

# **《数据结构 A》实验指导书**

**石家庄铁道学院计算机系**  
**2006.8**

# 目 录

实验指导书概述 .....	3
实验大纲实习题 .....	5
实习报告规范 .....	9
实习步骤 .....	10
附录 1：实验报告示例.....	12
附录 2：实验教学大纲.....	15

## 实验指导书概述

“数据结构”是计算机专业一门重要的专业技术基础课程，是一门关键性核心课程。本课程系统地介绍了软件设计中常用的数据结构以及相应的存储结构和实现算法，介绍了多种常用的查找和排序技术，并对其进行了性能分析和比较，内容非常丰富。本课程的学习将为后续课程的学习以及软件设计水平的提高打下良好的基础。

由于以下原因，使得掌握这门课程具有较大难度：

- (1) 内容多，时间短，给学习带来困难；
- (2) 贯穿全书的动态链表存储结构和递归技术是学习中的重点和难点；
- (3) 隐含在各部分的技术和方法丰富，也是学习的重点和难点；
- (4) 先修课程中所介绍的专业性知识不多，加大了学习难度。

由于数据结构课程的技术性与实践性，《数据结构》课程实验的设置十分必要。为了帮助学生更好地学习本课程，理解和掌握算法设计所需的技术，为整个专业学习打好基础，要求运用所学知识，上机解决一些典型问题，通过分析、设计、编码、调试等各环节的训练，使学生深刻理解、牢固掌握所用到的一些技术。数据结构中稍微复杂一些的算法设计中可能同时要用到多种技术和方法，如算法设计的构思方法，动态链表，算法的编码，递归技术，与特定问题相关的技术等，要求重点掌握线性链表、二叉树和树、图结构、数组结构相关算法的设计。在掌握基本算法的基础上，掌握分析、解决实际问题的能力。通过实验实践内容的训练，突出构造性思维训练的特征，提高学生组织数据及编写大型程序的能力。

上机实习是对学生的一种全面综合训练，是与课堂听讲、自学和练习相辅相成的必不可少的一个教学环节。较大的实习题比平时的习题要复杂得多，也更接近实际。实习着眼于原理与应用的结合点，使学生学会如何把书上学到的知识用于解决实际问题，培养软件工作所需要的动手能力。实习还能使书上的知识变“活”，达到深化理解和灵活掌握教学内容的目的。平时的练习较偏重于如何编写功能单一的“小”算法，而实习题是软件设计的综合训练，包括问题分析，总体结构设计，用户界面设计，程序设计基本技能和技巧，多人合作，以至一整套软件工作规范的训练和科学作风的培养。此外，还有很重要的一点是：机器是比任何教师都严格的检查者。

每个实习题采取了统一的格式，由**问题描述**、**基本要求**、**测试数据**、**实现提示**和**选做内容**等5个部分组成。

**问题描述**旨在为读者建立问题提出的背景环境，指明问题“是什么”；

**基本要求**则对问题进一步求精，划出问题的边界，指出具体的参量或前提条件，并规定该题的最低限度要求；

**测试数据**部分旨在为检查学生上机作业提供方便，在完成实习题时应自己设计完整和严格的测试方案，当数据输入量较大时，提倡以文件形式向程序提供输入数据；

**实现提示**对实现中的难点及其解法思路等问题作了简要提示，个别问题给出了参考实现；

**选做内容**向那些尚有余力的读者提出了更严峻的挑战，同时也能开拓其他读者的思路，在完成基本要求时就力求避免就事论事的不良思想方法，尽可能寻求具有普遍意义的解法，使得程序结构合理，容易修改扩充。

在实现的时候应注意，要尽量减少依赖于具体机器计算环境的用法，若使用，也应在

注释中指出。这样得出的程序易于在不同机器上运行，有好的可移植性。C 语言是结构化程序设计语言，具有递归能力，可移植性也较好，是特别推荐的实现语言。

本书的一个特点是为实习制定了严格的规范。一种普遍存在的错误观念是，调试程序全凭运气。学生花 2 个小时的机上时间只找出一个错误，甚至一无所获的情况是常见的。其原因在于，很多人只认识到找错误，而没有认识到努力预先避免错误的重要性，也不知道应该如何努力。实际上，结构不好、思路和概念不清的程序可能是根本无法调试正确的。严格按照实习步骤规范进行实习，不但能有效地避免上述种种问题，更重要的是有利于培养软件工作者不可缺少的科学工作方法和作风。

