       输出功能名称及编号; */
}
void info_search(struct product prod[L],struct s_person per[M],
struct sale sa[N])
{
    /* 变量定义;
       调用submenu_search()函数输出子菜单;
       提示用户选择菜单项的编号;
       若用户选择查询产品，则调用search_product()函数查询产品信息;
```

```
        若用户选择查询销售员，则调用search_person()函数查询销售员信息；
        若用户选择查询销售记录，则调用search_num_sale()函数输出销售记录信息； */
}
void sort_product(struct product prod[L])
{
    /* 变量定义；
       提示用户选择按单价排序还是按销售额排序；
       若选择单价，则按产品单价排序，排序结果输出至屏幕上；
       若选择销售额，则按产品销售额排序，排序结果输出至屏幕上； */
}
void sort_amount_person(struct s_person per[M])
{
    /* 变量定义；
       按销售员的销售额排序，排序结果输出至屏幕上； */
}
void submenu_sort()
{
    /* 输出子菜单名称；
       输出功能名称及编号； */
}
void record_sort(struct product prod[L],struct s_person per[M])
{
    /* 变量定义；
       调用submenu_sort()函数输出子菜单；
       提示用户选择菜单项的编号；
       若用户选择对产品排序，则调用sort_product()函数对产品信息进行排序；
       若用户选择对销售员排序，则调用sort_amount_person()函数对销售员进行排序；*/
}
void info_to_file(struct product prod[L],struct s_person per[M],
struct sale sa[N])
{
    /* 变量定义；
       建立数据文件；
       将所有产品信息逐条记录写入数据文件；
       将所有销售员的信息逐条记录写入数据文件；
       将所有销售记录信息逐条记录写入数据文件； */
}
```

3. 主函数

```
int main()
{
    /* 变量定义；
       调用menu()函数，显示系统主菜单；
       提示用户输入编号；
       根据编号调用函数，执行相应的功能模块； */
}
```

3.3.5 功能测试

为方便测试输入，以 5 种产品信息、5 位销售员信息和 10 条产品销售记录为例，

如表 3-3～表 3-5 所示，将其作为初始信息输入，进行功能测试。

表 3-3　产品信息测试数据

Code	Name	Price	Count	Sum
1	aaa	25	0	0
2	bbb	32	0	0
3	ccc	14	0	0
4	ddd	16	0	0
5	eee	92	0	0

表 3-4　销售员信息测试数据

Code	Name	Amount
1	m001	0
2	m002	0
3	m003	0
4	m004	0
5	m005	0

表 3-5　产品销售记录测试数据

Number	P_code	S_code	Count
1	2	3	5
2	5	3	3
3	4	2	20
4	3	1	30
5	2	4	10
6	1	5	9
7	2	1	24
8	1	5	15
9	4	3	21
10	3	3	30

1）显示系统主菜单，如图 3-24 所示。

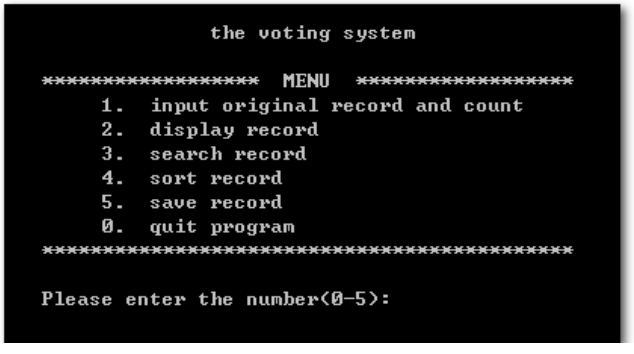


图 3-24　系统主菜单

2）用户选择编号“1”，根据提示从键盘逐项输入产品信息，如图 3-25 所示。

```
Please enter the number(0-5): 1

Please input the information of 5 products:
product1's code: 1
product1's name: aaa
product1's price: 25
product1's count: 0
product1's sum: 0
product2's code:
```

图 3-25　输入产品信息

根据提示逐条输入销售员信息，如图 3-26 所示。

```
Please input the information of 5 salespersons:
salesperson1's code: 1
salesperson1's name: m001
salesperson1's price: 0
salesperson2's code:
```

图 3-26　输入销售员信息

根据提示逐条输入销售记录，如图 3-27 所示。

```
Please input the information of 10 sales records:
record1's number: 1
record1's product_code: 2
record1's person_code: 3
record1's count: 5
record2's number:
```

图 3-27　输入销售记录

输入完成后，返回系统主菜单，继续选择。

3）用户选择编号“2”，输出子菜单，如图 3-28 所示。

```
      output information
***********  MENU  ***********
   1.  output product
   2.  output salesperson
   3.  output sales record
   4.  output all
******************************

Please enter the number(1-4):
```

图 3-28　输出子菜单

用户选择子菜单中的编号“1”，输出所有产品的完整信息，如图 3-29 所示。

```
Please enter the number(1-4): 1

All products:
Code    Name    Price   Count   Sum
1       aaa     25      24      600
2       bbb     32      39      1248
3       ccc     14      60      840
4       ddd     16      41      656
5       eee     92      3       276
```

图 3-29　输出所有产品的信息

若用户输入编号“2”，则输出所有销售员的完整信息；若用户输入编号“3”，则输出所有销售记录信息；若用户输入编号“4”，则输出这三种信息。

返回系统主菜单，继续选择。

4）用户选择编号“3”，输出子菜单，如图 3-30 所示。

```
Please enter the number(0-5): 3

        search record
***********  MENU  ***********
   1.  search product
   2.  search salesperson
   3.  search sales record
******************************

Please enter the number(1-3):
```

图 3-30　查询子菜单

用户选择子菜单中的编号“1”，选择按产品名称查询，输入名称“eee”后，查询结果如图 3-31 所示。

```
Please enter the number(1-3): 1

Search for code(1) or name(2)? enter the number: 2

Please input the name: eee
Search result:
Code    Name    Price   Count   Sum
5       eee     92      3       276
```

图 3-31　按产品名称查询

返回系统主菜单，继续选择。

5）用户选择编号“4”，输出子菜单，如图 3-32 所示。

```
Please enter the number(0-5): 4

      sort record
********  MENU  ********
  1.  sort product
  2.  sort salesperson
************************

Please enter the number(1-2):
```

图 3-32　排序子菜单

用户选择子菜单中的编号“1”，选择按产品单价对产品信息进行排序，排序结果如图 3-33 所示。

```
Please enter the number(1-2): 1

Sort by price(1) or sum(2)? enter the number: 1
Sort result:
Code    Name    Price   Count   Sum
3       ccc     14      60      840
4       ddd     16      41      656
1       aaa     25      24      600
2       bbb     32      39      1248
5       eee     92      3       276
```

图 3-33　按产品单价排序

返回系统主菜单，继续选择。

6）用户选择编号“5”，将信息保存到数据文件中，返回系统主菜单，继续选择，如图 3-34 所示。

```
Please enter the number(0-5): 5
save record finish!

              the voting system

******************  MENU  ******************
     1.  input original record and count
     2.  display record
     3.  search record
     4.  sort record
     5.  save record
     0.  quit program
*********************************************

Please enter the number(0-5):
```

图 3-34　保存信息后返回主菜单

7）在没有输入原始产品信息、销售员信息和产品销售记录的情况下，选择“0”和“1”以外的编号，则给出提示信息，提示用户先进行原始信息的输入，如图 3-35 所示。

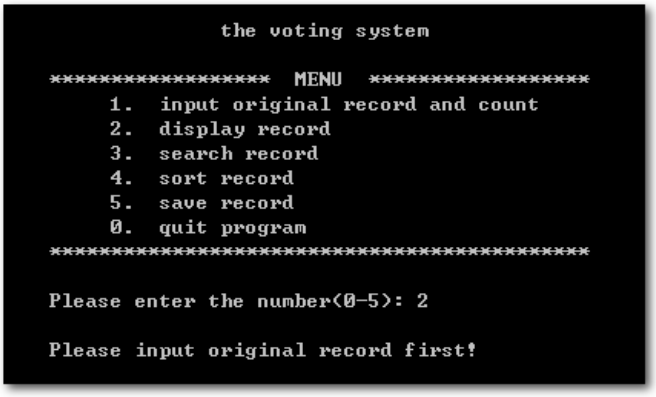


图 3-35　提示先输入原始信息

8）若用户选择编号“0”，则退出系统。若用户输入编号“0”～“5”以外的整数，则给出输入错误的提示。

功能测试时，不必按菜单项编号顺序进行测试，可随机选择编号，执行程序各个功能模块，测试系统各功能模块执行结果是否正确。使用多组不同的产品信息、销售员信息和产品销售记录进行测试，不断修改程序，使其更加完善。

3.3.6　拓展思考

本实验是对产品信息、销售员信息和销售记录的处理，在已有功能的基础上，可以对系统功能进行补充，如扩展以下功能。

1）丰富销售员信息，统计销售员对不同产品的销售情况。

2）添加信息修改功能，允许修改产品单价、销售记录等信息，同步更新相关联的销售员信息等。

3）销售记录中增加产品销售时间信息，统计某段时间内产品的销量，或某段时间内销售员的销售业绩。

4）根据不同销售时间段，对产品信息及销售员信息进行排序，统计销售淡季和旺季的销售情况。

5）增加客户信息，可根据客户代码或名称查询其购买历史，统计其对不同产品的需求变化，以便调整销售策略。

6）根据客户购买历史，划分客户等级，对不同等级的客户，制定相应的销售优惠策略。

实际应用中，有关企业管理信息处理的问题还有很多，例如，产品生产过程中的信息管理、产品销售及售后过程中的信息管理、企业员工信息管理等。更为复杂的企业信息管理需借助数据库管理系统来完成。

3.4 图书管理系统

图书管理系统是单位进行图书管理的必备工具，它能够对海量的图书信息进行规范处理，使管理工作系统化、自动化，提高图书管理工作的效率。图书信息同样需要使用结构体变量来存放，与前面使用结构体数组处理数据不同，图书管理系统主要利用单链表实现。本实验涉及函数、结构体、链表、文件等方面的知识，旨在使读者掌握利用链表处理数据的方法，熟练掌握文件操作，构建综合程序设计的思路及框架，提高综合程序设计的能力。

3.4.1 功能描述与要求

1. 功能描述

设计一个图书管理系统，利用单链表来处理图书信息，要求实现如下系统功能。

（1）输入图书信息

从键盘或数据文件输入图书信息，将其存入单链表。图书信息包括图书编号、图书名称、出版社名称、图书价格、借阅人和借阅标志。

（2）输出图书信息

将单链表中的图书信息输出至屏幕上。

（3）查询图书信息

按图书编号、书名或出版社进行查询。提示用户输入待查图书的编号、书名或出版社，在单链表中逐个结点查找，若找到该图书，则输出图书信息；若未找到，则输出提示信息。

（4）删除图书信息

删除指定图书的信息。提示用户输入待删除图书的编号，在单链表中逐个结点查找，若找到该图书，则删除图书信息；若未找到，则输出提示信息。

（5）修改图书信息

修改指定图书的信息。提示用户输入待修改图书的编号，在单链表中逐个结点查找，若找到该图书，则修改图书信息；若未找到，则输出提示信息。

（6）插入图书信息

插入新的图书信息到指定位置。提示用户输入某个图书编号作为插入点，逐项输入新图书的信息，设置借阅标志为“未借出”，将新的图书信息结点插入到单链表中插入点的后面。

（7）图书排序

按图书编号、书名、出版社或价格进行排序。提示用户输入选择的排序字段，按照用户所选择的字段进行排序，排序结果输出至屏幕上。

（8）图书借阅

借阅指定图书。提示用户输入所借图书的编号，在单链表中逐个结点查找，若未找到该图书，输出提示信息。若找到该图书，就判断该书的借阅标志，若该书已借出，则输出提示；若未借出，则将借阅标志设为“已借出”，输入借阅人编号。输出所借图书的信息至屏幕上。

（9）图书归还

归还指定图书。提示用户输入所归还图书的编号，在单链表中逐个结点查找，找到该图书后，将其借阅标志设为“未借出”，借阅人编号设为“0”。输出所归还图书的信息至屏幕上。

（10）保存图书信息

建立一个数据文件，将图书信息从单链表中写入到数据文件中进行保存。

（11）系统主界面

进入图书管理系统时，输出系统主界面。在主界面中显示系统各功能的名称及编号，用户根据需要选择执行相应的功能模块。

结合以上内容编写源程序，并写出本实验的综合程序设计报告。

2. 要求

（1）源程序编写要求

根据系统功能描述，采用模块化程序设计方法进行程序设计，要求程序结构清晰。图书信息的输入、输出、查询、删除、修改、插入、排序、保存，图书借阅及图书归还等功能，分别用函数实现，在主函数中通过调用这些函数，完成系统功能的要求。代码书写要规范，有简要的注释，给出数据和函数说明。

（2）设计报告撰写要求

设计报告内容包括题目内容和要求、总体设计、详细设计、源代码、调试过程中的问题、总结等。

总体设计：对程序的整体设计思路进行描述，画出图书管理系统的总体功能模块图，说明系统使用的主要数据结构，给出实现图书信息输入输出、查询图书信息、删除图书信息、修改图书信息、插入图书信息、图书信息排序、图书借阅和图书归还等模块的函数设计。

详细设计：分析实现各函数功能的算法，画出函数的算法流程图。

调试过程中的问题：记录程序编写和调试过程中遇到的各种问题，以及解决这些问题的途径和方法。

总结：回顾整个综合程序设计的过程，对学习到的设计方法和思路进行总结，写出个人体会。

设计报告的撰写参见第 4 章。

3.4.2 问题分析

图书信息由多个数据项组成，可以存储在结构体数组中，但系统功能涉及图书信息的插入和删除，使用链表结构存储信息更有利于插入和删除处理。因此，本实验将图书信息存储在单链表中，链表的每个结点是结构体类型，由数据域和指针域构成。数据部分包括图书编号、书名、出版社、价格、借阅人编号及借阅标志共六个成员；指针域存储其直接后继结点的地址。

输入初始图书信息时，每输入一条记录前，要创建一个新的结点，为其分配存储单元，输入数据信息，将新结点加入链表的尾部。新录入的图书还未被借阅，其信息中的借阅人

编号和借阅标志应输入 0。输出图书信息时，从链表的第一个结点开始，输出一条记录后，找到下一个结点，输出信息，直到链表的尾结点。

图书信息的查找、删除、修改、插入、借阅、归还都需要先查找，找到目标结点后再进行相应操作。对图书信息的排序，可采用选择排序、冒泡排序、插入排序等，针对单链表结构的特点，图书管理系统可采用插入排序的方法实现按不同字段的排序。图书借阅时，找到目标结点后，需根据借阅标志判断图书是否借出，再完成后续操作。保存信息后，完整的图书信息被写入数据文件。

3.4.3 总体设计

1. 功能模块设计

根据系统功能描述和问题分析，可将系统功能划分为若干模块，如图 3-36 所示。

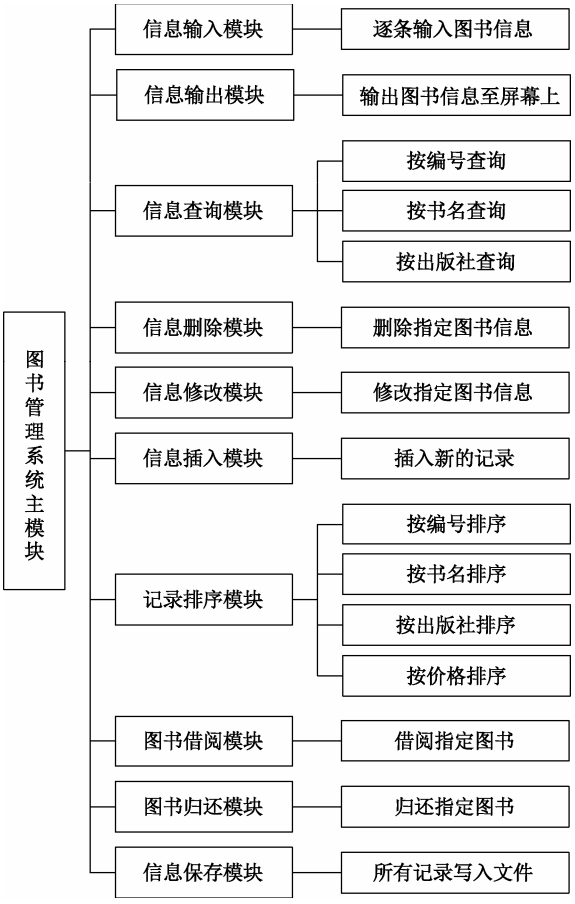


图 3-36　图书管理系统功能模块图

进入图书管理系统，首先显示系统主界面，菜单列出 11 个选项：输入原始图书信息、输出所有图书信息、记录查询、记录删除、记录修改、插入记录、记录排序、图书借阅、图书归还、信息保存和退出程序。各选项编号为 1～10，0 为退出程序选项编号。用户根据

主菜单中显示的功能模块名称，输入编号，选择执行相应的功能。用户必须先选择输入原始图书信息，才可以执行其他模块。所选模块对应的操作执行完后，返回系统主界面，用户可以继续选择其他操作，直至选择编号 0（退出系统）。

2. 数据结构设计

实验内容涉及多条记录的处理，每条记录含有不同类型的数据项，由于要进行记录插入和删除的操作，使用单链表来存放这些记录较为合适。根据功能描述，可将存放图书信息的链表结点，即结构体 book，定义如下。

```
struct book
{
     int num;
     char name[20];
     char press[10];
     int price;
     int reader;
     int flag;
     struct book *next;
};
```

其中各数据域的含义如下。

num：图书编号。

name[20]：图书名称。

press[10]：出版社名称。

price：图书价格。

reader：借阅人编号。

flag：图书借阅标志（若 flag 为 1，则图书已借出；若 flag 为 0，则图书未借出）。

next：单链表中的指针域。

在单链表结构中，每个结点的 next 域为单链表的指针域，用于存储其后继结点的地址；其他域为数据域，存储结点中的数据。可在主函数中定义链表的头结点：

```
struct book *head;
```

将其作为参数传递给输入函数，调用输入函数，创建链表，并返回表头结点指针。主函数将含有初始图书信息的单链表的头结点指针作为函数参数，传递给各子模块，完成后续处理。

3. 函数设计

（1）input_info()

图书信息输入函数。函数参考原型为

struct book *input_info()。

input_info()函数用于从键盘或数据文件输入图书信息，将其存储在单链表中。若从键盘输入数据，则可以给出提示信息，提示用户输入图书编号、书名、出版社、价格、借阅者编号和借阅标志等信息。

（2）display_info()

显示图书信息函数。函数参考原型为

void display_info(struct book *head)。

display_info()函数用于将图书信息逐条输出至屏幕上。

（3）search_num_record()

按编号查询信息函数。函数参考原型为

void search_num_record(struct book *head)。

search_num_record()函数实现按编号查询图书信息的功能。提示用户输入要查询的图书的编号，在单链表中逐个结点查找。若找到，则输出查询到的信息至屏幕上；若未找到，则输出提示信息。

（4）search_name_record()

按书名查询信息函数。函数参考原型为

void search_name_record(struct book *head)。

search_name_record()函数实现按书名查询图书信息的功能。提示用户输入要查询的图书的书名，在单链表中逐个结点查找。若找到，则输出查询到的信息至屏幕；若未找到，则输出提示信息。

（5）search_press_record()

按出版社查询信息函数。函数参考原型为

void search_press_record(struct book *head)。

search_press_record()函数实现按出版社名称查询图书信息的功能。提示用户输入要查询的图书的出版社，在单链表中逐个结点查找。若找到，则输出查询到的信息至屏幕上；若未找到，则输出提示信息。

（6）search_record()

查询图书信息函数。函数参考原型为

void search_record(struct book *head)。

search_record()函数实现图书信息的查询。提示用户选择查询方式，按图书编号查询、按书名查询或按出版社查询。根据用户的输入，选择调用 search_num_record()函数、search_name_record()函数或 search_press_record()函数实现相应的查询功能。

（7）delete_record()

删除图书信息函数。函数参考原型为

struct book *delete_record(struct book *head)。

delete_record()函数用于删除指定编号的图书信息。提示用户输入要删除的图书的编号，在单链表中逐个结点查找。若找到，则从链表中删除该图书结点，并将所删除图书的信息输出至屏幕上；若未找到，则输出提示信息。

（8）modify _record()

修改图书信息函数。函数参考原型为

void modify_record(struct book *head)。

modify _record()函数用于修改图书信息。提示用户输入要修改的图书的编号，在单链表中逐个结点查找。若找到，则提示用户逐项输入修改信息，并将修改后的图书信息输出至屏幕上；若未找到，则输出提示信息。

（9）insert _record()

插入图书信息函数。函数参考原型为

void insert_record(struct book *head)。

insert_record()函数用于插入新的图书信息。提示用户输入一个图书编号作为插入位置，即新图书的信息插入在该图书结点的后面。在单链表中逐个结点查找，若找到，则提示用户逐项输入新图书的信息，将新的图书信息插入在指定结点后面，输出新插入的图书信息至屏幕上；若未找到插入位置，则输出提示信息。

（10）sort_num_record()

按图书编号排序函数。函数参考原型为

struct book *sort_num_record(struct book *head)。

sort_num_record()函数实现按照图书编号对所有图书信息进行排序的功能。使用插入排序法对所有图书信息按图书编号进行排序，排序结果输出至屏幕上。

（11）sort_name_record()

按书名排序函数。函数参考原型为

struct book *sort_name_record(struct book *head)。

sort_name_record()函数实现按照书名对所有图书信息进行排序的功能。使用插入排序法对所有图书信息按书名进行排序，排序结果输出至屏幕上。

（12）sort_press_record()

按出版社名称排序函数。函数参考原型为

struct book *sort_press_record(struct book *head)。

sort_press_record()函数实现按照出版社名称对所有图书信息进行排序的功能。使用插入排序法对所有图书信息按出版社名称进行排序，排序结果输出至屏幕上。

（13）sort_price_record()

按价格排序函数。函数参考原型为

struct book *sort_price_record(struct book *head)。

sort_price_record()函数实现按照图书价格对所有图书信息进行排序的功能。使用插入排序法对所有图书信息按图书价格进行排序，排序结果输出至屏幕上。

（14）submenu_sort()

输出排序子菜单函数。函数参考原型为

void submenu_sort()。

submenu_sort()函数用于输出排序子菜单供用户选择。子菜单中显示四个菜单项：按图书编号排序、按书名排序、按出版社排序、按图书价格排序；其对应的编号分别为 1、2、3、4。

（15）sort_record()

信息排序函数。函数参考原型为

struct book *sort_record(struct book *head)。

sort_record()函数用于按图书信息进行排序。调用 submenu_sort()函数输出子菜单，提示用户选择排序的方式。根据用户输入的编号，选择调用 sort_num_record()函数、sort_name_record()函数、sort_press_record()函数或 sort_price_record()函数实现对应的排序功能。

（16）borrow_book()

图书借阅函数。函数参考原型为

void borrow_book(struct book *head)。

borrow_book()函数实现图书借阅功能。提示用户输入借阅图书的编号及借阅人的编号，根据图书编号进行查找，若找到该图书，判断图书是否已借出，若已借出，则输出提示信息；若未借出，将该图书的借阅标志赋值为 1，为借阅人编号赋值，将所借图书的信息输出至屏幕上。若未找到所借图书，则输出提示信息。

（17）return_book()

图书归还函数。函数参考原型为

void return_book(struct book *head)。

return_book()函数实现图书归还功能。提示用户输入所归还图书的编号，将该图书的借阅标志赋值为 0，并将借阅人编号清零。输出所归还图书的信息至屏幕上。

（18）info_to_file()

信息保存函数。函数参考原型为

void info_to_file(struct book *head)。

info_to_file()函数用于保存所有图书信息。建立一个数据文件，将完整的图书信息写入数据文件进行保存。

（19）menu()

显示系统主界面函数。函数参考原型为

void menu()。

menu()函数用于显示系统主菜单。用户进入图书管理系统后，屏幕显示系统主菜单，在菜单中显示系统名称、各功能模块名称及编号。用户根据功能模块名称，输入某一编号，选择执行相应的功能模块。

（20）main()

图书管理系统主函数。主函数实现数据定义、模块调用等功能，完成整个系统的功能控制。调用 menu()函数显示系统主菜单，提示用户输入选择的功能模块编号，根据编号执行用户选择的功能模块，所选模块功能完成后返回主菜单，由用户继续选择功能模块。若用户输入的编号超出范围，输出提示信息。

3.4.4 源代码参考框架

1. 预处理

```
#include "stdio.h"
#include "string.h"
…
```

2. 子模块定义

```
void menu()
{
      /* 输出图书管理系统名称;
         输出功能模块名称; */
}
struct book *input_info()
```

```
{
    /* 变量定义;
       提示用户输入图书信息;
       为新结点开辟存储单元;
       逐项输入每本图书的信息; */
}
void display_info(struct book *head)
{
    /* 变量定义;
       输出所有图书信息记录; */
}
void search_num_record(struct book *head)
{
    /* 变量定义;
       提示用户输入所查询图书的编号;
       若找到该图书，则输出其图书信息;
       若未找到该图书，则输出提示信息; */
}
void search_name_record(struct book *head)
{
    /* 变量定义;
       提示用户输入所查询图书的书名;
       若找到该图书，则输出其图书信息;
       若未找到该图书，则输出提示信息; */
}
void search_press_record(struct book *head)
{
    /* 变量定义;
       提示用户输入所查询图书的出版社名称;
       若找到该图书，则输出其图书信息;
       若未找到该图书，则输出提示信息; */
}
void submenu_search()
{
    /* 输出查询子菜单名称;
       输出功能模块名称及编号; */
}
void search_record(struct book *head)
{
    /* 变量定义;
       调用submenu_search()函数输出子菜单;
       提示用户选择菜单项的编号;
       若选择按图书编号进行查询，则调用search_num_record()函数查询信息;
       若选择按书名进行查询，则调用search_name_record()函数查询信息;
       若选择按出版社进行查询，则调用search_press_record()函数查询信息; */
}
struct book *delete_record(struct book *head)
{
    /* 变量定义;
       提示用户输入要删除的图书的编号;
       按编号进行查找;
```

```
        若找到，则输出图书信息，删除结点，释放结点空间；
        若未找到，则输出提示信息；  */
}
void modify_record(struct book *head)
{
    /* 变量定义；
        提示用户输入要修改的图书的编号；
        按编号进行查找；
        若找到，则提示用户逐项修改其图书信息，输出修改后的图书信息；
        若未找到，则输出提示信息；  */
}
void insert_record(struct book *head)
{
    /* 变量定义；
        提示用户输入某图书编号，作为插入位置；
        按编号进行查找；
        若找到插入位置，则为新结点开辟存储单元，输入各项信息；
        若未找到插入位置，则输出提示信息；  */
}
struct book *sort_num_record(struct book *head)
{
    /* 变量定义；
        按图书编号对图书信息排序；
        调用display_info()函数输出排序后的图书信息；  */
}
struct book *sort_name_record(struct book *head)
{
    /* 变量定义；
        按书名对图书信息排序；
        调用display_info()函数输出排序后的图书信息；  */
}
struct book *sort_press_record(struct book *head)
{
    /* 变量定义；
        按出版社名称对图书信息排序；
        调用display_info()函数输出排序后的图书信息；  */
}
struct book *sort_price_record(struct book *head)
{
    /* 变量定义；
        按图书价格对图书信息排序；
        调用display_info()函数输出排序后的图书信息；  */
}
void submenu_sort()
{
    /* 输出排序子菜单名称；
        输出功能模块名称及编号；  */
}
struct book *sort_record(struct book *head)
{
    /* 变量定义；
```

```
            调用submenu_sort()函数输出子菜单;
            提示用户选择菜单项的编号;
            若选择按图书编号进行排序，则调用sort_num_record()函数对图书信息排序;
            若选择按书名进行排序，则调用sort_name_record()函数对图书信息排序;
            若选择按出版社名称进行排序，则调用sort_press_record()函数对图书信息排序;
            若选择按图书价格进行排序，则调用sort_price_record()函数对图书信息排序; */
}
void borrow_book(struct book *head)
{
      /* 变量定义;
         提示用户输入所借图书的编号;
         提示用户输入借阅人编号;
         按编号进行查找;
         若找到所借图书，判断是否已借出;
         若已借出，则输出提示信息;
         若未借出，则将借阅标志赋值为1，修改借阅人编号;
         若未找到所借图书，则输出提示信息; */
}
void return_book(struct book *head)
{
      /* 变量定义;
         提示用户输入所还图书的编号;
         若找到该图书，则将其借阅标志赋值为0，借阅人编号赋值为0;
         输出该图书的信息至屏幕上; */
}
void info_to_file(struct book *head)
{
      /* 变量定义;
         建立数据文件;
         将所有图书信息写入数据文件;
         关闭数据文件; */
}
```

3. 主函数

```
int main()
{
      /* 变量定义;
         调用menu()函数，显示系统主菜单;
         提示用户输入编号;
         根据编号调用函数，执行相应的功能模块; */
}
```

3.4.5 功能测试

为方便测试输入，以 5 条图书信息记录为例，如表 3-6 所示，将其作为初始信息输入，进行功能测试。

表 3-6　图书信息测试数据

Number	Name	Press	Price	Reader	Flag
105	food	meishi	25	0	0
303	scenery	yishu	42	0	0
205	campus	jiaoyu	56	0	0
102	culture	wenhua	35	0	0
610	sport	tiyu	36	0	0

1）输出系统主菜单，如图 3-37 所示。

```
                the book management system

**********************  MENU  ************************

   1.  input  record              2.  display record
   3.  search record              4.  delete record
   5.  modify record              6.  insert record
   7.  sort record                8.  borrow book
   9.  return book                10. save record
   0.  quit program

*******************************************************

Please enter the number(0-10):
```

图 3-37　系统主菜单

2）用户选择编号“1”，根据提示从键盘逐项输入图书信息，如图 3-38 所示。

```
Please enter the number(0-10): 1

Please input the information of books:
book's number: 105
book's name: food
book's press: meishi
book's price: 25
book's reader: 0
book's flag: 0
Do you want to input another book(y/n)? y
book's number:
```

图 3-38　输入图书信息

输入完成后，返回系统主菜单，继续选择。

3）用户选择编号“2”，输出所有图书信息，如图 3-39 所示。

```
Please enter the number(0-10): 2

All books:
Number  Name     Press    Price   Reader  Borrow
105     food     meishi   25      0       0
303     scenery  yishu    42      0       0
205     campus   jiaoyu   56      0       0
102     culture  wenhua   35      0       0
610     sport    tiyu     36      0       0
```

图 3-39　输出所有图书信息

返回系统主菜单，继续选择。

4）用户选择编号“3”，输出子菜单，如图 3-40 所示。

```
Please enter the number(0-10): 3

        search record
***********  MENU  ***********
   1.  search by number
   2.  search by book name
   3.  search by press name
******************************

Please enter the number(1-3):
```

图 3-40　查询子菜单

用户选择子菜单中的编号“1”，若继续输入编号 205，则显示该图书信息，如图 3-41 所示。

```
Please enter the number(1-3): 1

Please input the number: 205
Search result:
Number  Name    Press   Price   Reader  Borrow
205     campus  jiaoyu  56      0       0
```

图 3-41　所查询图书

若用户输入编号“2”，则按书名查询图书信息；若用户输入编号“3”，则按出版社查询图书信息。

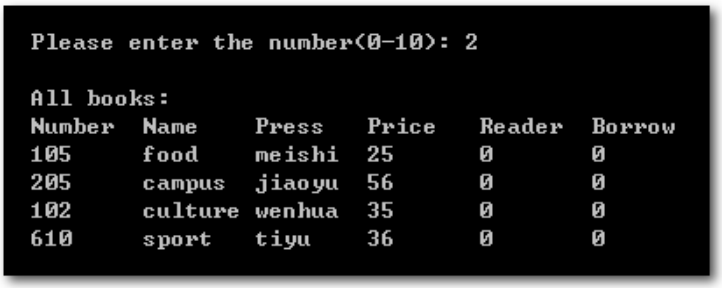
返回系统主菜单，继续选择。

5）用户选择编号“4”，输入图书编号，则显示删除图书的信息，如图 3-42 所示。

```
Please enter the number(0-10): 4

Please input the book number: 303
The deleted record is:
Number  Name    Press   Price   Reader  Borrow
303     scenery yishu   42      0       0
```

图 3-42　所删除图书的信息

返回系统主菜单，继续选择编号“2”，查看删除记录后的图书信息，如图 3-43 所示。

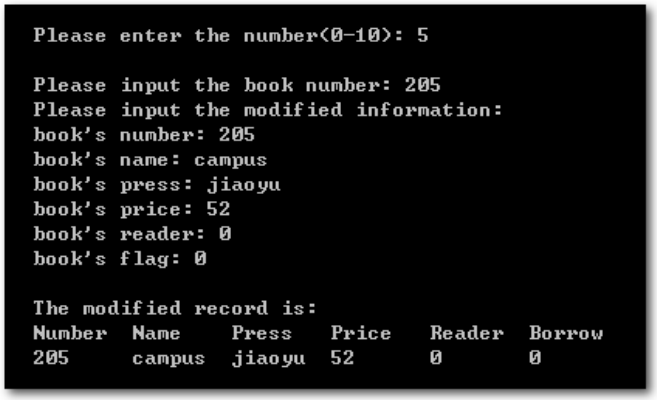


```
Please enter the number(0-10): 2

All books:
Number  Name    Press   Price   Reader  Borrow
105     food    meishi  25      0       0
205     campus  jiaoyu  56      0       0
102     culture wenhua  35      0       0
610     sport   tiyu    36      0       0
```

图 3-43　删除记录后的图书信息

返回系统主菜单，继续选择。

6）用户选择编号“5”，输入待修改的图书的编号“205”，逐项输入修改信息，如图 3-44 所示。

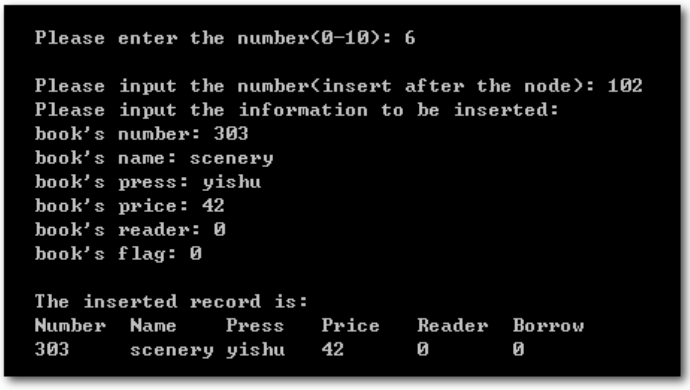


```
Please enter the number(0-10): 5

Please input the book number: 205
Please input the modified information:
book's number: 205
book's name: campus
book's press: jiaoyu
book's price: 52
book's reader: 0
book's flag: 0

The modified record is:
Number  Name    Press   Price   Reader  Borrow
205     campus  jiaoyu  52      0       0
```

图 3-44　修改图书信息

返回系统主菜单，继续选择。

7）用户选择编号“6”，输入插入位置为“102”，逐项输入所插入图书的信息，如图 3-45 所示。



```
Please enter the number(0-10): 6

Please input the number(insert after the node): 102
Please input the information to be inserted:
book's number: 303
book's name: scenery
book's press: yishu
book's price: 42
book's reader: 0
book's flag: 0

The inserted record is:
Number  Name    Press   Price   Reader  Borrow
303     scenery yishu   42      0       0
```

图 3-45　插入图书信息

返回系统主菜单，继续选择编号“2”，查看插入记录后的图书信息，如图 3-46 所示。

```
Please enter the number(0-10): 2

All books:
Number  Name     Press    Price   Reader  Borrow
105     food     meishi   25      0       0
205     campus   jiaoyu   52      0       0
102     culture  wenhua   35      0       0
303     scenery  yishu    42      0       0
610     sport    tiyu     36      0       0
```

图 3-46　插入记录后的图书信息

返回系统主菜单，继续选择。

8）用户选择编号“7”，输出子菜单，如图 3-47 所示。

```
Please enter the number(0-10): 7

           sorting
**********  MENU  **********
   1.  sort by number
   2.  sort by book name
   3.  sort by press
   4.  sort by price
****************************

Please enter the number(1-4):
```

图 3-47　排序子菜单

用户选择子菜单中的编号“2”，按书名对图书信息排序，如图 3-48 所示。

```
Please enter the number(1-4): 2

Sort by name finish!