在附录中提供了一个完整的实习报告示例，在起到实习报告规格范例作用的同时，还隐含地提供了很多有益的东西，比如基于数据类型的系统划分方法以及所提倡的程序设计风格等等。计算机学科在不断发展，可以使用的语言工具越来越丰富，在本书中的实习示例是应用面向过程的语言进行设计和编程，同样的实习题，也可以用面向对象的语言来实现。

## 实验大纲实习题

### 实习一 线性表应用

本次实习的主要目的在于熟悉线性表的基本运算在两种存储结构上的实现,其中以熟悉链表的各种操作为侧重点。通过本次实习还可帮助读者复习高级语言的使用方法。

#### 约瑟夫环

##### [问题描述]

约瑟夫 (Joseph) 问题的一种描述是: 编号为  $1, 2, \dots, n$  的  $n$  个人按顺时针方向围坐一圈, 每人持有一个密码 (正整数)。一开始任选一个正整数作为报数上限值  $m$ , 从第一个人开始按顺时针方向自 1 开始顺序报数, 报到  $m$  时停止报数。报  $m$  的人出列, 将他的密码作为新的  $m$  值, 从他在顺时针方向上的下一个人开始重新从 1 报数, 如此下去, 直至所有人全部出列为止。试设计一个程序求出出列顺序。

##### [基本要求]

利用单向循环链表存储结构模拟此过程, 按照出列的顺序印出各人的编号。

##### [测试数据]

$m$  的初值为 20; 密码: 3, 1, 7, 2, 4, 8, 4 (正确的结果应为 6, 1, 4, 7, 2, 3, 5)。

##### [实现提示]

程序运行后首先要求用户指定初始报数上限值, 然后读取各人的密码。设  $n \leq 30$ 。

##### [选作内容]

向上述程序中添加在顺序结构上实现的部分。

### 实习二 栈和队列应用

仅仅认识到栈和队列是两种特殊的线性表是远远不够的, 本次实习的目的在于使读者深入了解栈和队列的特征, 以便在实际问题背景下灵活运用它们, 同时还将巩固这两种结构的构造方法。

#### 停车场管理

##### [问题描述]

设停车场内只有一个的停放  $n$  辆汽车的狭长通道, 且只有一个大门可供汽车进出。汽车在停车场内按车辆到达时间的先后顺序, 依次由北向南排列 (大门在最南端, 最先到达的第一辆车停放在车场的最北端), 若车场内已停满  $n$  辆汽车, 则后来的汽车只能在门外的便道上等候, 一旦有车开走, 则排在便道上的第一辆车即可开入; 当停车场内某辆车要离开时, 在它之后开入的车辆必须先退出车场为它让路, 待该辆车开出大门外, 其它车辆再按原次序进入车场, 每辆停放在车场的车在它离开停车场时必须按它停留的时间长短交纳费用。试为停车场编制按上述要求进行管理的模拟程序。

##### [测试数据]

设  $n=2$ , 输入数据为: ('A', 1, 5), ('A', 2, 10), ('D', 1, 15), ('A', 3, 20), ('A', 4, 25), ('A', 5, 30), ('D', 2, 35), ('D', 4, 40), ('E', 0, 0)。其中, 'A' 表示到达; 'D' 表示离去, 'E' 表示输入结束。