All books:
Number  Name     Press    Price   Reader  Borrow
205     campus   jiaoyu   52      0       0
102     culture  wenhua   35      0       0
105     food     meishi   25      0       0
303     scenery  yishu    42      0       0
610     sport    tiyu     36      0       0
```

图 3-48　按书名排序

返回系统主菜单，继续选择。

9）用户选择编号“8”，输入要借阅的图书编号为“610”、借阅人编号“222”，如图 3-49 所示。

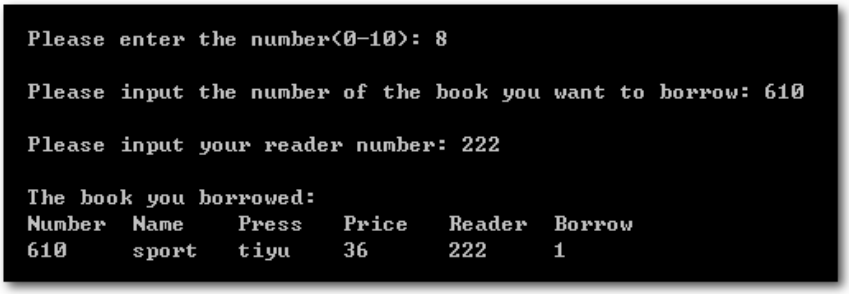


图 3-49　借阅图书

返回系统主菜单，继续选择。

10）用户选择编号“9”，输入所归还的图书编号为“610”，如图 3-50 所示。

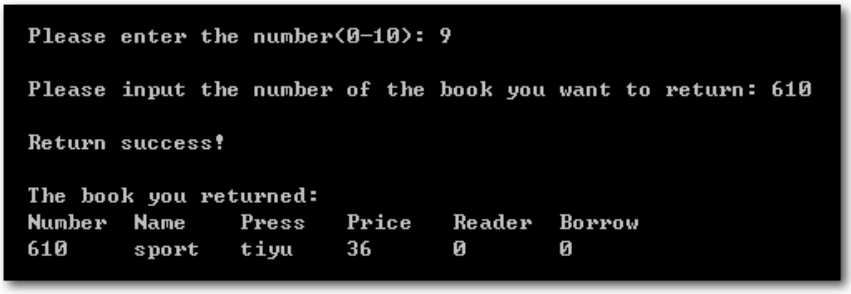


图 3-50　归还图书

返回系统主菜单，继续选择。

11）用户选择编号“10”，将信息保存到数据文件中，返回系统主菜单，继续选择，如图 3-51 所示。

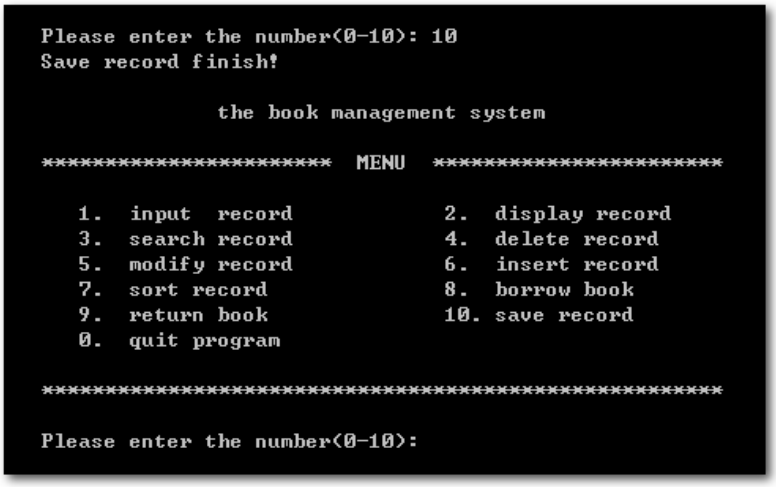


图 3-51　保存信息后返回主菜单

12）在没有输入原始图书信息的情况下，选择“0”和“1”以外的编号，则给出提示信息，提示用户要先输入原始信息，如图 3-52 所示。

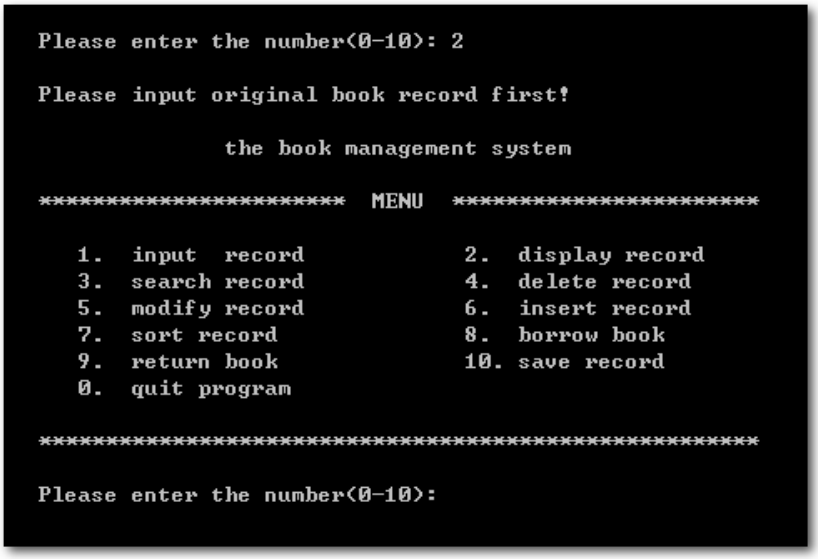


图 3-52　提示先输入原始信息

13）若用户选择编号“0”，则退出系统。若用户输入编号“0”～“10”以外的整数，则给出输入错误的提示。

功能测试时，不必按菜单项编号顺序进行测试，可随机选择编号，执行程序各个功能模块，测试系统各功能模块执行结果是否正确。使用多组不同的图书信息进行测试，不断修改程序，使其更加完善。

3.4.6　拓展思考

本实验利用单链表实现对图书信息的处理，在已有功能的基础上，可以对系统功能进行补充，如扩展以下功能。

1）在图书信息中添加出版时间域，实现查询某个出版时间出版的图书信息，或查询某段时间内出版的图书信息的功能；实现按图书的出版时间对图书信息进行排序的功能。

2）按关键字查询功能。给定某个关键字，实现查询所有书名中包含此关键字的图书的信息。该功能涉及对字符串的子字符串的处理，可参考本章第一个实验进行设计。

3）添加借阅人信息，扩充图书借阅部分的功能。借阅人信息中添加借阅数量域，增加判断借阅人当前借书量是否超出最大借阅数量的功能。借阅人信息中添加图书借阅时间域，用于判断所借图书是否超期。

4）对图书信息进行统计。图书信息中添加库存量域，统计该图书的库存量和借出量等信息。在图书信息中添加类别域，实现统计某一类别的图书信息的功能，如统计艺术类图书的总量，统计教育类图书在某段时间内的借出量等。

在实际应用中，类似图书信息处理的问题还有很多，如车辆信息管理系统、设备信息

管理系统等。此类信息处理系统可利用结构体数组存放数据信息，也可利用链表实现信息处理。

3.5 银行账户管理系统

银行账户管理系统是金融机构进行用户管理的必备工具，它能够对海量的储户信息规范管理，使管理工作系统化、自动化，提高管理工作的效率。本实验实现一个简单的账户管理系统，内容涉及函数、数组、链表、结构体、文件等方面的知识，旨在训练读者的基本编程能力，掌握利用链表处理数据的方法，熟悉文件操作，建立综合程序设计的框架，在解决实际问题的过程中巩固课程知识，训练综合程序设计的能力。

3.5.1 功能描述与要求

1. 功能描述

设计一个银行账户管理系统，利用单链表来处理银行账户信息，要求实现如下系统功能。

（1）输入账户信息

从键盘或数据文件输入账户信息，将其存入单链表。账户信息包括账号、开户人姓名、密码、金额和开户日期。

（2）输出账户信息

将单链表中的账户信息输出至屏幕上。

（3）账户查询

按账号、开户人姓名或开户日期进行查询。提示用户输入待查账户的账号、开户人姓名或开户日期，在单链表中逐个结点查找，若找到该账户，则输出账户信息；若未找到，则输出提示信息。

（4）开户

插入新的账户信息到指定位置。提示用户输入某个账号作为插入点，逐项输入新账户的信息，将新的账户信息结点插入到单链表中插入点的后面。

（5）销户

删除指定账户的信息。提示用户输入销户账号，在单链表中逐个结点查找，若找到该账户，则经密码确认后，将该账户金额清零并删除账户信息；若未找到，则输出提示信息。

（6）修改账户密码

修改指定账户的密码。提示用户输入待修改账户的编号，在单链表中逐个结点查找，若找到该账户，则经密码确认后修改账户密码；若未找到，则输出提示信息。

（7）账户排序

按账号、开户人姓名、金额或开户日期进行排序。提示用户输入选择的排序字段，按照用户所选择的字段进行排序，排序结果输出至屏幕上。

（8）存款

向指定账户存款。提示用户输入存款账户的账号和密码，输入存款金额，在单链表中

逐个结点查找，若找到该账户，则修改账户金额，输出账户信息至屏幕上；若未找到，则输出提示信息。

（9）取款

从指定账户取款。提示用户输入取款账户的账号和密码，输入取款金额，在单链表中逐个结点查找，若找到该账户，则在余额充足的情况下修改账户金额，输出账户信息至屏幕上；若未找到，则输出提示信息。

（10）保存账户信息

建立一个数据文件，将账户信息从单链表写入到数据文件中进行保存。

（11）系统主界面

进入银行账户管理系统时，输出系统主界面。在主界面中显示系统各功能的名称及编号，用户根据需要选择执行相应的功能模块。

结合以上内容编写源程序，并写出本实验的综合程序设计报告。

2. 要求

（1）源程序编写要求

根据系统功能描述，采用模块化程序设计方法进行程序设计，要求程序结构清晰。账户信息的输入、输出、查询、开户、销户、修改密码、排序、存款、取款和保存信息等功能，分别用函数实现，在主函数中通过调用这些函数，完成系统功能的要求。代码书写要规范，有简要的注释，给出数据和函数说明。

（2）设计报告撰写要求

设计报告内容包括题目内容和要求、总体设计、详细设计、源代码、调试过程中的问题、总结等。

总体设计：对程序的整体设计思路进行描述，画出银行账户管理系统的总体功能模块图，说明系统使用的主要数据结构，给出实现账户信息输入输出、查询账户信息、开户、销户、修改密码、账户信息排序、存款和取款等模块的函数设计。

详细设计：分析实现各函数功能的算法，画出函数的算法流程图。

调试过程中的问题：记录程序编写和调试过程中遇到的各种问题，以及解决这些问题的途径和方法。

总结：回顾整个综合程序设计的过程，对学习到的设计方法和思路进行总结，写出个人体会。

设计报告的撰写参见第 4 章。

3.5.2 问题分析

银行账户信息由多个数据项组成，且涉及信息的插入和删除，使用链表结构存储账户信息。链表的每个结点都是结构体类型，由数据域和指针域构成。数据部分包括账号、开户人姓名、账户密码、账户金额和开户日期等；指针域存储其直接后继结点的地址。

输入初始账户信息时，输入一条记录前，要创建一个新的结点，为其分配存储单元，输入数据信息，将新结点加入到链表的尾部。开户日期的格式为 8 位数字，如“20150101”，以便于按开户日期排序。输出账户信息时，从链表的第一个结点开始，输出一条记录后，

找到下一个结点，输出信息，直到链表的尾结点。

账户信息的查找、修改、开户、销户、存款、还款都需要先查找，找到目标结点后再进行相应操作。对账户信息的排序，可采用选择排序、冒泡排序、插入排序等，针对单链表结构的特点，银行账户管理系统可采用插入排序的方法实现按不同字段的排序；由于开户日期以字符串形式存储，按开户日期排序时可使用字符串比较函数确定日期的先后。取款操作时，找到目标结点后，需先判断余额是否充足，再完成后续操作。保存信息后，最终的账户信息被写入数据文件。

3.5.3 总体设计

1. 功能模块设计

根据系统功能描述和问题分析，可将系统功能划分为若干模块，如图 3-53 所示。

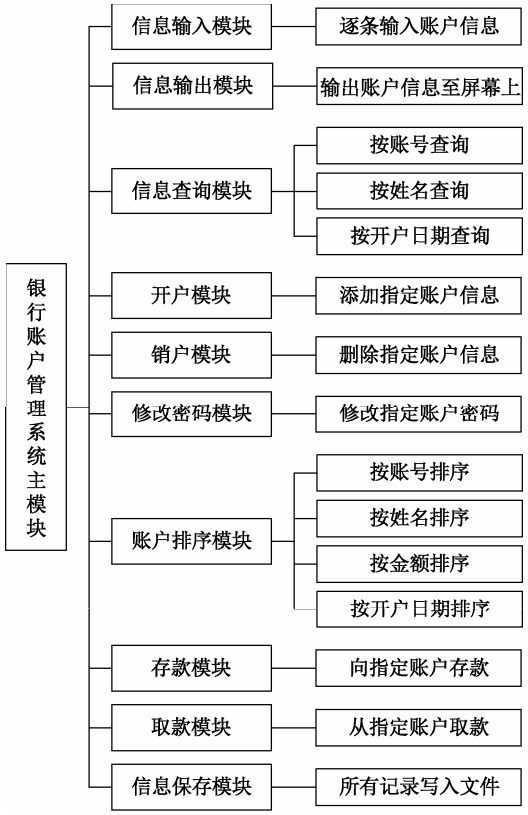


图 3-53　银行账户管理系统功能模块图

进入银行账户管理系统，首先显示系统主界面，菜单列出 11 个选项：输入原始账户信息、输出所有账户信息、账户查询、开户、销户、修改密码、账户排序、存款、取款、信息保存和退出程序。各选项编号为 1～10，0 为退出程序选项编号。用户根据主菜单中显示的功能模块名称，输入编号，选择执行相应的功能。用户必须先输入原始账户信息，才可以执行其他模块。所选模块对应的操作执行完后，返回系统主界面，用户可以继续选择其

他操作，直至选择编号 0（退出系统）。

2. 数据结构设计

实验内容涉及多条记录的处理，每条记录含有不同类型的数据项，由于要进行记录的插入和删除操作，使用单链表来存放这些记录较为合适。根据功能描述，可将存放账户信息的链表结点结构体定义如下。

```
struct account
{
     int num;
     char name[20];
     char password[6];
     int sum;
     char date[8];
     struct account *next;
};
```

其中各数据域的含义如下。

num：账号。

name[20]：开户人姓名。

password[6]：账户密码。

sum：账户金额。

date[8]：开户日期。

next：单链表中的指针域。

在单链表结构中，每个结点的 next 域为单链表的指针域，用于存储其后继结点的地址；其他域为数据域，存储结点中的数据。可在主函数中定义链表的头结点。

```
struct account *head;
```

将其作为参数传递给输入函数，调用输入函数，创建链表，并返回表头结点指针。主函数将含有初始账户信息的单链表的头结点的指针作为函数参数，传递给各子模块，完成后续处理。

3. 函数设计

（1）input_info()

账户信息输入函数。函数参考原型为

struct account *input_info()。

input_info()函数用于从键盘或数据文件输入账户信息，将其存储在单链表中。若从键盘输入数据，则可给出提示信息，提示用户输入账号、开户人姓名、账户密码、账户金额和开户日期等信息。

（2）display_info()

显示账户信息函数。函数参考原型为

void display_info(struct account *head)。

display_info()函数用于将账户信息逐条输出至屏幕上。

（3）search_num_account()

按账号查询信息函数。函数参考原型为

void search_num_account(struct account *head)。

search_num_account()函数实现按账号查询账户信息的功能。提示用户输入要查询的账户的账号，在单链表中逐个结点查找。若找到，则输出查询到的信息至屏幕上；若未找到，则输出提示信息。

（4）search_name_account()

按姓名查询信息函数。函数参考原型为

void search_name_account(struct account *head)。

search_name_account()函数实现按开户人姓名查询账户信息的功能。提示用户输入要查询的账户的开户人姓名，在单链表中逐个结点查找。若找到，则输出查询到的信息至屏幕上；若未找到，则输出提示信息。

（5）search_date_account()

按开户日期查询信息函数。函数参考原型为

void search_date_account(struct account *head)。

search_date_account() 函数实现按开户日期查询账户信息的功能。提示用户输入要查询的账户的开户日期，在单链表中逐个结点查找。若找到，则输出查询到的信息至屏幕；若未找到，则输出提示信息。

（6）search_account()

查询账户信息函数。函数参考原型为

void search_account(struct account *head)。

search_account() 函数实现账户信息的查询。提示用户选择查询方式，按账号查询、按开户人姓名查询或按开户日期查询。根据用户的输入，选择调用 search_num_account()函数、search_name_account ()函数或 search_date_account()函数实现相应的查询功能。

（7）open_account()

开户函数。函数参考原型为

void open_account(struct account *head)。

open_account()函数用于插入新开户的账户信息。提示用户输入一个账号作为插入位置，即新开户的账户信息插入到该账号所在结点的后面。在单链表中逐个结点查找，若找到，则提示用户逐项输入新开户的账户信息，将新开户的账户信息插入在指定结点后面，输出新开户的账户信息至屏幕上；若未找到插入位置，则输出提示信息。

（8）close_account()

销户函数。函数参考原型为

struct account *close_account(struct account *head)。

close_account()函数用于删除指定账号的账户信息。提示用户输入要销户的账户的账号，在单链表中逐个结点查找。若找到，则从链表中删除该账户信息结点，并将所删除账户的信息在屏幕显示；若未找到，则输出提示信息。

（9）change_password()

修改密码函数。函数参考原型为

void change_password(struct account *head)。

change_password()函数用于修改指定账户的密码。提示用户输入要修改密码的账户的账号，在单链表中逐个结点查找。若找到，提示用户输入原密码，若输入正确，则提示用

户输入新密码，将修改密码后的账户信息在屏幕上显示；若原密码输入错误，则输出提示信息。若未找到指定的账户，则输出提示信息。

（10）sort_num_account()

按账号排序函数。函数参考原型为

struct account *sort_num_account(struct account *head)。

sort_num_account()函数实现按照账号对所有账户信息进行排序的功能。使用插入排序法对所有账户信息按账号进行排序，排序结果输出至屏幕上。

（11）sort_name_account()

按姓名排序函数。函数参考原型为

struct account *sort_name_account(struct account *head)。

sort_name_account()函数实现按开户人姓名对所有账户信息进行排序的功能。使用插入排序法对所有账户信息按开户人姓名进行排序，排序结果输出至屏幕上。

（12）sort_sum_account()

按金额排序函数。函数参考原型为

struct account *sort_sum_account(struct account *head)。

sort_sum_account()函数实现按照账户金额对所有账户信息进行排序的功能。使用插入排序法对所有账户信息按账户金额进行排序，排序结果输出至屏幕上。

（13）sort_date_account()

按开户日期排序函数。函数参考原型为

struct account *sort_date_account(struct account *head)。

sort_date_account()函数实现按照开户日期对所有账户信息进行排序的功能。使用插入排序法对所有账户信息按开户日期进行排序，排序结果输出至屏幕上。

（14）submenu_sort()

输出排序子菜单函数。函数参考原型为

void submenu_sort()。

submenu_sort()函数用于输出排序子菜单供用户选择。子菜单中显示四个菜单项：按账号排序、按开户人姓名排序、按账户金额排序、按开户日期排序；其对应的编号分别为 1、2、3、4。

（15）sort_account()

信息排序函数。函数参考原型为

struct account *sort_account(struct account *head)。

sort_account()函数用于对账户信息进行排序。调用 submenu_sort()函数输出子菜单，提示用户选择排序的方式。根据用户输入的编号，选择调用 sort_num_account()函数、sort_name_account()函数、sort_sum_account()函数或 sort_date_account()函数实现对应的排序功能。

（16）deposit()

存款函数。函数参考原型为

void deposit(struct account *head)。

deposit()函数实现向账户存款的功能。提示用户输入存款账号，根据指定的账号进行查找，若找到该账户，则提示用户输入密码，若密码输入正确，则提示用户输入存款金额，

完成存款；若密码输入错误，则输出提示信息。若找不到指定的账户，则输出提示信息。

（17）withdrawal()

取款函数。函数参考原型为

void withdrawal(struct account *head)。

withdrawal()函数实现从账户取款的功能。提示用户输入取款账号，根据指定的账号进行查找，若找到该账户，则提示用户输入密码，若密码输入正确，则提示用户输入取款金额，完成取款；若密码输入错误，则输出提示信息。若找不到指定的账户，则输出提示信息。

（18）info_to_file()

信息保存函数。函数参考原型为

void info_to_file(struct account *head)。

info_to_file()函数用于保存账户信息。建立一个数据文件，将最终的账户信息写入数据文件进行保存。

（19）menu()

显示系统主界面函数。函数参考原型为

void menu()。

menu()函数用于显示系统主菜单。用户进入银行账户管理系统后，屏幕显示系统主菜单，在菜单中显示系统名称，各功能模块名称及编号。用户根据功能模块名称，输入某一编号，选择执行相应的功能模块。

（20）main()

银行账户管理系统主函数。主函数实现数据定义、模块调用等功能，完成整个系统的功能控制。调用 menu()函数显示系统主菜单，提示用户输入选择的功能模块编号，根据编号执行用户选择的功能模块，所选模块功能完成后返回主菜单，由用户继续选择功能模块。若用户输入的编号超出范围，则输出提示信息。

3.5.4 源代码参考框架

1. 预处理

```
#include "stdio.h"
#include "string.h"
…
```

2. 子模块定义

```
void menu()
{
      /*输出银行账户管理系统名称;
        输出功能模块名称;  */
}
struct account *input_info()
{
      /*变量定义;
        提示用户输入账户信息;
```

```
        为新结点开辟存储单元;
        逐项输入每个账户的信息; */
}
void display_info(struct account *head)
{
    /*变量定义;
      输出每条账户信息记录; */
}
void search_num_account(struct account *head)
{
    /*变量定义;
      提示用户输入所查询账户的账号;
      若找到该账户,则输出其账户信息;
      若未找到该账户,则输出提示信息; */
}
void search_name_account(struct account *head)
{
    /*变量定义;
      提示用户输入所查询账户的开户人姓名;
      若找到该开户人的账户,则输出其账户信息;
      若未找到,则输出提示信息; */
}
void search_date_account(struct account *head)
{
    /*变量定义;
      提示用户输入所查询账户的开户日期;
      若找到该账户,输出其账户信息;
      若未找到该账户,输出提示信息; */
}
void submenu_search()
{
    /*输出查询子菜单名称;
      输出功能模块名称及编号; */
}
void search_account(struct account *head)
{
    /*变量定义;
      调用submenu_search()函数输出子菜单;
      提示用户选择菜单项的编号;
      若选择按账号查询,则调用search_num_account()函数查询信息;
      若选择按开户人姓名查询,则调用search_name_account()函数查询信息;
      若选择按开户日期查询,则调用search_date_account()函数查询信息; */
}
void open_account(struct account *head)
{
    /*变量定义;
      提示用户输入某账户的账号,该账号所在结点作为插入位置;
      按指定账号进行查找;
      若找到插入位置,则为新结点开辟存储单元,输入各项信息;
```

```
        若未找到插入位置，则输出提示信息；  */
}
struct account *close_account(struct account *head)
{
     /*变量定义；
       提示用户输入要销户的账号；
       按账号进行查找；
       若找到该账户，则提示用户输入密码；
       若密码输入正确，则输出其账户信息，删除结点，释放结点空间；
       若密码输入错误，则输出提示信息；
       若未找到该账户，则输出提示信息；  */
}
void change_password(struct account *head)
{
     /*变量定义；
       提示用户输入要修改密码的账户的账号；
       按账号进行查找；
       若找到该账户，则提示用户输入原密码；
       若密码输入正确，则提示用户输入新密码，完成修改；
       若密码输入不正确，则输出提示信息；
       若未找到该账户，则输出提示信息；  */
}
struct account *sort_num_account(struct account *head)
{
     /*变量定义；
       按账号对账户信息排序；
       调用display_info()函数输出排序后的账户信息；  */
}
struct account *sort_name_account(struct account *head)
{
     /*变量定义；
       按开户人姓名对账户信息排序；
       调用display_info()函数输出排序后的账户信息；  */
}
struct account *sort_sum_account(struct account *head)
{
     /*变量定义；
       按账户金额对账户信息排序；
       调用display_info()函数输出排序后的账户信息 */
}
struct account *sort_date_account(struct account *head)
{
     /*变量定义；
       按开户日期对账户信息排序；
       调用display_info()函数输出排序后的账户信息；  */
}
void submenu_sort()
{
     /*输出排序子菜单名称；
```

```
        输出功能模块名称及编号；  */
}
struct account *sort_account(struct account *head)
{
    /*变量定义；
      调用submenu_sort()函数输出子菜单；
      提示用户选择菜单项的编号；
      若选择按账号进行排序，则调用sort_num_account()函数对账户信息排序；
      若选择按姓名进行排序，则调用sort_name_account()函数对账户信息排序；
      若选择按账户金额进行排序，
      则调用sort_sum_account()函数对账户信息排序；
      若选择按开户日期进行排序，
      则调用sort_date_account()函数对账户信息排序；  */
}
void deposit(struct account *head)
{
    /*变量定义；
      提示用户输入存款账号；
      按账号进行查找；
      若找到该账户，则提示用户输入密码；
      若密码输入正确，则提示用户输入存款金额，完成存款；
      若密码输入不正确，则输出提示信息；
      若未找到存款账户，则输出提示信息；  */
}
void withdrawal(struct account *head)
{
    /*变量定义；
      提示用户输入取款账号；
      按账号进行查找；
      若找到该账户，提示用户输入密码；
      若密码输入正确，提示用户输入取款金额，完成取款；
      若密码输入不正确，输出提示信息；
      若未找到存款账户，输出提示信息；  */
}
void info_to_file(struct account *head)
{
    /*变量定义；
      建立数据文件；
      将所有账户信息写入数据文件；
      关闭数据文件；   */
}
```

3. 主函数

```
int main()
{
    /*变量定义；
      调用menu()函数，显示系统主菜单；
      提示用户输入编号；
      根据编号调用函数，执行相应的功能模块；  */
}
```

3.5.5 功能测试

为方便测试输入，以 5 条账户信息记录为例，如表 3-7 所示，将其作为初始信息输入，进行功能测试。

表 3-7 账户信息测试数据

Number	Name	Password	Sum	Date
102	zhang	000000	21000	20150102
106	wang	111111	3000	20141201
103	li	222222	5000	20140902
104	qian	333333	35000	20150105
108	sun	444444	12000	20141031

1）输出系统主菜单，如图 3-54 所示。

```
          the bank account management system

************************  MENU  ************************

   1.  input  record              2.  display record
   3.  search record              4.  open an account
   5.  close an account           6.  change password
   7.  sort record                8.  deposit
   9.  withdrawal                 10. save record
   0.  quit program

************************************************************

Please enter the number(0-10):
```

图 3-54 系统主菜单

2）用户选择编号“1”，根据提示从键盘逐项输入账户信息，如图 3-55 所示。

```
Please enter the number(0-10): 1

Please input the information of accounts:
account number: 102
holder's name: zhang
password: 000000
sum: 21000
account opening date: 20150102
Do you want to input another account(y/n)? y
account number:
```

图 3-55 输入账户信息

输入完成后，返回系统主菜单，继续选择。

3）用户选择编号“2”，输出所有账户信息，如图 3-56 所示。

```
Please enter the number(0-10): 2

All accounts:
Number  Name    Password        Sum     Date
102     zhang   000000          21000   20150102
106     wang    111111          3000    20141201
103     li      222222          5000    20140902
104     qian    333333          35000   20150105
108     sun     444444          12000   20141031
```

图 3-56　输出所有账户信息

返回系统主菜单，继续选择。

4）用户选择编号“3”，输出子菜单，如图 3-57 所示。

```
Please enter the number(0-10): 3

            search record
**********  MENU  **********
   1.  search by account number
   2.  search by holder name
   3.  search by account opening date
****************************************

Please enter the number(1-3):
```

图 3-57　查询子菜单

若用户选择子菜单中的编号“1”，继续输入账号“103”，则输出该账户信息，如图 3-58 所示。

```
Please enter the number(1-3): 1

Please input the account number: 103
Search result:
Number  Name    Password        Sum     Date
103     li      222222          5000    20140902
```

图 3-58　输出所查询的账户信息

若用户输入编号“2”，则按开户人姓名查询账户信息；若用户输入编号“3”，则按开户日期查询账户信息。

返回系统主菜单，继续选择。

5）用户选择编号“4”，输入插入位置为“106”，逐项输入新开账户的信息，如图 3-59 所示。

```
Please enter the number(0-10): 4

Please input the number(insert after the node): 106
Please input the information to be inserted:
account number: 125
holder's name: zhao
password: 555555
sum: 15000
account opening date: 20141105

The inserted record is:
Number  Name    Password        Sum     Date
125     zhao    555555          15000   20141105
```

图 3-59　输入新开账户信息

返回系统主菜单，继续选择编号“2”，查看插入记录后的账户信息，如图 3-60 所示。

```
Please enter the number(0-10): 2

All accounts:
Number  Name    Password        Sum     Date
102     zhang   000000          21000   20150102
106     wang    111111          3000    20141201
125     zhao    555555          15000   20141105
103     li      222222          5000    20140902
104     qian    333333          35000   20150105
108     sun     444444          12000   20141031
```

图 3-60　插入记录后的账户信息

返回系统主菜单，继续选择。

6）用户选择编号“5”，输入账号“125”，显示销户信息，如图 3-61 所示。

```
Please enter the number(0-10): 5

Please input the account number: 125

Please input the password: 555555
The deleted record is:
Number  Name    Password        Sum     Date
125     zhao    555555          15000   20141105
```

图 3-61　销户信息

返回系统主菜单，继续选择编号“2”，查看销户后的账户信息，如图 3-62 所示。

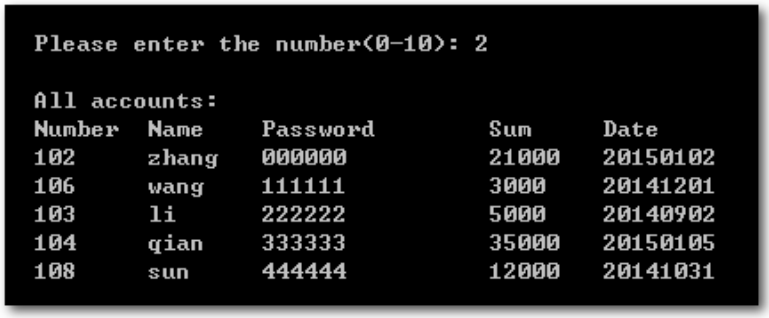


图 3-62　销户后的信息

返回系统主菜单，继续选择。

7）用户选择编号“6”，输入待修改密码的账号“103”，输入原密码及新密码，如图 3-63 所示。

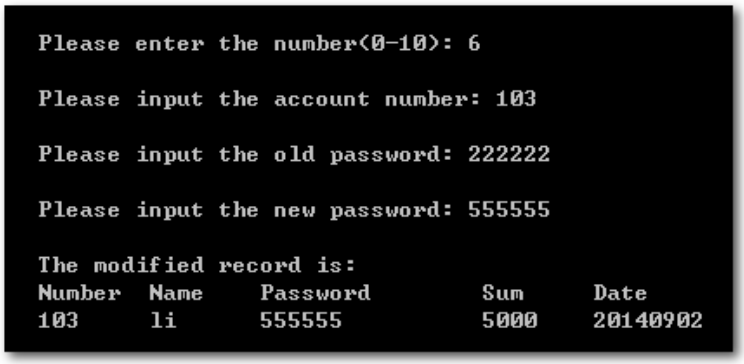


图 3-63　修改密码

返回系统主菜单，继续选择。

8）用户选择编号“7”，输出子菜单，如图 3-64 所示。

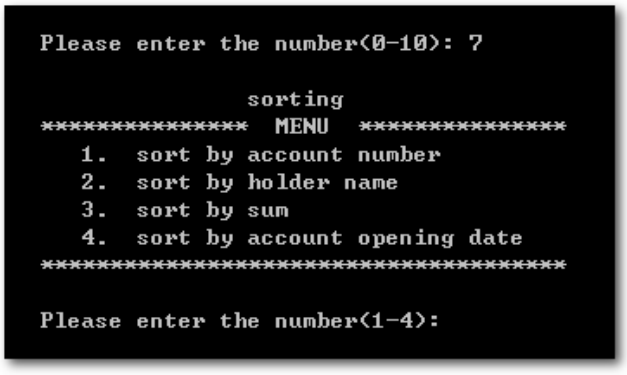


图 3-64　排序子菜单

用户选择子菜单中的编号“2”，按开户人姓名对账户信息进行排序，如图 3-65 所示。

```
Please enter the number(1-4): 2

Sort by holder name finish!

All accounts:
Number  Name    Password        Sum     Date
103     li      555555          5000    20140902
104     qian    333333          35000   20150105
108     sun     444444          12000   20141031
106     wang    111111          3000    20141201
102     zhang   000000          21000   20150102
```

图 3-65　按开户人姓名排序

返回系统主菜单，继续选择。

9）用户选择编号“8”，输入存款账号“102”，输入密码、存款金额，输出该账户的信息，如图 3-66 所示。

```
Please enter the number(0-10): 8

Please input the number of the deposit account: 102

Please input the password: 000000

Please input the amount of deposit: 5000

The deposit account:
Number  Name    Password        Sum     Date
102     zhang   000000          26000   20150102
```

图 3-66　存款信息

返回系统主菜单，继续选择。

10）用户选择编号“9”，输入取款账号“104”，输入密码，输入取款金额，则输出该账户的信息，如图 3-67 所示。

```
Please enter the number(0-10): 9

Please input the number of the drawing account: 104

Please input the password: 333333

Please input the amount of drawing: 5000

The drawing account:
Number  Name    Password        Sum     Date
104     qian    333333          30000   20150105
```

图 3-67　取款信息

返回系统主菜单，继续选择。

11）用户选择编号“10”，将信息保存到数据文件中，返回系统主菜单，继续选择，如图 3-68 所示。

```
Please enter the number(0-10): 10
Save account finish!

          the bank account management system

************************  MENU  ************************

   1.  input  record            2.  display record
   3.  search record            4.  open an account
   5.  close an account         6.  change password
   7.  sort record              8.  deposit
   9.  withdrawal               10. save record
   0.  quit program

*************************************************************

Please enter the number(0-10):
```

图 3-68　保存信息后返回主菜单

12）在没有输入原始账户信息的情况下，选择“0”和“1”以外的编号，会给出提示信息，提示用户先输入原始信息，如图 3-69 所示。

```
Please enter the number(0-10): 3

Please input original account information first!

          the bank account management system

************************  MENU  ************************

   1.  input  record            2.  display record
   3.  search record            4.  open an account
   5.  close an account         6.  change password
   7.  sort record              8.  deposit
   9.  withdrawal               10. save record
   0.  quit program

*************************************************************

Please enter the number(0-10):
```

图 3-69　提示先输入原始信息

13）若用户选择编号“0”，则退出系统。若用户输入编号“0”～“10”以外的整数，则给出输入错误的提示。

功能测试时，不必按菜单项编号顺序进行测试，可随机选择编号，执行程序各个功能模块，测试系统各功能模块执行结果是否正确。使用多组不同的账户信息进行测试，不断修改程序，使其更加完善。

3.5.6 拓展思考

本实验利用单链表实现对银行账户信息的处理，在已有功能的基础上，可以对系统功能进行补充，如扩展以下功能。

1）账户信息中添加状态域，限制账户密码输入错误的次数，若超过允许的密码错误次数，则通过设置账户状态冻结账户。

2）除存储账户信息外，增加存储交易信息，提供统计储户交易信息的功能。例如，查询开户人的所有存款记录或取款记录，查询某段时间内开户人的交易信息等。

3）提供计算利息功能。根据储户存款及取款的时间，按指定计算方法，算出储户取款时应得的利息金额。

4）增加开户人信息，提供查询同一开户人的所有账户信息等功能。

5）根据开户人的开户时间及账户金额等信息，设置 VIP 客户，为 VIP 客户提供较高的管理权限。

实际应用中，类似银行账户信息处理的问题还有很多，如工资管理系统、停车收费管理系统等。此类信息系统涉及记录的频繁变更，适合利用链表来存储数据信息，也可利用结构体数组实现信息处理。

3.6 实训题目

1. 学生证管理系统

设计一个学生证管理系统，实现如下系统功能。

（1）输入学生证信息

从键盘或数据文件输入学生证信息，将其保存在结构体数组或链表中。学生证信息包括学号、姓名、性别、年级和籍贯。

（2）输出学生证信息

将数组或链表中的信息输出至屏幕上。

（3）查询学生证信息

按学号、姓名或年级进行查询。若找到学生证，则输出学生证信息；若未找到，则输出提示信息。

（4）插入学生证信息

插入新的学生证信息到指定位置。提示用户输入某个学号，将新的学生证信息插入到该学号对应的学生证信息的后面。

（5）删除学生证信息

删除指定学生证的信息。提示用户输入待删除学生证的学号，若找到该学生证，则删除学生证信息；若未找到，则输出提示信息。

（6）修改学生证信息

修改指定学生证的信息。提示用户输入待修改学生证的学号，若找到该学生证，则修改学生证信息；若未找到，则输出提示信息。

（7）对学生证信息排序

按学号或姓名对学生证信息排序。提示用户输入排序字段（学号或姓名），按照用户所选择的字段进行排序，输出排序结果。

（8）保存学生证信息

建立一个数据文件，将学生证信息从数组或链表中写入到数据文件中进行保存。

结合以上内容完成综合程序设计，并写出设计报告。

2. 协会会员管理系统

设计一个某协会的会员管理系统，实现如下系统功能。

（1）输入会员信息

从键盘或数据文件输入协会会员信息，将其保存在结构体数组或链表中。协会会员信息包括会员号、姓名、入会日期、会员积分。

（2）输出会员信息

将数组或链表中的会员信息输出至屏幕上。

（3）查询会员信息

按会员号或姓名进行查询。若找到会员，则输出会员信息；若未找到，则输出提示信息。

（4）插入会员信息

插入新的会员信息到指定位置。提示用户输入某个会员号，将新的会员信息插入到该会员号所对应的会员信息的后面。

（5）删除会员信息

删除指定会员的信息。提示用户输入待删除会员的会员号，若找到该会员，则删除会员信息；若未找到，则输出提示信息。

（6）修改会员信息

修改指定会员的信息。提示用户输入待修改会员的会员号，若找到该会员，则修改会员信息；若未找到，则输出提示信息。

（7）对会员信息排序

按会员号或姓名对会员信息排序。提示用户输入排序字段（会员号或姓名），按照用户所选择的字段进行排序，输出排序结果。

（8）保存会员信息

建立一个数据文件，将会员信息从数组或链表中写入到数据文件中进行保存。

结合以上内容完成综合程序设计，并写出设计报告。

3. 交通罚单管理系统

设计一个交通罚单管理系统，实现如下系统功能。

（1）输入罚单信息

从键盘或数据文件输入罚单信息，将其保存在结构体数组或链表中。罚单信息包括罚单号、车牌号、开单交警和开单时间。

（2）输出罚单信息

将数组或链表中的罚单信息输出至屏幕上。

（3）统计罚单信息

统计某一时间段内某个车牌的罚单数量。提示用户输入某车牌号，并输入起始时间和结束时间，统计此时间段内该车的罚单数量，并输出这些罚单的信息。

（4）查询罚单信息

按罚单号或车牌号进行查询。若找到罚单，则输出罚单信息；若未找到，则输出提示信息。

（5）插入罚单信息

插入新的罚单信息到指定位置。提示用户输入某个罚单号，将新的罚单信息插入到该罚单号所对应的罚单信息的后面。

（6）删除罚单信息

删除指定罚单的信息。提示用户输入待删除罚单的罚单号，若找到该罚单，则删除罚单信息；若未找到，则输出提示信息。

（7）罚单排序

按照罚单号对罚单信息排序，输出排序结果。

（8）保存罚单信息

建立一个数据文件，将罚单信息从数组或链表中写入到数据文件中进行保存。

结合以上内容完成综合程序设计，并写出设计报告。

4. 学生作业完成情况管理系统

设计一个学生作业完成情况管理系统，实现如下系统功能。

（1）输入作业完成情况信息

从键盘或数据文件输入作业完成情况信息，将其保存在结构体数组或链表中。作业完成情况信息包括学号、姓名、3 次作业成绩、交作业次数和作业总评成绩。

（2）输出作业完成情况信息

将数组或链表中的作业完成情况信息输出至屏幕上。

（3）查询学生作业完成情况信息

按学号或姓名进行查询。若找到学生作业完成情况信息，则输出作业完成情况信息；若未找到，则输出提示信息。

（4）插入学生作业完成情况信息

在指定位置插入新的学生作业完成情况信息。提示用户输入某个学号，将新的学生作业完成情况信息插入到该学号所对应的学生作业完成情况信息的后面。

（5）删除学生作业完成情况信息

指定一个学号或姓名，在链表中逐个查找，若找到该学生，则删除作业完成情况信息，若未找到，则输出提示信息。

（6）修改学生作业完成情况信息

输入需要修改信息的学号或姓名，在链表中逐个查找，若找到该学生，则修改作业完成情况信息，若未找到，则输出提示信息。

（7）学生作业完成情况信息排序

按交作业次数或作业总评成绩进行排序，提示用户输入排序字段，实现排序，将排序结果输出至屏幕上。

（8）统计学生作业完成情况信息

统计 3 次作业成绩都在 90 分以上的学生作业完成情况信息，输出统计结果至屏幕上。

（9）保存作业完成情况信息

建立一个数据文件，将作业完成情况信息从数组或链表中写入到数据文件中进行保存。

结合以上内容完成综合程序设计，并写出设计报告。

5. 歌曲排行榜

设计一个歌曲排行系统，实现如下系统功能。

（1）输入歌曲相关信息

从键盘或数据文件输入歌曲信息和歌曲关注度信息，将其保存在结构体数组或链表中。歌曲信息包括歌曲编号、曲目名称、歌手和得分。歌曲的关注度信息包括歌曲编号、点击次数、下载次数。所有歌曲按 1、2、3…顺序编号。

（2）统计歌曲得分

根据歌曲的关注度信息统计每首歌曲的得分。歌曲被点击一次得 2 分，被下载一次得 3 分，根据关注度信息中的点击次数和下载次数统计歌曲得分。

（3）输出歌曲相关信息

输出所有歌曲信息及歌曲的关注度信息至屏幕上。建立一个数据文件，将数组或链表中的歌曲信息和歌曲的关注度信息写入数据文件。

（4）查询歌曲信息

按歌曲编号或歌曲名称进行查询。提示用户输入待查询歌曲的编号或名称，若找到歌曲，则输出歌曲信息；若未找到，则输出提示信息。

（5）查询歌曲关注度信息

按歌曲编号查询歌曲的关注度信息。提示用户输入歌曲编号，若找到歌曲关注度信息，则输出关注度信息；若未找到，则输出提示信息。

（6）歌曲排序

按照得分从高到低的顺序对歌曲信息排序，如果得分相同，则按曲目名称递增排序。列出歌曲排行榜，格式如下。

名次　　歌曲编号　　曲目名称　　歌手

将歌曲排行榜输出至屏幕上，并追加写入数据文件。

结合以上内容完成综合程序设计，并写出设计报告。

第 4 章　综合程序设计报告

前面主要介绍了综合程序设计相关的基本概念及详细的开发设计过程，本章将提供一些程序设计报告的书写规范及书写内容，并结合第 2 章的实例，给出一个完整的设计报告样例。

4.1　综合程序设计报告的意义

撰写报告是综合程序设计的一个重要组成部分，报告的完整性是评价综合程序设计质量的重要因素之一。为了强化对书写综合程序设计报告重要性的认识，训练书写报告的能力，必须认真完成综合程序设计报告的撰写，其重要意义如下。

1. 提高可见度

把综合程序设计开发的过程以某种可阅读的形式记录在报告中，这些记录可作为检查综合程序设计质量的依据。

2. 提高开发效率

报告的撰写，使开发人员对各个阶段的工作都进行了周密的思考，全盘权衡，发现错误和不一致性，便于及时纠正。

3. 便于程序维护

报告可提供程序运行及维护的信息，便于各类人员之间的交流与协作，熟悉程序系统，找出并修正错误。

总之，一篇高质量的综合程序设计报告应该满足下列条件。

报告内容完整、正确；

层次清晰、语言流畅、用词准确；

绘制的图表规范；

能够反映结构化程序设计方法的基本原则。

4.2　综合程序设计报告的内容及规范

综合程序设计报告应该能够反映综合程序设计题目的全部完成情况，包括封面、题目内容和要求、总体设计、详细设计（功能模块设计、数据结构设计、函数功能描述）、源代码、调试过程中遇到的问题、总结等。

1. 封面

封面用于说明设计的题目、学生姓名、专业等信息，名称应恰当、准确地反映综合程序设计的研究内容。

2. 题目内容和要求

这部分内容用于叙述综合程序设计的主要内容和要求，文字要准确精炼。

3. 总体设计

根据问题分析，采用某种设计方法，将一个复杂的系统按功能划分成相应的模块，并确定每个模块的功能，模块之间的调用关系，模块之间的接口，即模块之间传递的信息。

4. 功能模块设计

依据前面总体设计的结果，重在描述每个模块的功能要求，说明每部分的算法设计思路（可以是描述算法的流程图）。

5. 数据结构设计

对程序中主要数据的类型及结构进行描述，包括数组、结构体、链表等复合型数据的声明或定义。

6. 源代码

使用结构化方法来编写代码，实现各个模块的功能，重点关注函数的变量和使用的算法，并在适当的地方加上规范的程序注释。

7. 调试过程中遇到的问题

记录在程序调试过程中存在的问题及解决方法。

8. 总结

总结包括程序设计过程的收获及算法的改进设想。

综合程序设计报告示例

河南科技大学

综合程序设计报告

学生成绩信息管理系统

学　　院 _____

年级专业 _____

学生姓名 _____

指导教师 _____

一、题目内容和要求

设计一个小型的成绩信息管理系统，要求能以简便高效的方式对学生成绩进行管理和检索，实现查询、更新、修改、显示、统计、保存信息功能，并且以链表和文件的形式来存放学生信息，功能如下。

1. 创建学生信息

学生信息包括学号、姓名、数学成绩等相关信息。

2. 查询学生信息

输入一个学号或姓名，查找此学生。若找到，则输出此学生的全部信息；若找不到，则输出查找失败的信息。

3. 添加学生信息

要求能实现在文件最后追加学生信息的功能。

4. 插入学生信息

要求能实现在指定位置插入新的学生信息的功能。

5. 修改学生信息

输入一个学号或姓名，查找此学生。若找到，则修改此学生的相关信息；若找不到，则输出查无此学生的信息。

6. 删除学生信息

能实现删除指定的学生信息的功能。

7. 显示学生信息

能实现显示全部学生信息的功能。

8. 统计学生信息

能实现统计学生总分及平均分、优秀人数的功能。

9. 退出系统

要求能将全部学生的信息保存在文件中，系统界面以菜单方式工作，界面友好，易于操作。

二、总体设计

为实现系统功能，本程序主要分为九个模块。它们分别为创建学生信息、查询一条学生记录、追加几条学生信息、在指定位置插入学生信息、修改学生信息、删除学生信息、显示所有学生信息、统计学生信息、保存并退出该系统。其总体设计模块如图 4-1 所示。

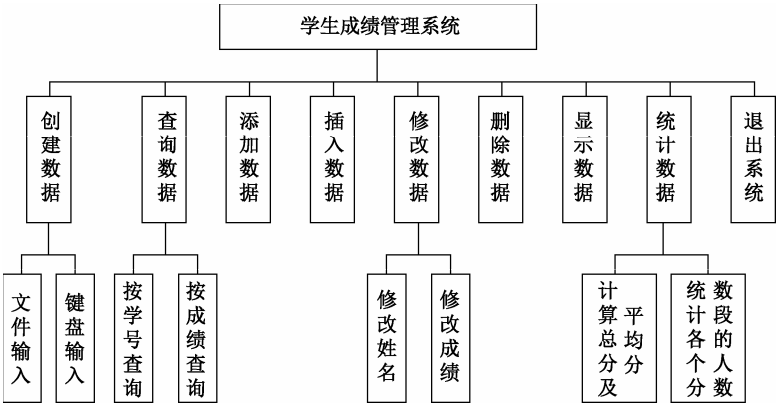


图 4-1　总体设计模块图

三、详细设计

1. 函数设计

图 4-1 中的各模块分别由下列若干函数实现。

Creat（List）函数用于从键盘上录入学生信息；

Read（ ）函数用于从文件中读入学生信息；

Search（List）函数用于对系统中指定信息的查询；

Add（List）函数用于添加学生信息；

Insert（List,int）函数用于在指定位置插入学生信息；

Del（List）函数用于删除系统中指定的学生信息；

Modi（List）函数用于修改系统中指定的学生信息；

ListTra（List）函数用于显示所有学生信息；

Sum（List）函数用于统计学生成绩；

Save（List）函数用于将系统中所有学生信息保存到文件中；

Savea（List）函数用于追加学生信息到文件中。

2. 数据结构