##### [基本要求]

以栈模拟停车场，以队列模拟车场外的便道，按照从终端读入的输入数据序列进行模拟管理。每一组输入数据包括三个数据项：汽车“到达”或“离去”信息、汽车牌照号码及到达或离去的时刻，对每一组输入数据进行操作后的输出数据为：若是车辆到达，则输出汽车在停车场内或便道上的停车位置；若是车离去；则输出汽车在停车场内停留的时间和应交纳的费用（在便道上停留的时间不收费）。栈以顺序结构实现，队列以链表实现。

[实现提示]

需另设一个栈，临时停放为给要离去的汽车让路而从停车场退出来的汽车，也用顺序存储结构实现。输入数据按到达或离去的时刻有序。栈中每个元素表示一辆汽车，包含两个数据项：汽车的牌照号码和进入停车场的时刻。

[选作内容]

- (1) 两个栈共享空间，思考应开辟数组的空间是多少？
- (2) 汽车可有不同种类，则它们的占地面积不同，收费标准也不同，如 1 辆客车和 1.5 辆小汽车的占地面积相同，1 辆十轮卡车占地面积相当于 3 辆小汽车的占地面积。
- (3) 汽车可以直接从便道上开走，此时派在它前面的汽车要先开走让路，然后再依次排到队尾。
- (4) 停放在便道上的汽车也收费，收费标准比停放在停车场的车低，请思考如何修改结构以满足这种要求。

### 实习三 二叉树应用

#### 哈夫曼编/译码器

[问题描述]

利用哈夫曼编码进行通信可以大大提高信道利用率，这要求在发送端通过一个编码系统对待传输预先编码，在接收端将传来的数据进行译码。对于双工通道，每端都需要一个完整的编/译码系统。

[基本要求]

试为这样的信息收发站写一个哈夫曼码的编/译码系统。

[实现提示]

构造哈夫曼树的算法实现：

假设哈夫曼树采用双亲孩子表示法存储，并增加权值域，构造哈夫曼树的叶子结点（树木的权）有  $N$  个，合并次数为  $N-1$  次，则森林中总共有  $2N-1$  棵树，（包含合并后删除的）。

存储结构描述为：

```
const int    n=maxn      //maxn表示叶子数目
const int    m=2*n-1     //m为森林中树的棵数
class tree
{
    float weight;         //权值
    int  parent;          //双亲
    int  lch, rch;        //左，右孩子
}
tree  hftree[m+1];       //规定从第一个元素hftree[1]开始使用数组元素，故定义长度
```

度为  $m+1$  而不为  $m$

结构类型：

```

typedef struct
{
char data;
int weight;
int parent;
int lchild;
int rchild;
}huffnode;
typedef struct
{
char cd[MAX];
int start;
}huffcode;

主程序
int main()
{
    初始化：输入字符代码以及权值。
    编制哈夫曼码：根据权值建立二叉树，输出相应的根节点到叶结点的路径,便是哈夫曼编码。
    编码：输入字符,输出哈夫曼码。
    译码：输入哈夫曼,输出字符代码。
    退出：结束进程,退出程序。
    return 0;
}

```

#### 实习四 图的基本操作-校园导游

图是应用极为广泛的数据结构，也是这门课程的重点。它的特点在于非线性。稀疏矩阵的十字链表存储结构也是图的一种存储结构，故也把它们归在这次实习中。本章实习继续突出了数据结构加操作的程序设计观点，但根据这两种结构的非线性特点，将操作进一步集中在遍历操作上，因为遍历操作是其他众多操作的基础。遍历逻辑的（或符号形式的）结构，访问动作可是任何操作。本次实习还希望达到熟悉各种存储结构的特征，以及如何应用树和图结构解决具体问题（即原理与应用的结合）等目的。