```
typedef struct Node
{
char name[15];
char num[15];
int  math;
struct Node *next;      /*指向结构体类型的指针*/
}Node,*List;

struct Node *p, *q, *r;
```

3. 程序流程图

Search( )函数的流程如图 4-2 所示。

Listinsert( )函数的流程如图 4-3 所示。

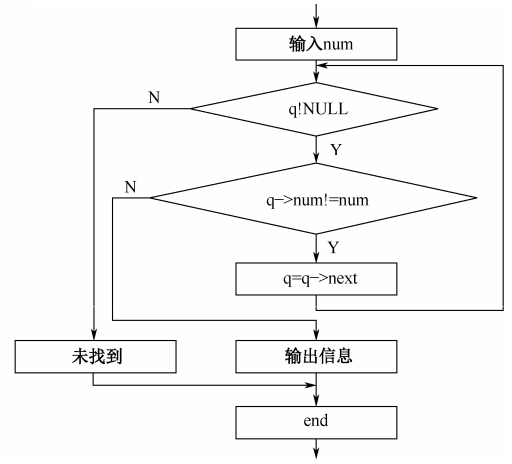


图 4-2　Search( )函数

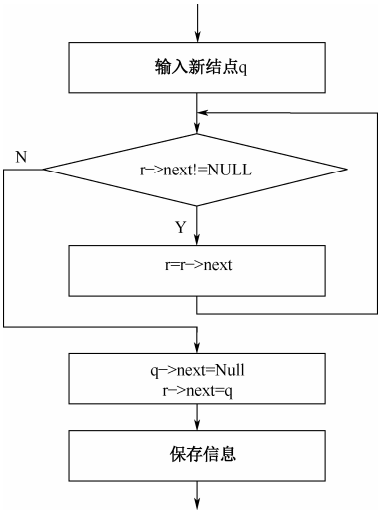


图 4-3　Listinsert ( )函数

Creat( )函数的流程如图 4-4 所示。

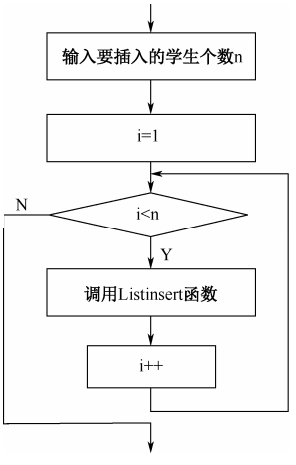


图 4-4　Creat ( )函数

四、源代码

student.c 中的源代码如下。

```
#include <stdio.h>
#include "student.h"
int main( )
{

 struct Stuscore *head;
 int c;
 int flag =1;
 int n;
 head = NULL;
 while ( flag )
 {
     system ("cls") ;
     printf ("\t\t*********学生成绩管理系统**********\n\n") ;
     printf ("***********1 创建学生信息***********\t") ;
     printf ("***********2 查询学生信息***********\n\n") ;
     printf ("***********3 添加学生信息***********\t") ;
     printf ("***********4 插入学生信息***********\n\n") ;
     printf ("***********5 修改学生信息***********\t") ;
     printf ("***********6 删除学生信息***********\n\n") ;
     printf ("***********7 显示学生信息***********\t") ;
     printf ("***********8 统计学生信息***********\n\n") ;
     printf ("\t\t***********0 退出系统***********\n\n") ;
     printf ("请选择操作：") ;
     scanf ("%d",&n) ;
     while (8<n||n<0)
     {
             printf ("输入错误，请重新输入:") ;
             scanf ("%d",&n) ;
     }
     switch ( n )
     {
             case 1:
             {
                 printf ("从文件输入，请选择 1\n") ;
                 printf ("从键盘输入，请选择 0\n") ;
                 scanf ("%d",&c) ;
                 while (c<0||c>1)
                 {
                     printf ("请输入0或1") ;
                     scanf ("%d",&c) ;
                 }
                 if (c==0)
                     head = Creat ( head ) ;//从键盘创建学生信息
```

```
            if (c==1)
                head=Read( );          //从文件读入学生信息
        }
        break;
    case 2:
           head=Read( );
           Search ( head ) ;           //查询
           break;
    case 3:
           head=Read( );
           Add ( head ) ;              //添加
           ListTra ( head ) ;
           break;
    case 4:
           head=Read( );               //插入
           if (head!=NULL)
           {
               ListTra ( head ) ;
               Insert ( head ) ;
               Save ( head ) ;
           }
           break;
    case 5:
           head=Read( );
           ListTra ( head ) ;
           Modi ( head ) ;             //修改
           break;
    case 6:
           head=Read( );
           ListTra ( head ) ;
           head = Del ( head ) ;       //删除并写入文件
           Save ( head ) ;
           break;
    case 7:
           head=Read( );
           ListTra ( head ) ;          //显示
           break;
    case 8:
           head = Read( );
           Sum ( head ) ;              //统计总分及平均分
           break;
    case 0:
    {
        printf ("谢谢使用\n") ;
        getchar( );
        return 0;
    }
```

```
     }
 printf ("按任意键继续:") ;
 getchar( );
 }
 return 0;
}
```

student.h中的代码如下。

```
//头文件student.h源代码

#define OK 1

#include<stdlib.h>
#include<string.h>

#define HEAD1 " ----------------------STUDENT------------------\n"
#define HEAD2 " No|   num      |     name    | math  |next      | \n"
#define HEAD3 "   |-------------|---------------|-------|-------| \n"
#define HEAD4 "|   num      |     name    | math  |next      | \n"
#define HEAD5 "|-------------|---------------|-------|----------| \n"
#define PRIN  "|    %-8s |%-15s|%4d   |%8d   |\n"
#define PRINN "%3d|    %-8s |%-15s|%4d   |%8d   |\n"
#define DAA i,q->num,q->name,q->math,q->next
#define DA  q->num,q->name,q->math,q->next
#define E    "-----------------------------------------------------\n"

/*链表结点信息  */
typedef struct Stuscore
{
     char name[15];
     char num[15];
     int  math;
     struct Stuscore *next;
} Node,*List;

void Save ( List head ) ;/*保存文件*/

//使用尾插法创建链表
struct Stuscore *Listinsert ( List head,int i)  //尾插法
{
 List p,q,r ;  //定义三个指向结构体的指针p、q、r
 q =  (List) malloc (sizeof (Node) ) ;
 p=head;
 if (!q)
 {
     printf ("结点分配空间失败") ;
```

```
        return head;
    }
        printf("请输入第%d个学生姓名:",i);
        scanf("%s",q->name);
        printf("请输入学号:");
        scanf("%s",q->num);
        while( p!= NULL)
        {
            if(p->num == q->num )
            {
                printf("学号重复，请重新输入");
                scanf("%s",q->num);
            }
            else
            p=p->next ;
        }
    printf("请输入数学成绩:");
    scanf("%d",&q->math);
    while( q->math<0 || q->math>100 )
    {
        printf("请输入0～100的成绩");
        scanf("%d",&q->math);
    }
    r=head;
    while( r->next != NULL ) //使r指向最后一个结点
        r = r->next;
    q->next=NULL;    //使插入结点的next为空
    r->next = q;     //使新结点成为最后一个结点
    Save(head);      //输入信息保存到链表中
    return head;
}

//Creat( )函数实现从键盘录入学生信息的功能
struct Stuscore *Creat (  List head )
{
    int number,i;
    head = ( List)malloc(sizeof(Node));
    head->next = NULL;
    printf("请输入要插入的学生个数:");
    scanf("%d",&number);
    for(i=1;i<=number;i++)
    {
        Listinsert(head,i);
    }
    return head;
}
```

```
//ListTra( )函数用于显示记录
void ListTra(List head)
{
 List q;
 FILE *fp;
 int i=1;                          /*用于显示学生序号*/
 fp = fopen("c:\\student","r");/*以只读方式打开二进制文件*/
 if(fp == NULL)
 {
     printf("\n***can not open file !***\n"); /*打开文件失败*/
     getchar( );
     exit( 0 );
 }
 else if(fgetc(fp)==EOF)
 {
     printf("文件为空\n");
    getchar( );
 }
 else
 {
     q=head;
     printf(HEAD1);
     printf(HEAD2);
     printf(HEAD3);
     while( q->next != NULL )
     {
         q=q->next;
         printf(PRINN,DAA);
         i++;
     }
     printf( E );
 }
 puts(" ");
}

//用于保存信息到文件中
void Save( List head )
{
 FILE *fp;
 List q;
 int count = 0;
 fp = fopen("c:\\student","w");/*以只写方式打开二进制文件*/
 if(fp == NULL)
 {
     printf("\n***can not open file !***\n");/*打开文件失败*/
     getchar();
     return ;
```

```
 }
   q = head;
 while (q->next != NULL)
 {
     q=q->next;
     fprintf (fp,"%-15s%-18s%-4d\r\n",q->num,q->name,q->math) ;
     count++;
 }
 if ( count > 0 )
 printf ("\n***保存了%d个信息***\n",count) ;
 fclose (fp) ;
}

//用于追加信息到文件中
void Savea ( List head ) /*以追加方式保存信息到文件*/
{
 FILE *fp;
 List q;
 int count = 0;
 fp = fopen ("c:\\student","a") ;/*以添加方式打开二进制文件*/
 if (fp == NULL)
 {
     printf ("\n***can not open file !***\n") ; /*打开文件失败*/
     getchar( );
     return ;
 }
 q = head;
 while ( q != NULL )
 {
     fprintf (fp,"%-15s%-18s%-4d\r\n",q->num,q->name,q->math) ;
     q=q->next;
     count++;

 }
 getchar( );
 if ( count > 0 )
 printf ("\n***追加了%d个信息***\n",count) ;
 fclose (fp) ;
}

//从文件将链表读入到程序中
List Read( )
{
 FILE *fp;
 List head,end,p;
 int count = 0;
```

```
 head = end = (struct Stuscore*) malloc (sizeof (struct Stuscore) ) ;
 end->next = NULL;
 fp = fopen ("c:\\student","r") ;/*以只读方式打开二进制文件*/
 if (fp == NULL)
 {
     printf ("\n***can not open file !***\n") ; /*打开文件失败*/
     getchar( );
     exit ( 0 ) ;
 }
 else if (fgetc (fp) ==EOF)
 {
     printf ("文件为空，无法读入\n") ;
     getchar( );
     return NULL;
 }
 else
 {
     fseek (fp,0L,SEEK_SET) ;           //定位文件读写指针
     while (!feof (fp) )               //!feof (fp) 值为0，不再执行循环语句
     {
         p= (struct Stuscore*) malloc (sizeof (struct Stuscore) ) ;
         fscanf (fp," %s %s %d \n",p->num,p->name,&p->math) ;
         p->next = NULL;
         end->next = p;
         end = p;
         count++;
     }
     getchar( );
     if ( count > 0 )
     printf ("\n***读取了%d条信息***\n",count) ;
 }
 fclose (fp) ;
 return head;
}

//用于查找学生信息
void Search ( List head )
{
 List q;
 int chengji;
 char nu[15];
 int log=0,sel;
 printf ("1***按成绩查找\n2***按学号查找\n请选择:") ;
 scanf ("%d",&sel) ;
 while (sel<1||sel>2)
 {
```

```
    printf（"请输入1或2:"）;
    scanf（"%d",&sel）;
  }
  if（sel==2）
  {
      printf（"请输入要找的学生学号:"）;
      scanf（"%s",nu）;
  }
  else
  {
      printf（"请输入要查找的学生成绩"）;
      scanf（"%d",&chengji）;
      while（chengji<1||chengji>100）
      {
        printf（"请输入1～100的数字:"）;
        scanf（"%d",&chengji）;
      }

  }
  q=head->next;
  while（ q ）
{
      if（strcmp（q->num,nu）==0||q->math==chengji）
      {
          log=1;
          printf（"找到该学生信息为:\n"）;
          printf（HEAD4）;
          printf（PRIN,DA）;
          getchar( );
          break;
      }
      else
          q=q->next;
  }
  if（log==0）
  {
      printf（"未找到对应学生信息\n"）;
      getchar( );
  }
}

//Del（List）用于删除学生信息
struct Stuscore *Del（ List head ）
{
  char nu[15];
  List q,p;
```

```
 int log=0;                /*标志变量*/
 printf（"请输入删除的学生学号:"）;
 scanf（"%s",nu）;
 q=（List）malloc（sizeof（Node））;
 if（!q）
     printf（"分配结点失败"）;
 p=head;
 q=head->next ;
 while（ q ）
 {
     if（strcmp（q->num,nu）==0）
     {
         log=1;
         p->next=q->next;
         free（q）;
         printf（"删除成功\n"）;
         getchar( );
         break;
     }
     else
     {
         p=q;
         q=q->next;
     }

 }
 if（log==0）
 {
     printf（"要删除的学生不存在\n"）;
     getchar( );
 }
 return head;
}

//Add（List）添加学生函数
void Add（ List head ）
{
 int n,i;
 List q,r ,test,p;
 printf（"请输入要插入的学生个数:"）;
 scanf（"%d",&n）;

 r=head;
 while（ r->next != NULL ）/*找到尾结点并保留它*/
     r = r->next;
 p = r;                    /*尾结点*/
 for（i=1;i<=n;i++）
```

```
    {
        q = (List)malloc(sizeof(Node));
        if(!q)
        {
            printf("结点分配空间失败");
            return ;
        }
        printf("请输入学号:");
        scanf("%s",q->num);

        test=head;              /*检测学号是否重复*/
        while( test!= NULL)
        {
            if(strcmp(test->num ,q->num) == 0 )
            {
                printf("学号重复，请重新输入");
                scanf("%s",q->num);
                test = head;
            }
            else
                test=test->next ;
        }
        printf("请输入第%d个学生姓名:",i);
        scanf("%s",q->name);

        printf("请输入数学成绩:");
        scanf("%d",&q->math);
        while( q->math<0 || q->math>100 )
        {
            printf("请输入0-100之间的成绩");
            scanf("%d",&q->math);
        }

        q->next=NULL;       //新结点指针域为空
        r->next = q;        //上次的尾指针指向新结点
        r=q;                //使尾指针等于新结点，成为本次的尾指针
    }
    Savea( p->next );       //输入信息保存到链表中
}


//Modi(List)修改学生信息
void Modi( List head )
{
    char nu[15];
    int log=0;
    List q,p;
```

```
    printf（"请输入修改的学生学号:"）;
    scanf（"%s",nu）;
    q=（List）malloc（sizeof（Node））;
    if（!q）
    {
        printf（"结点分配空间失败"）;
        return ;
    }
    p=head;
    q=head->next;
    while（ q ）
    {
        if（strcmp（q->num,nu）==0）
        {
            log=1;
            system（"cls"）;
            printf（"学生的信息为:\n"）;
            printf（HEAD4）;
            printf（PRIN,DA）;
            printf（"请输入学生新姓名:"）;
            scanf（"%s",q->name）;

            printf（"请输入新数学成绩:"）;
            scanf（"%d",&q->math）;
            while（ q->math<0 || q->math>100 ）
            {
                printf（"请输入0～100之间的成绩"）;
                scanf（"%d",&q->math）;
            }
            printf（"修改后学生的信息为:\n"）;
            printf（HEAD4）;
            printf（PRIN,DA）;
            q=q->next;
        }
        else
            q=q->next;
    }
    if（log==0）
    {
        printf（"修改的学生不存在\n"）;
    }
    Save（ head ）;
}

//Insert（List）用于插入学生信息
void Insert（ List head ）
{
```

```
List q,p,r,newnode,test;
int number,count=0,i=1;
int log=0;
r= (List) malloc (sizeof (Node) ) ;
r=head->next;
p=head;
while (p->next)      /*此循环计算链表中记录的条数*/
{
    p=p->next;
    count++;
}

printf ("请输入要插入位置的序号:") ;
scanf ("%d",&number) ;
while (number<i || number >count+1)
{
    printf ("序号无效，请重新输入要插入位置的序号:") ;
    scanf ("%d",&number) ;
}

while (i<number)         /*确定插入位置在q和r之间*/
{
    q=r;
    r=r->next;
    i++;
}

newnode = (List) malloc (sizeof (Node) ) ;
if (!newnode)
{
    printf ("结点分配空间失败") ;
    return ;
}
printf ("请输入待插入学生的学号:") ;
scanf ("%s",newnode->num) ;
test=head ;
while ( test!= NULL)   /*用于检测插入的学号是否已存在*/
{
    if (strcmp (test->num ,newnode->num) == 0 )
    {
        printf ("学号重复，请重新输入") ;
        scanf ("%s",newnode->num) ;
        test = head;
    }
    else
        test=test->next ;
}
printf ("请输入待插入学生的姓名:") ;
scanf ("%s",newnode->name) ;
printf ("请输入待插入学生的数学成绩:") ;
```

```
    scanf ("%d",&newnode->math) ;
    while ( newnode->math<0 || newnode->math>100 )
    {
        printf ("请输入0～100的成绩") ;
        scanf ("%d",&newnode->math) ;
    }
    if (number==1)      /*插入到第一个位置*/
    {
        newnode->next = head->next;
        head->next = newnode;
    }
    else
    {
        newnode->next=r;
        q->next = newnode;
    }
    getchar( );
    Save ( head ) ;
}

// Sum (List) 用于求和及平均分
void Sum ( List head)
{

  float sum,average;
  int n;          /*n用于统计学生人数*/
  List p;
  sum = 0,n = 0;
  p = (List) malloc (sizeof (Node) ) ;

  p = head;
  while (p->next!=NULL)
  {
      p = p->next;
      sum += p->math;
      n++;
  }

  average = sum/n;
  printf ("the sum of math is %f\n",sum) ;
  printf ("the average of math is %.2f\n",average) ;

}
```

五、调试过程中的问题

本程序的调试总体上比较顺利，但其中也出现了如下问题。

出现的警告如下。

1）函数使用了未经初始化的变量，或定义变量了却没有使用。

2）函数原型有返回值，但函数体内没有 return 值，反之亦然。例如，int main（void）

{
……，或者
}

```
void main( )
{
   return 0;
}
```

3）在使用字符变量时，由于回车键也属于字符，所以在下一个变量从终端读取字符时会出错。通过单步调试发现该问题，于是考虑在适当的地方用 getchar()消除回车符。

4）文件写入部分虽然保存了信息，但是第二次写入时，覆盖了之前的内容，不能正确写入信息，经查，发现文件读写方式不正确。

出现的错误如下。

1）语句缺少“;”号。

2）括号不配对。当一个语句中使用多层次括号时常出现这类错误，如 while（（ c = getchar( ) != ’ #’）putchar（c）;。

3）所调用的函数在调用语句之后才定义，并且在调用之前未声明。

4）函数声明与函数定义不匹配。

5）需要加头文件时没有用#include 命令包含头文件。例如，程序中用到 fabs 函数，没有用#include<math.h> 语句。

6）误认为形参值的改变会影响实参的值。

7）函数的实参和形参类型不一样。

8）没有注意函数参数的求值顺序。

9）混淆了结构体类型与结构体变量的区别，对一个结构体类型赋值。

10）使用文件时忘记打开，或打开方式与使用情况不匹配。

11）使用了 fgetc（fp）函数，但忘记用 fseek（fp,0L,SEEK_SET）函数重新定位文件指针，导致文件总是读写错误。

六、总结

在本次综合程序设计中，我们首先对系统的整体功能进行了构思，然后用结构化分析方法进行了分析，将整个系统清楚地划分为若干个模块，并使用程序流程图来描述模块的功能，再根据每个模块的功能编写代码。

在函数的编写过程中，不仅用到了 for 循环、while 循环和 switch 语句，还用到了函数之间的调用及链表和文件的相关操作。调试程序时，先将一个个函数调试成功后，再组装成所需要的模块。

程序还有一些不完善的地方，如当输入的数据不符合定义的数据格式时，程序会出现一些错误，有时会出现主菜单的死循环。

通过程序设计实训，巩固了以前所学的程序设计语言的基础知识，掌握了更多的综合程序设计的实用技巧和技能。

附录 A　全国计算机等级考试二级 C 语言模拟试题及答案

一、选择题（每小题 1 分，共 40 分）

1．下列关于栈叙述正确的是（　　）。

A．栈顶元素最先能被删除　　B．栈顶元素最后被删除

C．栈底元素永远不能被删除　　D．以上三种说法都不对

2．下列叙述中正确的是（　　）。

A．有一个以上根结点的数据结构不一定是非线性结构

B．只有一个根结点的数据结构不一定是线性结构

C．循环链表是非线性结构

D．双向链表是非线性结构

3．某二叉树共有 7 个结点，其中叶子结点只有 1 个，则该二叉树的深度为（　　）（假设根结点在第 1 层）。

A．3　　B．4　　C．6　　D．7

4．在软件开发中，需求分析阶段产生的主要文档是（　　）。

A．软件集成测试计划　　B．软件详细设计说明书

C．用户手册　　D．软件需求规格说明书

5．结构化程序所要求的基本结构不包括（　　）。

A．顺序结构　　B．GOTO 跳转

C．选择（分支）结构　　D．重复（循环）结构

6．下面叙述中错误的是（　　）。

A．系统总体结构图支持软件系统的详细设计

B．软件设计是将软件需求转换为软件表示的过程

C．数据结构与数据库设计是软件设计的任务之一

D．PAD 图是软件详细设计的表示工具

7．负责数据库中查询操作的数据库语言是（　　）。

A．数据定义语言　　B．数据管理语言

C．数据操纵语言　　D．数据控制语言

8．一个教师可讲授多门综合程序设计，一门综合程序设计可由多个教师讲授，则实体教师和综合程序设计间的联系是（　　）。

A．1:1 联系　　B．1:m 联系

C．m:1 联系　　D．m:n 联系

9．有三个关系 R、S 和 T 如下：

R

A	B	C
a	1	2
b	2	1
c	3	1

S

A	B	C
d	3	2

T

A	B	C
a	1	2
b	2	1
c	3	1
d	3	2

则由关系 R 和 S 得到关系 T 的操作是（　　）。

A．选择　　B．投影　　C．交　　D．并

10．定义无符号整数类为 Uint，下面可以作为类 Uint 实例化值的是（　　）。

A．−369　　B．369

C．0.369　　D．整数集合{1,2,3,4,5}

11．计算机高级语言程序的运行方法有编译执行和解释执行两种，以下叙述中正确的是（　　）。

A．C 语言程序仅可以编译执行

B．C 语言程序仅可以解释执行

C．C 语言程序既可以编译执行又可以解释执行

D．以上说法都不对

12．以下叙述中错误的是（　　）。

A．C 语言的可执行程序是由一系列机器指令构成的

B．用 C 语言编写的源程序不能直接在计算机上运行

C．通过编译得到的二进制目标程序需要连接才可以运行

D．在没有安装 C 语言集成开发环境的机器上不能运行 C 源程序生成的.EXE 文件

13．以下选项中不能作为程序合法常量的是（　　）。

A．1,234　　B．'\123'　　C．123　　D．"\x7A"

14．以下选项中可作为程序合法实数的是（　　）。

A．.1e0　　B．3.0e0.2　　C．E9　　D．9.12E

15．若有定义语句 int a=3,b=2,c=1; ，则下列选项中错误的赋值表达式是（　　）。

A．a=（b=4）=3;　　B．a=b=c+1;

C．a=（b=4）+c;　　D．a=1+（b=c=4）;

16．有以下程序段：

```
char name[20];
int num;
scanf（"name=%snum=%d",name,&num）;
```

当执行上述程序段，并从键盘输入“name=Lili num=1001<回车>”后，name 的值为（　　）。

A．Lili　　B．name=Lili

C．Lilinum=　　D．name=Lili num=1001

17．if 语句的基本形式是 if（表达式）语句，以下关于“表达式”值的叙述中正确的是（　　）。

A．必须是逻辑值　　B．必须是整数值

C．必须是正数　　D．可以是任意合法的数值

18．有以下程序：

```
#include <stdio.h>
void main( )
{
     int x=011;
     printf ("%d\n",++x) ;
}
```

程序运行后的输出结果是（　　）。

A．12　　B．11　　C．10　　D．9

19．有以下程序：

```
#include <stdio.h>
void main( )
{
     int s;
     scanf("%d",&s);
     while(s>0)
     {
          switch(s)
          {
               case 1:printf("%d",s+5);
               case 2:printf("%d",s+4);break;
               case 3:printf("%d",s+3);
               default:printf("%d",s+1);break;
          }
          scanf("%d",&s);
     }
}
```

运行时，若输入“1 2 3 4 5 0<回车>”，则输出结果是（　　）。

A．6566456　　B．66656　　C．66666　　D．6666656

20．有以下程序段：

```
int i,n;
for(i=0;i<8;i++)
{
     n=rand()%5;
     switch (n)
     {
          case 1:
          case 3:printf("%d\n",n); break;
          case 2:
```

```
            case 4:printf("%d\n",n);continue;
            case 0:exit(0);
        }
        printf("%d\n",n);
    }
```

以下关于程序段执行情况的叙述，正确的是（　　）。

A．for 循环语句固定执行 8 次

B．当产生的随机数 n 为 4 时结束循环操作

C．当产生的随机数 n 为 1 和 2 时不做任何操作

D．当产生的随机数 n 为 0 时结束程序运行

21．有以下程序：

```
#include <stdio.h>
void main( )
{
    char s[]="012xy\08s34f4w2";
    int i,n=0;
    for(i=0;s[i]!=0;i++)
    if(s[i]>='0'&&s[i]<='9')
        n++;
    printf("%d\n",n);
}
```

程序运行后的输出结果是（　　）。

A．0　　B．3　　C．7　　D．8

22．若 i 和 k 都是 int 类型变量，有以下 for 语句：

```
for (i=0,k=-1; k=1; k++)
  printf ("*****\n") ;
```

则下面关于语句执行情况的叙述中正确的是（　　）。

A．循环体执行两次　　B．循环体执行一次

C．循环体一次也不执行　　D．构成无限循环

23．有以下程序：

```
#include <stdio.h>
void main( )
{
    char b,c;int i;
    b='a'; c='A';
    for(i=0;i<6;i++)
    {
        if(i%2)
            putchar(i+b);
        else
            putchar(i+c);
    }
    printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．ABCDEF　　B．AbCdEf　　C．aBcDeF　　D．abcdef

24．设有定义：double x[10], *p=x;，以下使数组 x 下标为 6 的元素读入数据的正确语句是（　　）

A．scanf（"%f",&x[6]）;　　B．scanf（"%lf",*（x+6））;

C．scanf（"%lf",p+6）;　　D．scanf（"%lf",p[6]）;

25．有以下程序（说明，字母 A 的 ASCII 码值是 65）：

```
#include <stdio.h>
void fun(char *s)
{
     while(*s)
     {
         if(*s%2)
             printf("%c",*s);
         s++;
     }
}
void main( )
{
     char a[ ]="BYTE";
     fun(a);
     printf("\n");
}
```

程序运行后的输出结果是（　　）。

A．BY　　B．BT　　C．YT　　D．YE

26．有以下程序段：

```
#include <stdio.h>
void main( )
{
  …
  while ( getchar()!='\n') ;
  …
}
```

以下叙述中正确的是（　　）。

A．此 while 语句将无限循环

B．getchar( )不可以出现在 while 语句的条件表达式中

C．当执行此 while 语句时，只有按回车键程序才能继续执行

D．当执行此 while 语句时，按任意键程序能继续执行

27．有以下程序：

```
#include <stdio.h>
void main( )
{
     int x=1,y=0;
```

```
        if(!x)
            y++;
        else if(x==0)
            if (x)  y+=2;
            else  y+=3;
        printf("%d\n",y);
    }
```

程序运行后的输出结果是（　　）。

A．3　　B．2　　C．1　　D．0

28．若有定义语句：char s[3][10],（*k）[3],*p;，则以下赋值语句正确的是（　　）。

A．p=s;　　B．p=k;　　C．p=s[0];　　D．k=s;

29．有以下程序：

```
#include <stdio.h>
void fun( char *c )
{
    while( *c )
    {
        if(*c>='a'&&*c<='z')
            *c=*c-('a'-'A');
        c++;
    }
}
void main( )
{
    char s[81];
    gets(s);
    fun(s);
    puts(s);
}
```

当执行程序时，从键盘上输入“HelloBeijing<回车>”，则程序的输出结果是（　　）。

A．hello beijing　　B．Hello Beijing

C．HELLO BEIJING　　D．hELLO Beijing

30．有以下程序：

```
#include <stdio.h>
#include <string.h>
void main( )
{
    char a[10]= “abcd”;
    printf(“%d,%d\n”,strlen(a),sizeof(a));
}
```

程序运行后的输出结果是（　　）。

A．7,4　　B．4,10　　C．8,8　　D．10,10

31．有以下程序：

```
#include <stdio.h>
void main( )
{
     char a[30],b[30];
     scanf("%s",a);
     gets(b);
     printf("%s\n %s\n",a,b);
}
```

程序运行时若输入

```
how areyou? I am fine<回车>
```

则输出结果是（　　）。

A．how areyou?　　B．howI am fine are you? I am fine

C．how areyou? I am fine　　D．how are you?

32．设有如下函数定义

```
int fun（int k）
{
    if（k<1） return 0;
    else if（k==1） return 1;
    else returnfun（k-1）+1;
}
```

若执行调用语句 n=fun（3）;，则函数 fun 总共被调用的次数是（　　）。

A．2　　B．3　　C．4　　D．5

33．有以下程序：

```
#include <stdio.h>
int fun (int x,int y)
{
     if(x!=y)
         return ((x+y)/2);
     else
         return (x);
}
void main( )
{
     int a=4,b=5,c=6;
     printf("%d\n",fun(2*a,fun(b,c)));
}
```

程序运行后的输出结果是（　　）。

A．3　　B．6　　C．8　　D．12

34．有以下程序：

```
#include <stdio.h>
int fun( )
{
     static int x=1;
     x*=2;
     return x;
```

```
}
void main( )
{
     int i,s=1;
     for(i=1;i<=3;i++)
         s*=fun( );
     printf("%d\n",s);
}
```

程序运行后的输出结果是（　　）。

A．0　　B．10　　C．30　　D．64

35．有以下程序：

```
#include <stdio.h>
#define S(x)4*(x)*x+1
void main( )
{
     int k=5,j=2;
     printf("%d\n",S(k+j));
}
```

程序运行后的输出结果是（　　）。

A．197　　B．143　　C．33　　D．28

36．设有定义：

```
struct
{
     char mark[12];
     int num1;
     double num2;
} t1,t2;
```

若变量均已正确赋初值，则以下语句中错误的是（　　）。

A．t1=t2;　　B．t2.num1=t1.num1;

C．t2.mark=t1.mark;　　D．t2.num2=t1.num2;

37．有以下程序：

```
#include <stdio.h>
struct ord
{
     int x,y;
}dt[2]={1,2,3,4};
void main( )
{
     struct ord *p=dt;
     printf ("%d,",++ (p->x) ) ;
     printf ("%d\n",++ (p->y) ) ;
}
```

程序运行后的输出结果是（　　）。

A．1,2　　B．4,1　　C．3,4　　D．2,3

38．有以下程序：

```
#include  <stdio.h>
struct S
{
     int a,b;
}data[2]={10,100,20,200};
void main( )
{
     struct S p=data[1];
     printf("%d\n",++(p.a));
}
```

程序运行后的输出结果是（　　）。

A．10　　B．11　　C．20　　D．21

39．有以下程序：

```
#include  <stdio.h>
void main( )
{
     unsigned char a=8,c;
     c=a>>3;
     printf("%d\n",c);
}
```

程序运行后的输出结果是（　　）。

A．32　　B．16　　C．1　　D．0

40．设 fp 已定义，执行语句 fp=fopen（"file","w"）;后，以下针对文本文件 file 操作的叙述中正确的是（　　）。

A．写操作结束后可以从头开始读　　B．只能写不能读

C．可以在原有内容后追加写　　D．可以随意读和写

二、填空题

函数 FUN 的功能是逆置数组元素中的值。形参 N 给出数组中的数据的个数。

例如，若 A 所指数组中的数据依次为 1、2、3、4、5、6、7、8、9，则逆置后依次为 9、8、7、6、5、4、3、2、1。

注意：给出了部分源程序，请勿改动主函数 main 和其他函数中的任何内容，仅在横线上填入所编写的若干表达式或语句。

试题程序如下。

```
#include   <stdio.h>
void fun ( int a[ ],  int n )
{
  int i, t;
  for  (i=0; i<____1____; i++)
  {
    t = a[i];
    a[i] = a[n-1-     2     ];
         3     = t;
  }
}
```

```
main( )
{
  int b[9] = {1, 2, 3, 4, 5, 6, 7, 8, 9}, i;
  printf ("\nThe original data :\n") ;
  for (i=0; i<9; i++)
    printf ("%4d ", b[i]) ;
  printf ("\n") ;
  fun (b, 9) ;
  printf ("\nThe data after invert :\n") ;
  for (i=0; i<9; i++)
    printf ("%4d ", b[i]) ;
  printf ("\n") ;
}
```

三、改错题

下列给定程序中，函数 fun 的功能如下：从 s 所指字符串中，找出 t 所指子串的个数作为函数值返回，例如，当 s 所指字符串中的内容为 abcdabfab, t 所指字符串的内容为 ab，则函数返回整数 3。