##### 图的基本操作

###### [问题描述]

分别用邻接矩阵和邻接表实现以下操作：图的创建、遍历、插入、删除、最短路径。参考题目为校园导游程序

###### [基本要求]

熟悉图的常用存储结构和基本操作。

###### [实现提示]

设图的结点不超过 30 个，每个结点用一个编号表示（如果一个图有  $n$  个结点，则它们的编号分别为  $1, 2, \dots, n$ ）。通过输入图的全部边输入一个图，每个边为一个数对，可以对边的输入顺序作出某种限制。注意，生成树的边是有向边，端点顺序不能颠倒。

###### [选作内容]

（1）借助于栈类型（自己定义和实现）将深度优先遍历用非递归算法实现。

（2）以邻接表为存储结构建立深度优先生成树和广度优先生成树，再按凹入表或树形打印生成树。

## 实习五 哈希表设计

本次实习旨在集中对几个专门的问题作较为深入的探讨和理解,也强调对某些特定的编程技术的训练。

### 哈希表设计

#### [问题描述]

针对某个集体中人名设计一个哈希表,使得平均查找长度不超过  $R$ ,并完成相应的建表和查表程序。

#### [基本要求]

假设人名为中国人姓名的汉语拼音形式。待填入哈希表的人名共有 30 个,取平均查找长度的上限为 2。哈希函数用除留余数法构造,用线性探测再散列法或链地址法处理冲突。

#### [测试数据]

取读者周围较熟悉的 30 个人名。

#### [选作内容]

(1) 从教科书上介绍的集中哈希函数构造方法中选出适用者并设计几个不同的哈希函数,比较他们的地址冲突率(可以用更大的名字集合作实验)。

(2) 研究这 30 个人名的特点,努力找一个哈希函数,使得对于不同的拼音名一定不发生地址冲突。

(3) 在哈希函数确定的前提下尝试各种不同处理冲突的方法,考察平均查找长度的变化和造好的哈希表中关键字的聚集性。

## 实习六 常用排序算法的对比分析

本次实习旨在集中对几个专门的问题作较为深入的探讨和理解,也不强调对某些特定的编程技术的训练。

### 常用排序算法的对比分析

#### [问题描述]

各种内部排序算法的时间复杂度分析结果只给出了算法执行时间的阶,或大概执行时间。试通过随机的数据比较各算法的关键字比较次数和关键字移动次数,以取得直观感受。

#### [基本要求]

对以下常用的内部排序算法进行比较:起泡排序、直接插入排序、简单选择排序、快速排序、希尔排序、堆排序。

#### [测试数据]

由随机产生器决定。

#### [实现提示]

待排序表的表长不少于 100;其中的数据要用伪随机数产生程序产生;至少要用 5 组不同的输入数据作比较;比较的指标为有关关键字参加的比较次数和关键字移动次数(关键字交换计为 3 次移动)。

#### [选作内容]

对不同的输入表长做试验,观察检查两个指标相关于表长的变化关系。还可以对稳定性做验证。



## 实习报告规范

实习报告的开头应给出题目、班级、姓名、学号和完成日期，并包括以下7个内容：

### 1. 需求分析

以无歧义的陈述说明程序设计的任务，强调的是程序要做什么？并明确规定：

- (1)输入的形式和输入值的范围；
- (2)输出的形式；
- (3)程序所能达到的功能；
- (4)测试数据：包括正确的输入及其输出结果和含有错误的输入及其输出结果。

### 2. 概要设计

说明本程序中用到的所有抽象数据类型的定义、主程序的流程以及各程序模块之间的层次(调用)关系。

### 3. 详细设计

实现概要设计中定义的所有数据类型，对每个操作只需要写出伪码算法；对主程序和其他模块也都需要写出伪码算法(伪码算法达到的详细程度建议为：按照伪码算法可以在计算机键盘直接输入高级程序设计语言程序)；画出函数和过程的调用关系图。

### 4. 调试分析

内容包括：

- a. 调试过程中遇到的问题是如何解决的以及对设计与实现的回顾讨论和分析；
- b. 算法的时空分析(包括基本操作和其他算法的时间复杂度和空间复杂度的分析)和改进设想；
- c. 经验和体会等。

### 5. 用户使用说明

说明如何使用你编写的程序，详细列出每一步的操作步骤。

### 6. 测试结果

列出你的测试结果，包括输入和输出。这里的测试数据应该完整和严格，最好多于需求分析中所列。

### 7. 附录

带注释的源程序。如果提交源程序软盘，可以只列出程序文件名的清单。

在以下各实习单元中都提供了实习报告实例。值得注意的是，实习报告的各种文档资料，如：上述中的前三部分要在程序开发的过程中逐渐充实形成，而不是最后补写(当然可以也应该最后用实验报告纸誊清或打印)。

## 实习步骤

随着计算机性能的提高，它所面临的软件开发的复杂度也日趋增加。然而，编制一个 10000 行的程序的难度绝不仅仅是一个 5000 行的程序的两倍，因此软件开发需要系统的方法。一种常用的软件开发方法，是将软件开发过程分为分析、设计、实现和维护四个阶段。虽然数据结构课程中的实习题的复杂度远不如(从实际问题中提出来的)一个“真正的”软件，但为了培养一个软件工作者所应具备的科学工作的方法和作风，我们制订了如下所述完成实习的 5 个步骤：

### 1. 问题分析和任务定义

通常，实习题目的陈述比较简洁，或者说是模棱两可的含义。因此，在进行设计之前，首先应该充分地分析和理解问题，明确问题要求做什么，限制条件是什么。注意：本步骤强调的是做什么，而不是怎么做。对问题的描述应避免算法和所涉及的数据类型，而是对所需完成的任务作出明确的回答。例如：输入数据的类型、值的范围以及输入的形式；输出数据的类型、值的范围及输出的形式；若是会话式的输入，则结束标志是什么，是否接受非法的输入，对非法输入的回答方式是什么等等。这一步还应该为调试程序准备好测试数据，包括合法的输入数据和非法形式输入的数据。

### 2. 数据类型和系统设计

在设计这一步骤中需分逻辑设计和详细设计两步实现。逻辑设计指的是，对问题描述中涉及的操作对象定义相应的数据类型，并按照以数据结构为中心的原则划分模块，定义主程序模块和各抽象数据类型；详细设计则为定义相应的存储结构并写出各过程和函数的伪码算法。在这个过程中，要综合考虑系统功能，使得系统结构清晰、合理、简单和易于调试，抽象数据类型的实现尽可能做到数据封装，基本操作的规格说明尽可能明确具体。作为逻辑设计的结果，应写出每个抽象数据类型的定义(包括数据结构的描述和每个基本操作的规格说明)，各个主要模块的算法，并画出模块之间的调用关系图。详细设计的结果是对数据结构和基本操作的规格说明作出进一步的求精，写出数据存储结构的类型定义，按照算法书写规范用类 Pascal 语言写出过程或函数形式的算法框架。在求精的过程中，应尽量避免陷入语言细节，不必过早表述辅助数据结构和局部变量。

### 3. 编码实现和静态检查

编码是把详细设计的结果进一步求精为程序设计语言程序。程序的每行不要超过 60 个字符。每个过程(函数)体，即不计首部和规格说明部分，一般不要超过 40 行。最长不得超过 60 行，否则应该分割成较小的过程(函数)。要控制 IF 语句连续嵌套的深度。其他要求参见第一篇的算法书写规范。如何编写程序才能较快地完成调试是特别要注意的问题。对于编程很熟练的读者，如果基于详细设计的伪码算法就能直接在键盘上输入程序的话，则可以不必要用笔在纸上写出编码，而将这一步的工作放在上机准备之后进行，即在上机调试之前直接用键盘输入。