请改正程序中的错误，使它能得出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构。

试题程序如下。

```
#include <stdio.h>
#include <string.h>
int fun (char *s, char *t)
{
  int n;
  char *p, *r;
  n = 0;
  while (*s)
  {
    p = s;
    r = t;
    while (*r)
/********found********/
    if (r == p)
    {
          r++;
          p++;
    }
    else
    {
          break;
    }
/********found********/
    if (r == '\0')
      n++;
    s++;
  }
```

```
  return n;
}

void main( )
{
  char s[100], t[100];
  int m;
  printf ("\nPlease enter string s:") ;
  scanf ("%s", s) ;
  printf ("\nPlease enter substring t:") ;
  scanf ("%s", t) ;
  m = fun (s, t) ;
  printf ("\nThe result is: m=%d\n", m) ;
}
```

四、编程题

编写函数 fun，它的功能如下：将一个数字字符串转换为一个整数（不得调用 C 语言中提供的将字符串转换成整数的函数）。

例如，输入字符串“-1234“，函数把它转换为整数值-1234。

注意：给出了部分源程序，请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的大括号中填入所编写的若干语句。

试题程序如下。

```
long fun ( char *p)
{
     long s=0,t;
     int i=0,j,n=strlen(p),k,s1;
     if(p[0]=='-')
         i++;
     for(j=i;j<=n-1;j++)
     {
         t=p[j]-'0';
         s1=10;
         for(k=j;k<n-1;k++)
             t*=s1;
         s+=t;
     }
     if(p[0]=='-')
         return -s;
     else
         return s;
}
```

参 考 答 案

一、选择题

1～5	A	B	D	D	B	6～10	A	C	D	D	B
11～15	A	D	A	A	A	16～20	A	D	C	A	D
21～25	B	D	B	C	D	26～30	C	D	C	C	B
31～35	B	B	B	D	B	36～40	C	D	D	C	B

二、填空题

1～3n/2　　i　　a[n-i-1]

三、改错题

if （r==p）应改为 if（*r==*p），if（r==’\0’）应改为 if（*r==’\0’）。

四、编程题

```
long fun （ char *p）
{
  long s=0,t;
  int i=0,j,n=strlen（p）,k,s1;
  if（p[0]=='-'）
  i++;
  for（j=i;j<=n-1;j++）
  {
    t=p[j]-'0';
    s1=10;
    for（k=j;k<n-1;k++）
      t*=s1;
    s+=t;
  }
  if（p[0]=='-'）
    return -s;
  else
    return s;
}
```

附录 B　全国计算机等级考试
二级 C 语言上机考试

一、进入考试系统

双击桌面上的“二级 C 考试系统”图标，进入考试初始界面，如图 B-1 所示。

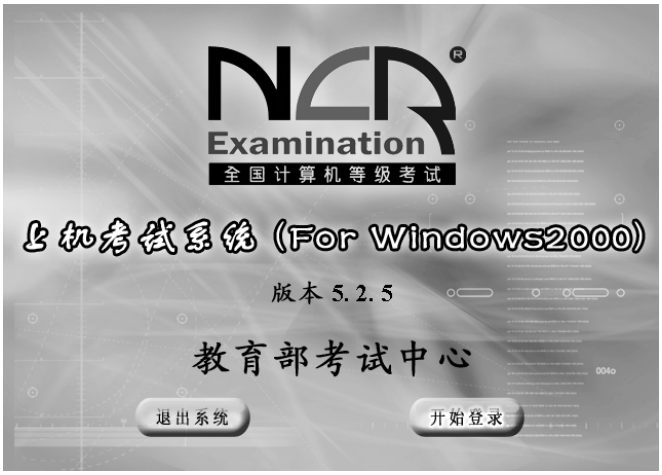


图 B-1　考试初始界面

二、输入考生信息

进入考试界面后，单击“开始登录”按钮即可进入考试环境进行“考生信息”的输入。信息输入界面如图 B-2 所示。

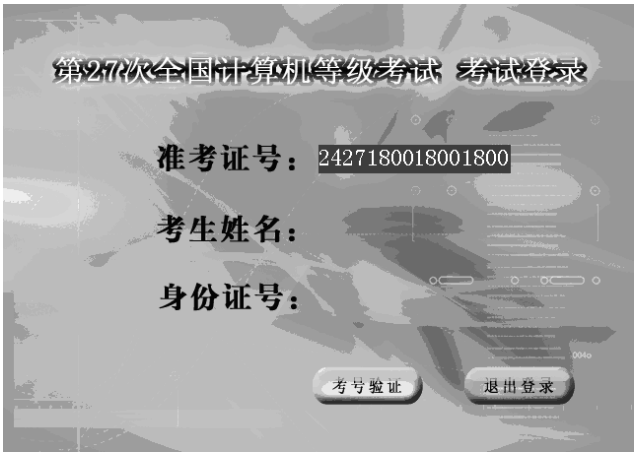


图 B-2　信息输入界面

考生输入准考证号（如“2427180018001800”）后，单击“考号验证”按钮进行信息核实，系统会弹出如图 B-3 所示的“登录提示”对话框。

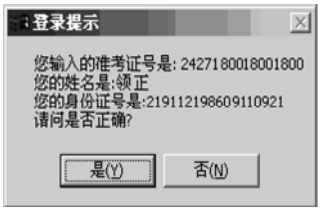


图 B-3　登录提示

在确定考生信息完全正确时，单击“是”按钮，进入答题界面，如图 B-4 所示，上方含有“程序填空题”、“程序修改题”、“程序设计题”三个按钮，自考试改为上机操作后，选择题只可进入一次，一旦退出，将不能再次进入。下面重点介绍程序填空题、程序修改题和程序设计题的操作注意事项。

三、开始答题

考生进入图 B-4 答题界面后，可以在程序填空题、程序修改题及程序设计题中任选一个开始答题，直到题目全部做完，做题过程中可反复保存，以最后一次为准，过程如下：

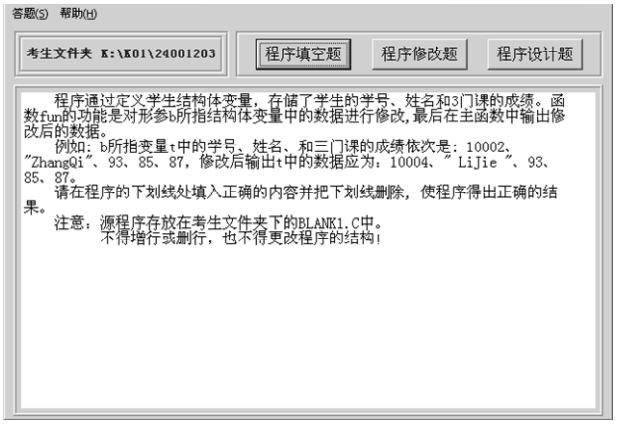


图 B-4　答题界面

1. 程序填空题

单击考试界面中的“程序填空题”按钮，题目显示区将显示题目对应的文字叙述信息，以了解该题目的内容。选择左上角“答题”菜单中的“启动 Microsoft Visual C++”菜单项，即可进入 Visual C++ 6.0 系统界面，如图 B-5 所示。

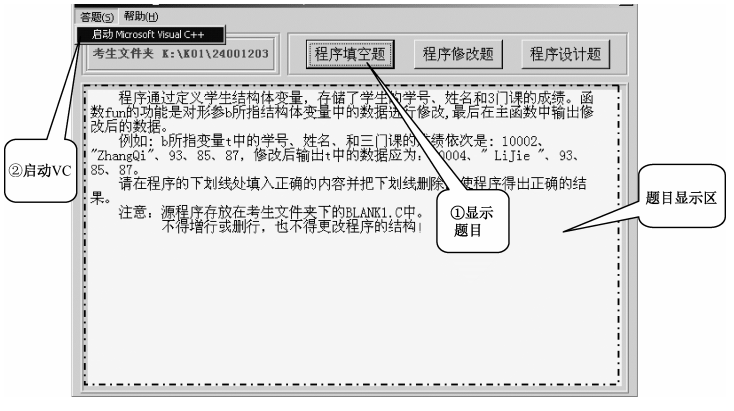


图 B-5　程序填空题

进入系统界面后，选择“文件”菜单中的“打开”选项，系统弹出“打开”对话框，选择“blank1.c”程序文件，单击左下角的“打开”按钮，如图 B-6 所示。

图 B-6　打开 blank1.c

打开“blank1.c”程序文件后，开始填空。填空方法如下：

1）在程序中找到“******** found *******”标识位置。

2）把“found”标识位置下面的需要填空的“占位符”删除（需要和横线一起删除），将程序的答案写在对应位置，如图 B-7 所示。

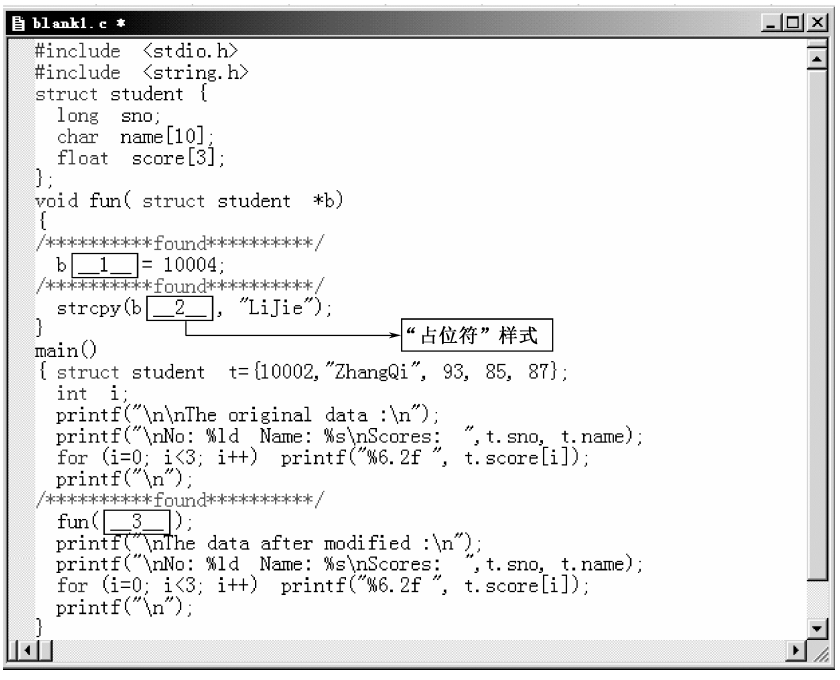


图 B-7　blank1.c

填完结果后的“blank1.c”程序如图 B-8 所示。

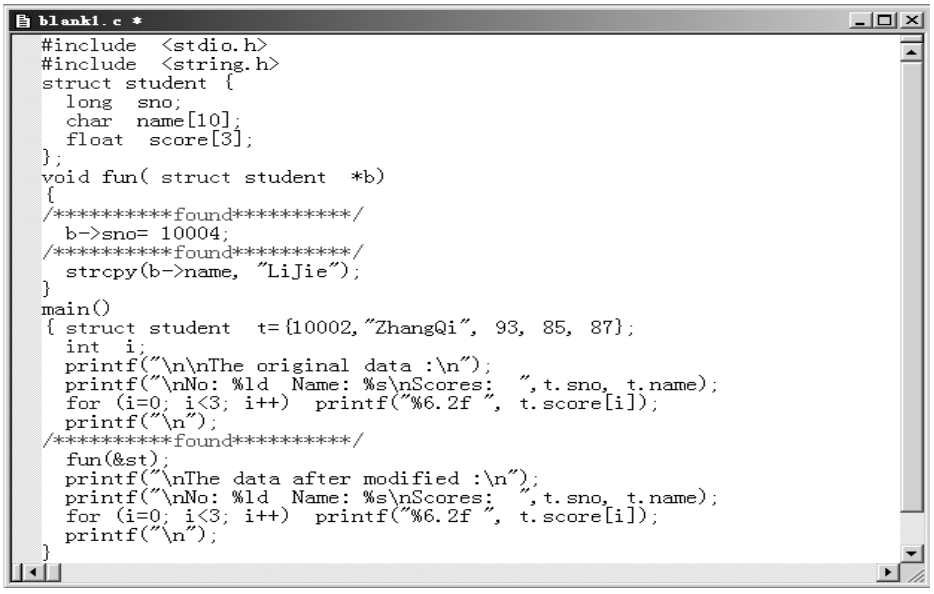


图 B-8　blank1.c 的结果

3）程序结果填写完成后，最好能“编译”、“组建”和“运行”，运行后，该文件自动进行保存。若感觉考试时间不够用，没有足够的时间运行程序，则一定要选择“文件”菜单中的“保存”选项，保存“blank1.c”程序文件，如图 B-9 所示。

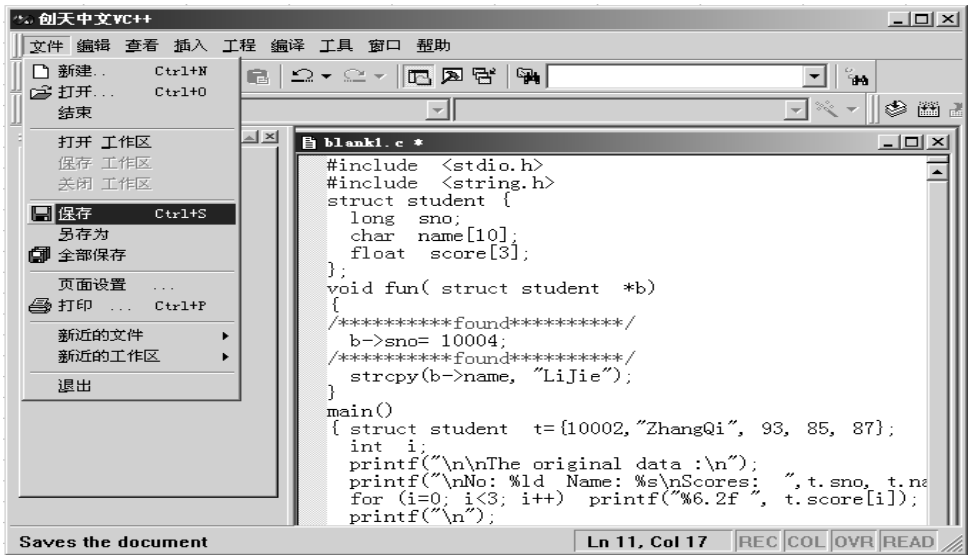


图 B-9　保存 blank1.c

4）“blank1.c”程序文件保存完成后，单击窗口右上角的“关闭”按钮，关闭“blank1.c”程序文件，并退出 Microsoft Visual C++ 6.0 系统界面，如图 B-10 所示，关闭“blank1.c”程序文件后，屏幕回到图 B-4 所示界面。

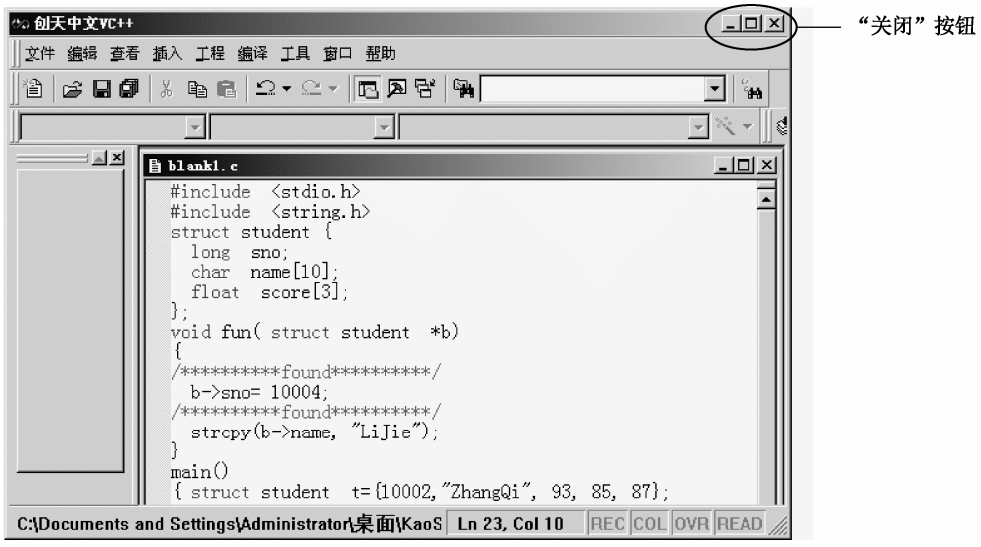


图 B-10　关闭 blank1.c

2. 程序修改题

单击考试界面中的“程序修改题”按钮后，题目显示区将显示题目对应的文字叙述信息。通过文字叙述可以了解到该题目的考试内容。选择“答题”菜单中的“启动 Microsoft Visual C++”选项，进入“Visual C++ 6.0”系统界面，如图 B-11 所示。