然而，不管你是否写出编码的程序，在上机之前，认真的静态检查却是必不可少的。多数初学者在编好程序后处于以下两种状态之一：一种是对自己的“精心作品”的正确性确信

不疑；另一种是认为上机前的任务已经完成，纠查错误是上机的工作。这两种态度是极为有害的。事实上，非训练有素的程序设计者编写的程序长度超过 50 行时，极少不含有除语法错误以外的错误。上机动态调试决不能代替静态检查，否则调试效率将是极低的。静态检查主要有两种方法，一是用一组测试数据手工执行程序(通常应先分模块检查)；二是通过阅读或给别人讲解自己的程序而深入全面地理解程序逻辑，在这个过程中再加入一些注解和断言。如果程序中逻辑概念清楚，后者将比前者有效。

#### **4. 上机准备和上机调试**

上机准备包括以下几个方面：

- (1) 高级语言文本(体现与编译程序用户手册)的扩充和限制。
- (2) 如果用 c 语言，要特别注意平时惯用的类 c 语言与标准 c 语言之间的细微差别。
- (3) 熟悉机器的操作系统和语言集成环境的用户手册，尤其是最常用的命令操作，以便顺利进行上机的基本活动。
- (4) 掌握调试工具，考虑调试方案，设计测试数据并手工得出正确结果。“磨刀不误砍柴工”。计算机各专业的学生应该能够熟练运用高级语言的程序调试器 **DEBUG** 调试程序。

上机调试程序时要带一本高级语言教材或手册。调试最好分模块进行，自底向上，即先调试底层过程或函数。必要时可以另写一个调用驱动程序。这种表面上麻烦的工作实际上可以降低调试所面临的复杂性，提高调试工作效率。

在调试过程中可以不断借助 **DEBUG** 的各种功能，提高调试效率。调试中遇到的各种异常现象往往是预料不到的，此时不应“苦思冥想”，而应动手确定疑点，通过修改程序来证实它或绕过它。调试正确后，认真整理源程序及其注释，印出带有完整注释的且格式良好的源程序清单和结果。

#### **5. 总结和整理实习报告**

## 附录 1: 实验报告示例

\_\_\_\_\_级 \_\_\_\_\_班 \_\_\_\_\_年\_\_\_\_\_月\_\_\_\_\_日  
姓名\_\_\_\_\_ 学号\_\_\_\_\_ 电话\_\_\_\_\_

### 1. 实验题目

编制一个演示单链表插入、删除、查找等操作的程序

### 2. 需求分析

本演示程序用 TC 编写, 完成单链表的生成, 任意位置的插入、删除, 以及确定某一元素在单链表中的位置。

① 输入的形式和输入值的范围: 插入元素时需要输入插入的位置和元素的值; 删除元素时输入删除元素的位置; 查找操作时需要输入元素的值。在所有输入中, 元素的值都是整数

② 输出的形式: 在所有三种操作中都显示操作是否正确以及操作后单链表的内容。其中删除操作后显示删除的元素的值, 查找操作后显示要查找元素的位置。

③ 程序所能达到的功能: 完成单链表的生成(通过插入操作)、插入、删除、查找操作

④ 测试数据:

A. 插入操作中依次输入 11, 12, 13, 14, 15, 16, 生成一个单链表

B. 查找操作中依次输入 12, 15, 22 返回这 3 个元素在单链表中的位置

C. 删除操作中依次输入 2, 5, 删除位于 2 和 5 的元素

### 3. 概要设计

1) 为了实现上述程序功能, 需要定义单链表的抽象数据类型:

ADT LinkList {

数据对象:  $D=\{a_i|a_i \in \text{IntegerSet}, i=0,1,2,\dots,n, n \geq 0\}$

数据关系:  $R=\{<a_i, a_{i+1}>|a_i, a_{i+1} \in D\}$

基本操作:

InitLinkList(&L)

操作结果: 构造一个空的单链表 L.

InsLinkList(&L, pos, e)

初始条件: 单链表 L 已存在

操作结果: 将元素 e 插入到单链表 L 的 pos 位置

DelLinkList(&L, pos, &e)

初始条件: 单链表 L 已存在

操作结果: 将单链表 L 中 pos 位置的元素删除, 元素值置入 e 中返回

LocLinkList(L, e)

初始条件: 单链表 L 依存在

操作结果: 单链表 L 中查找是否元素 e,

若存在, 返回元素在表中的位置; 若不存在, 返回-1.

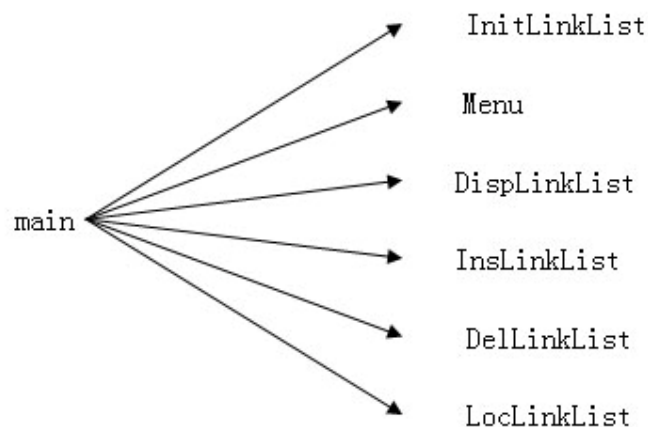
Menu()

操作结果: 在屏幕上显示操作菜单

2) 本程序包含 7 个函数:

- ① 主函数 main()
- ② 初始化单链表函数 InitLinkList()
- ③ 显示操作菜单函数 menu()
- ④ 显示单链表内容函数 dispLinkList()
- ⑤ 插入元素函数 InsLinkList()
- ⑥ 删除元素函数 DelLinkList()
- ⑦ 查找元素函数 LocLinkList()

各函数间关系如下:



#### 4. 详细设计

实现概要设计中定义的所有的数据类型, 对每个操作给出伪码算法。对主程序和其他模块也都需要写出伪码算法。

1) 结点类型和指针类型

```
typedef struct node {  
    int data;  
    struct node *next;  
}Node,*LinkList;
```

2) 单链表的基本操作

为了方便, 在单链表中设头结点, 其 data 域没有意义。

```
bool InitLinkList(LinkList &L)
```

(伪码算法)

```
void DispLinkList(LinkList L)
```

(伪码算法)

```
void menu()
```

(伪码算法)

```
bool InsLinkList(LinkList &L,int pos,int e)
```

(伪码算法)

```
bool DelLinkList(LinkList &L,int pos,int &e)
```

(伪码算法)

```
int LocLinkList(LinkList L,int e)
```

(伪码算法)

3) 其他模块伪码算法

## 5. 调试分析

(略)

## 6. 使用说明

程序名为 LinkList.exe，运行环境为 DOS。程序执行后显示

=====

0---EXIT

1---INSERT

2---DELETE

3---LOCATE

=====

SELECT:

在 select 后输入数字选择执行不同的功能。要求首先输入足够多的插入元素，才可以进行其他的操作。每执行一次功能，就会显示执行的结果（正确或错误）以及执行后单链表的内容。

选择 0: 退出程序

选择 1: 显示 “INSERT pos,e = ” ，

要求输入要插入的位置和元素的值（都是整数）。

选择 2: 显示 “DELETE pos = ” ，

要求输入要删除元素的位置，执行成功后返回元素的值。

选择 3: 显示 “LOCATE e = ” ，

要求输入要查找元素的值，执行成功后返回元素在表中的位置

## 7. 测试结果

1) 建立单链表:

» 选择 1，分别输入 (0, 11), (0, 12), (0, 13), (0, 14) (0, 15)。得到单链表 (15, 14, 13, 12, 11)

2) 插入:

» 选择 1 输入 (1, 100)，得到单链表 (15, 100, 14, 13, 12, 11)

» 选择 1 输入 (-1, 2)，显示输入错误

» 选择 1 输入 (7, 2)，显示输入错误

» 选择 1 输入 (6, 2)，得到单链表 (15, 100, 14, 13, 12, 11, 2)