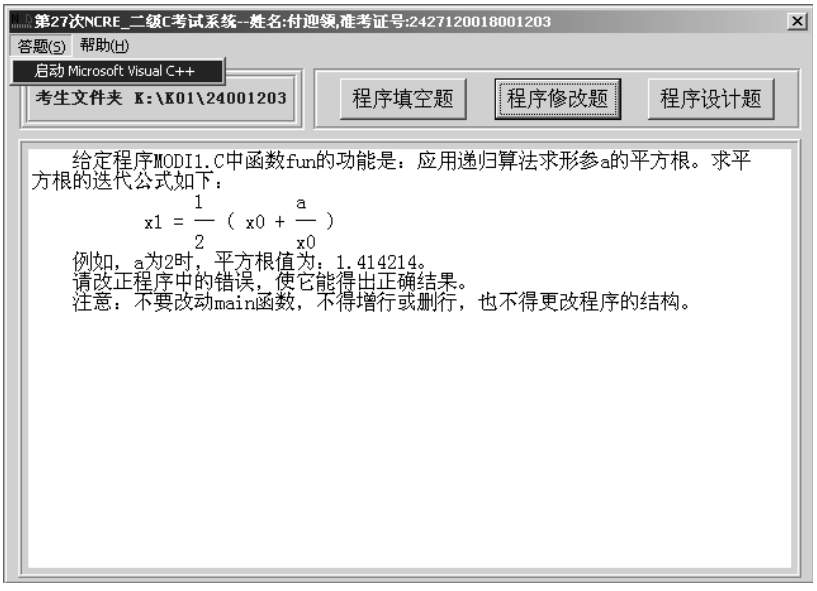


图 B-11　程序修改题

进入系统环境后，选择“文件”菜单中的“打开”选项，打开“modi1.c”程序文件，如图 B-12 所示。

图 B-12　打开程序修改题

打开“modi1.c”程序文件后，开始修改。修改方法如下。

1）在程序中找到“******** found *******”标识位置。

2）把“found”标识位置下面的错误程序修改成正确程序。图 B-13 所示为原“modi1.c”程序。

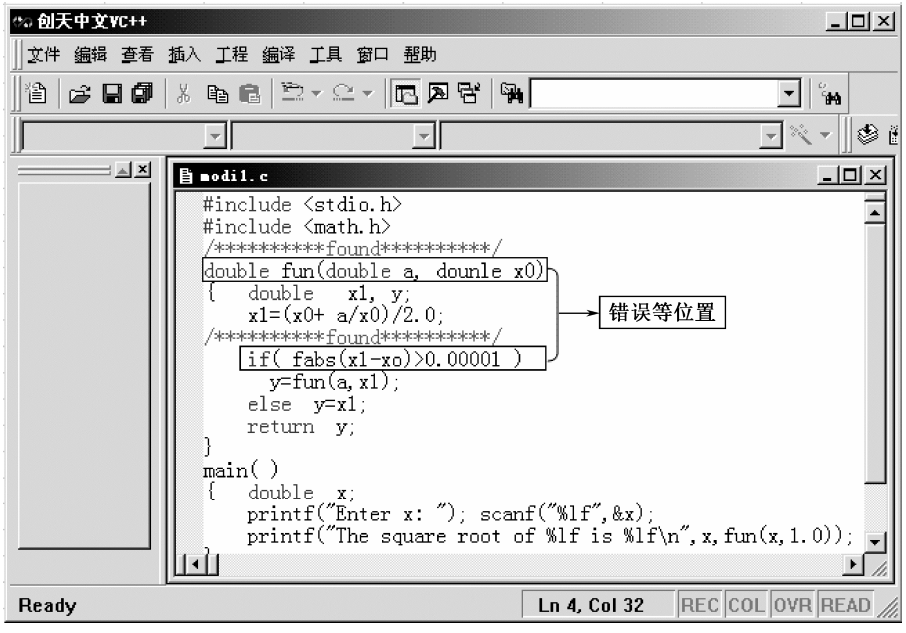


图 B-13　程序修改题原题

修改后的“modi1.c”程序如图 B-14 所示。

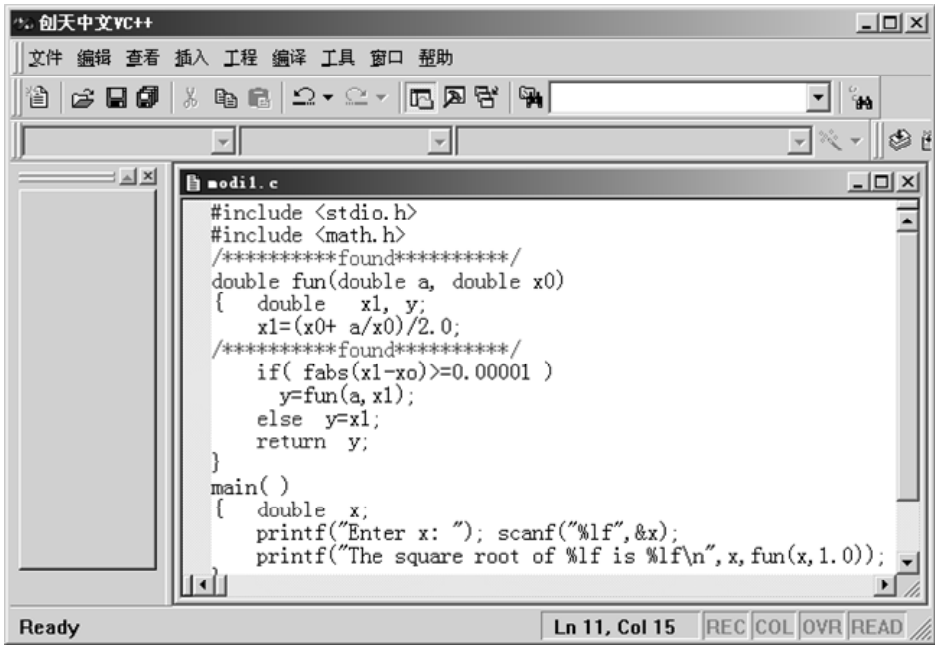


图 B-14　程序修改题结果

3）程序结果修改完成后，最好能“编译”、“组建”和“运行”，运行后，该文件自动进行保存。若感觉考试时间不够用，没有足够的时间运行程序，则一定要选择“文件”菜单中的“保存”选项，保存“modi1.c”程序文件，如图 B-15 所示。

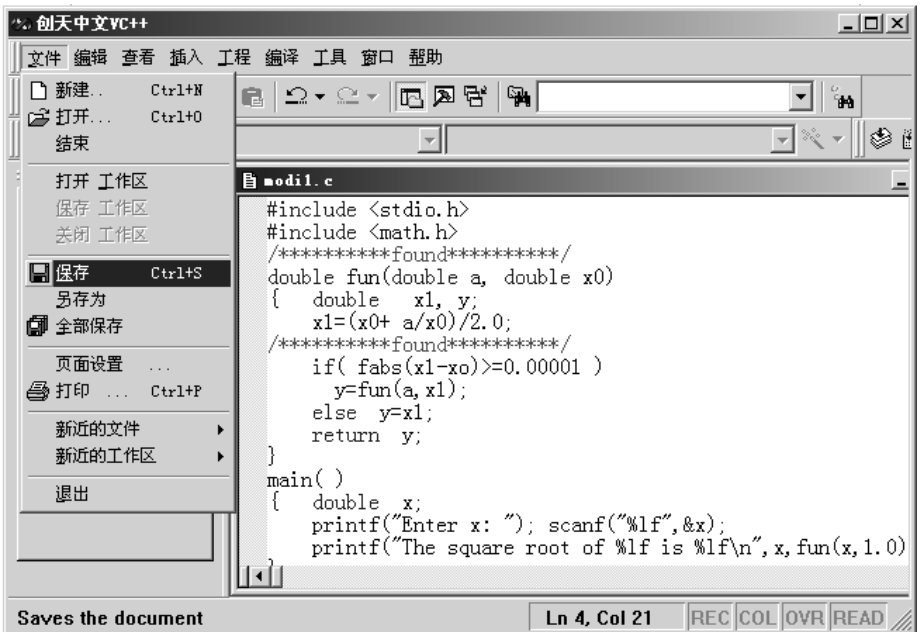


图 B-15　保存程序修改题

4）“modi1.c”程序文件保存完成后，单击窗口右上角的“关闭”按钮，关闭“modi1.c”

程序文件，并退出 Microsoft Visual C++ 6.0 系统，如图 B-16 所示。

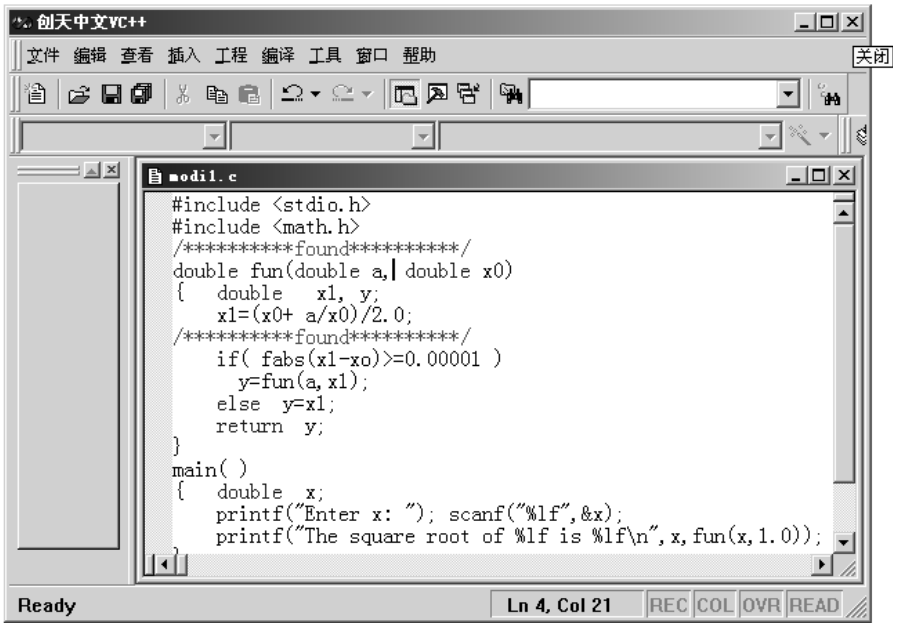


图 B-16　关闭程序修改题

3. 程序设计题

单击考试界面中的“程序设计题”按钮后，题目显示区将显示题目对应的文字叙述信息。通过文字叙述可以了解该题目的考试内容。选择“答题”菜单中的“启动 Microsoft Visual C++”进项，进入 Visual C++ 6.0 系统界面，如图 B-17 所示。

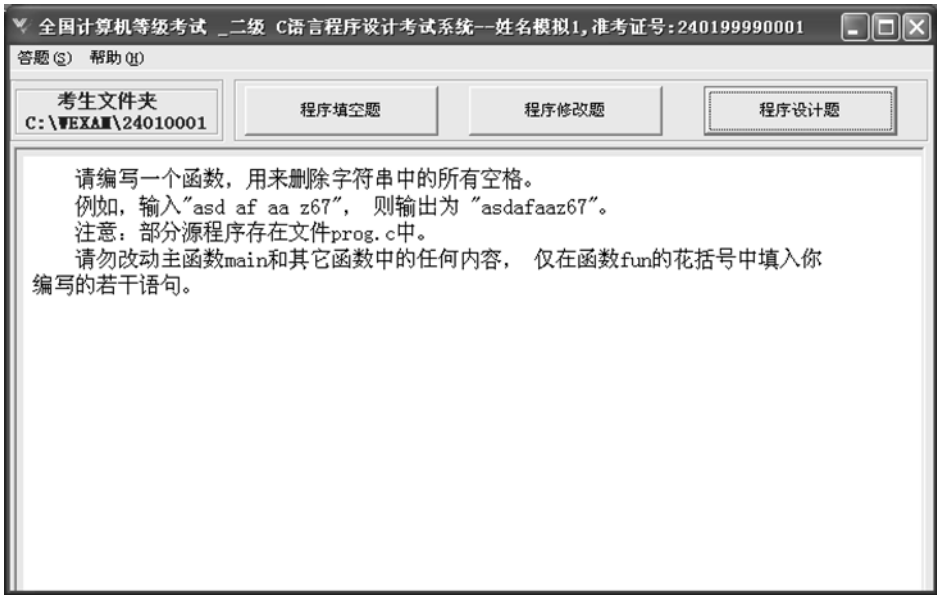


图 B-17　程序设计题

进入系统环境后，选择“文件”菜单中的“打开”选项，打开“prog.c”程序文件，如图 B-18 所示。

图 B-18　打开程序设计题

打开“prog.c”程序文件后，原“prog.c”程序如图 B-19 所示。

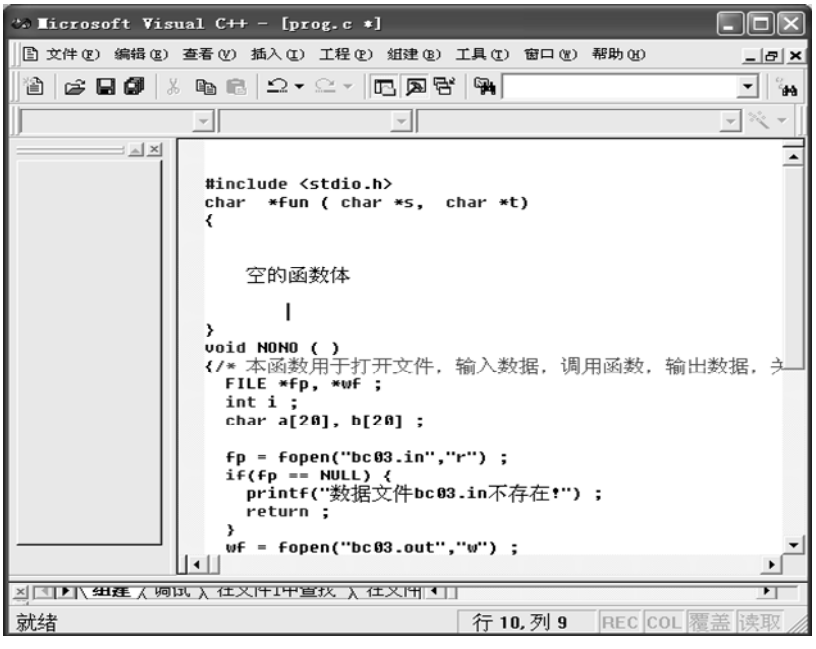


图 B-19　程序设计题原题

答题方法如下。

1）在程序中找到空的函数体。

2）将正确的程序代码写入空的函数体内。

修改后的“prog.c”程序如图 B-20 所示。

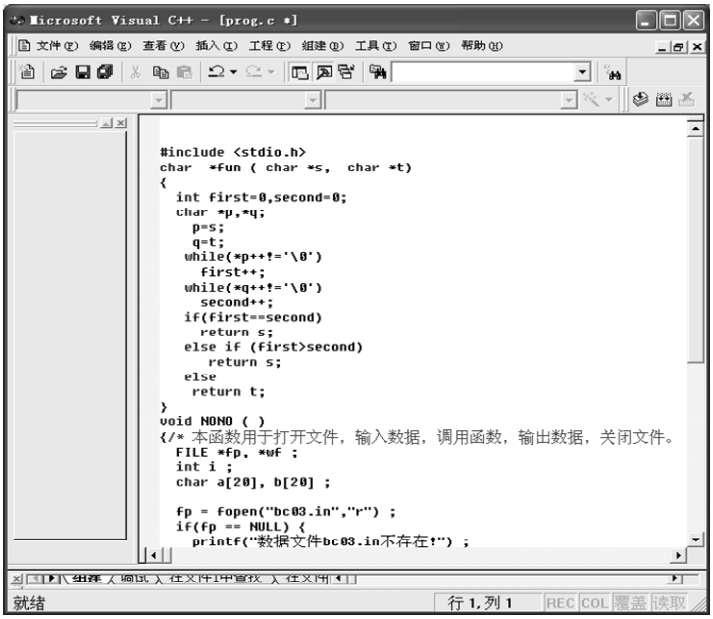


图 B-20　程序设计题结果

3）程序输入完成后，必须“编译”（图 B-21）、“组建”（图 B-22）和“运行”（图 B-23）“prog.c”程序文件，最后显示程序运行结果的输出窗口界面（图 B-24）。

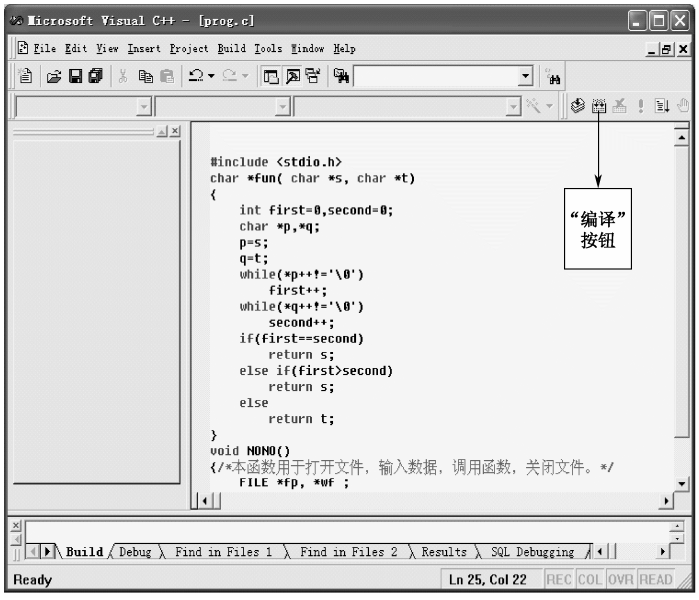


图 B-21　程序设计题（编译）

注意：单击“编译”按钮后，在弹出的对话框中一律都单击“是”按钮。

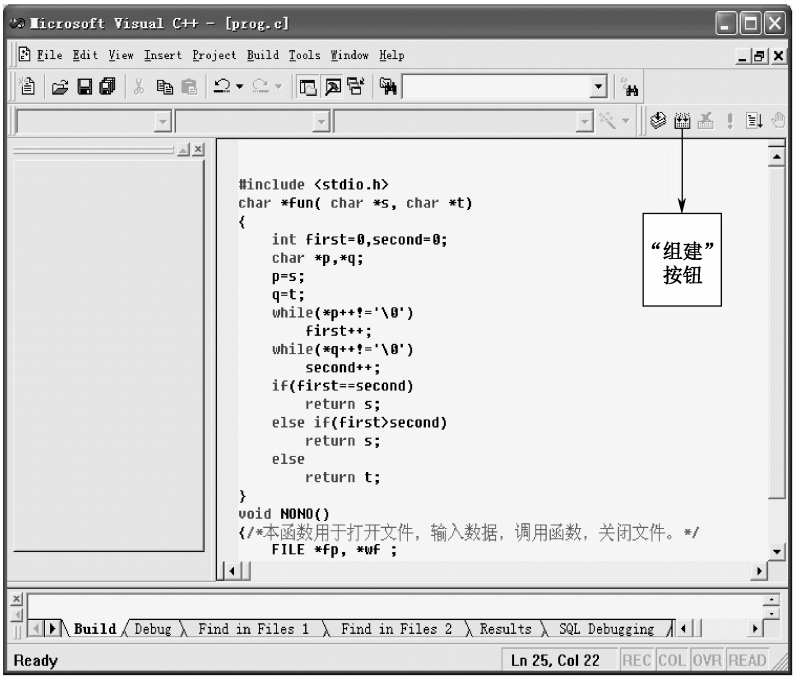


图 B-22　程序设计题（组建）

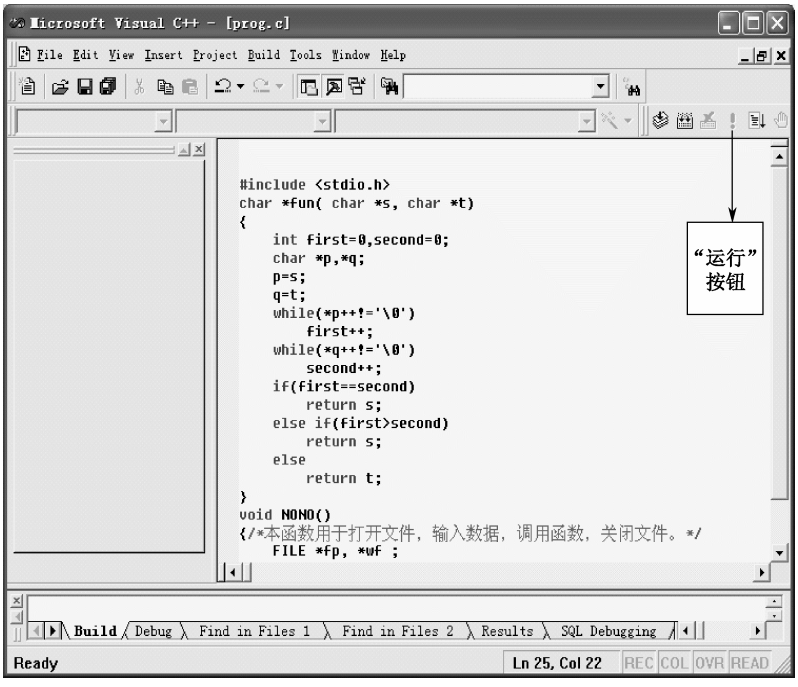


图 B-23　程序设计题（运行）

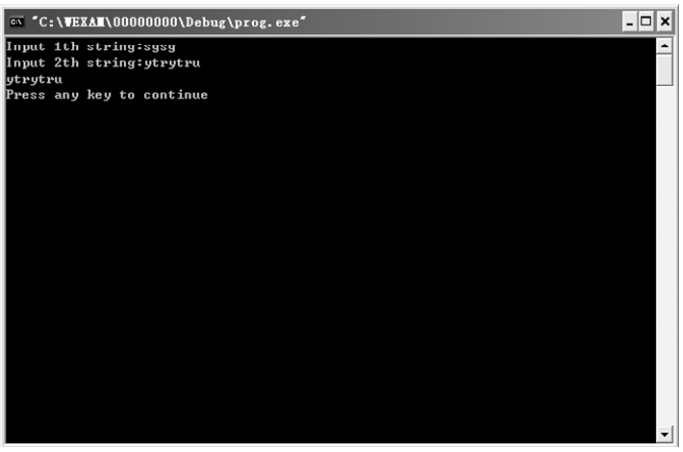


图 B-24　程序设计题运行结果

4）关闭所有窗口，单击“交卷”按钮，考试结束，如图 B-25 所示。

图 B-25　交卷

参 考 文 献

[1] 刘冰，赖涵，瞿中等．软件工程实践教程．北京：机械工业出版社，2009．
[2] 徐士良，葛兵．计算机软件技术基础．北京：清华大学出版社，2007．
[3] 张基温．C 语言程序设计案例教程．北京：清华大学出版社，2004．
[4] 廖湖声．C 语言程序设计案例教程．第 2 版．北京：人民邮电出版社，2010．
[5] 林冬梅，冉清．C 语言实训教程．北京：高等教育出版社，2011．
[6] 何中林．C 语言程序设计基础实训教程．天津：南开大学出版社，2012．