3) 删除:

» 选择 2，输入 1。返回 e=100，得到单链表 (15, 14, 13, 12, 11, 2)

» 选择 2，输入 0。返回 e=15，得到单链表 (14, 13, 12, 11, 2)

» 选择 2，输入 4。返回 e=2，得到单链表 (14, 13, 12, 11)

» 选择 2，输入 5。返回输入错误

4) 查找

» 选择 3，输入 14。返回 pos=0

» 选择 3，输入 100。返回输入错误

## 附录 2：实验教学大纲

课程编号：0901170

计划学时：12

面向专业：计算机科学与技术专业、电子商务

制 订：计算机软件教研室

执 笔 人：武守秋

审 定 人：邸书灵

### 一、课程性质、目的及任务

《数据结构》是计算机科学与技术专业的一门学位课，它是《数据库原理》、《操作系统》、《编译原理》等的先行课程。通过本课程上机练习，使学生在数据的逻辑特征和物理特征、数据结构的选择和应用，以及算法设计实现等方面加深对数据结构基本内容的理解和灵活应用。同时，在程序设计方法以及上机操作基本技能和科学作风方面受到比较系统严格的训练，培养动手能力。

### 二、主要参考书

- 1、《数据结构（C 语言版）》 严蔚敏，吴伟民 清华大学出版社 2002 年
- 2、《数据结构题集（C 语言版）》 严蔚敏等 清华大学出版社 2002 年
- 3、《数据结构—C 语言描述》 耿国华等 西安电子科技大学出版社 2002 年
- 4、Data Structure & Program Design in C Robert L. Kruse 清华大学出版社 2001 年
- 5、《数据结构（用面向对象方法与 C++描述）》 殷人昆 清华大学出版社 1999 年

### 三、考试考核办法

实验成绩按学生的实验态度、独立动手能力和实验报告综合评定。

### 四、实验项目与内容提要

序号	实验项目名称	实验内容提要	实验性质	实验者类别	学时数	开设组数	每组人数	实验消耗（元/人时）	主要仪器设备名称及配套数
1	线性表应用	约瑟夫环问题。用循环链表实现约瑟夫环问题	验证演示	研究生					微机、windows/

		题，熟悉链表结构的使用。	设计综合	本科生	2	70	1	1	unix/linux 、 TC/VB/ VC++
				专科生					
2	栈和队列应用	停车场管理。利用栈和队列模拟停车场管理，学习利用栈和队列解决实际问题。	验证演示设计综合	研究生					微机、 windows/ unix/linux 、 TC/VB/ VC++
				本科生	2	70	1	1	
				专科生					
3	二叉树应用	哈夫曼编/译码器。掌握哈夫曼树的构造及哈夫曼编/译码方法。	验证演示设计综合	研究生					微机、 windows/ unix/linux 、 TC/VB/ VC++
				本科生	2	70	1	1	
				专科生					
4	图的基本操作-校园导游	分别用邻接矩阵和邻接表实现以下操作：图的创建、遍历、插入、删除、最短路径。熟悉图的常用存储结构和基本操作。	验证演示设计综合	研究生					微机、 windows/ unix/linux 、 TC/VB/ VC++
				本科生	2	70	1	1	
				专科生					
5	哈希表设计	用除留余数法构造哈希函数，用线性探测再散列法处理冲突。	验证演示设计综合	研究生					微机、 windows/ unix/linux 、 TC/VB/ VC++
				本科生	2	70	1	1	
				专科生					
6	常用排序算法的对比分析	分别实现起泡排序、直接插入排序、简单选择排序、快速排序、希尔排序、堆排序。掌握常用排序算法的特点，以便根据实际情况选择使用。	验证演示设计综合	研究生					微机、 windows/ unix/linux 、 TC/VB/ VC++
				本科生	2	70	1	1	
				专科生					