



应用型本科信息大类专业“十三五”规划教材



从应用到实现

# 数据结构实验指导 与习题解析 (C语言版)

孙丽云 马睿 主编



华中科技大学出版社  
<http://www.hustp.com>



应用型本科信息大类专业“十三五”规划教材

# 数据结构实验指导 与习题解析 (C语言版)

主 编 孙丽云 马 睿

副主编 李 珊 宋丽辉 徐 梅



华中科技大学出版社

<http://www.hustp.com>

中国·武汉

## 内 容 提 要

本书是与《数据结构(C语言版)》配套的辅助教材。本书分为两大部分:第一部分为与《数据结构(C语言版)》中每章对应的实验指导,其中每章的实验分为“基本实验练习”“自己动手做练习”和“综合应用实例”三个部分,力求综合应用课本中的知识,解决实际生活中的例子;第二部分对《数据结构(C语言版)》书中的课后习题进行了详细解析。

为了方便教学,本书还配有电子课件等教学资源包,任课教师和学生可以登录“我们爱读书”网([www.ibook4us.com](http://www.ibook4us.com))免费注册并浏览,或者发邮件至 [hustpeiit@163.com](mailto:hustpeiit@163.com) 免费索取。

本书可作为高等院校计算机类、电子信息类、自动化类、电气类、光电类及其他相关专业学生的教材和教学参考书,也可作为工程技术人员的参考资料和感兴趣的读者的自学读物。

### 图书在版编目(CIP)数据

数据结构实验指导与习题解析:C语言版/孙丽云,马睿主编. —武汉:华中科技大学出版社,2017.2  
应用型本科信息大类专业“十三五”规划教材  
ISBN 978-7-5680-2607-9

I. ①数… II. ①孙… ②马… III. ①数据结构-高等学校-教学参考资料 ②C语言-程序设计-高等学校-教学参考资料 IV. ①TP311.12 ②TP312.8

中国版本图书馆 CIP 数据核字(2017)第 034168 号

### 数据结构实验指导与习题解析(C语言版)

Shuju Jiegou Shiyan Zhidao yu Xiti Jiexi(C Yuyan Ban)

孙丽云 马睿 主编

策划编辑:康序

责任编辑:康序

封面设计:原色设计

责任监印:朱玢

出版发行:华中科技大学出版社(中国·武汉)

电话:(027)81321913

武汉市东湖新技术开发区华工科技园

邮编:430223

录排:武汉正风天下文化发展有限责任公司

印刷:仙桃市新华印务有限责任公司

开本:787mm×1092mm 1/16

印张:8

字数:214千字

版次:2017年2月第1版第1次印刷

定价:25.00元



本书若有印装质量问题,请向出版社营销中心调换  
全国免费服务热线:400-6679-118 竭诚为您服务  
版权所有 侵权必究

# 前言

## PREFACE

“数据结构”课程是高等学校计算机及相关专业的一门重要的专业基础课程,但很多学生存在“理论强,实践弱”的现象,即课堂上所讲算法都明白,但自己无法实现这些算法。

本书的实验指导部分,在“基本实验练习”环节给出了一个问题的完整实现算法,学生可以先通过读代码,来理解算法;在“自己动手做练习”环节给出了一个相似的问题,学生可以借助解决相似问题,来进一步理解算法,并且在本部分只给出了程序运行的界面图,即给出了具体要求,但没有给出代码,学生可模仿“基本实验练习”中的代码来实现,进行自我练习;在“综合应用实例”环节给出了一个综合应用的例子,并给出了具体的实现代码,学生可以将本部分知识进行综合应用。

本书中所有算法都采用 C 语言函数的形式描述,这些函数中都在关键语句进行了详细注释,并已在 Visual C++6.0 运行环境下调试运行,便于读者理解算法,并方便读者对基本运算进行验证,从而在此基础上学会应用。

本书主编为燕京理工学院孙丽云和马睿,副主编为燕京理工学院李珊和宋丽辉、武汉工程科技学院徐梅。编写人员分工为:马睿编写了实验 1、实验 8、实验 9,第 1 章、第 6 章和第 7 章的课后习题解析;宋丽辉编写了实验 2 并完成了统稿工作;孙丽云编写了实验 3、实验 4、实验 5、实验 7、实验 10、综合实验 1、综合实验 2,第 2 章、第 3 章的课后习题解析,并进行了统稿;李珊编写了实验 6 和第 4 章的课后习题解析;邵兰洁编写了第 5 章的课后习题解析;徐梅编写了第 8 章的课后习题解析和自测练习题。课题组成员刘淑艳、刘佩贤、王慧、牛玉玲等提供了大量的编写素材。

本书在编写过程中得到了燕京理工学院信息科学与技术学院各位领导的指导和帮助,同时得到了华中科技大学出版社的大力支持,在此一并表示感谢。

为了方便教学,本书还配有电子课件等教学资源包,任课教师和学生可以登录“我们爱读书”网([www.ibook4us.com](http://www.ibook4us.com))免费注册并浏览,或者发邮件至 [hust-peiit@163.com](mailto:hust-peiit@163.com) 免费索取。

由于作者水平有限,书中难免有错误及疏漏之处,恳请同行专家及读者指正,以便进一步提高本书质量。

编者  
2016 年 12 月

目  
录

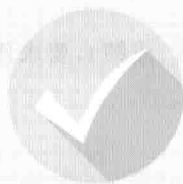
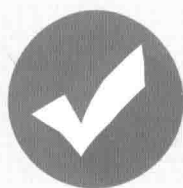
## CONTENTS

实验 1 结构体基本练习实验 .....	(2)
一、实验目的 .....	(2)
二、基本实验练习 .....	(2)
三、自己动手做练习 .....	(3)
四、综合应用实例 .....	(4)
实验 2 顺序表实验 .....	(5)
一、实验目的 .....	(5)
二、基本实验练习 .....	(5)
三、自己动手做练习 .....	(6)
四、综合应用实例 .....	(7)
实验 3 单链表实验 .....	(8)
一、实验目的 .....	(8)
二、基本实验练习 .....	(8)
三、自己动手做练习 .....	(10)
四、综合应用实例 .....	(10)
实验 4 栈实验 .....	(12)
一、实验目的 .....	(12)
二、基本实验练习 .....	(12)
三、自己动手做练习 .....	(14)
四、综合应用实例 .....	(14)



<b>实验 5 队列实验</b>	(19)
一、实验目的	(19)
二、基本实验练习	(19)
三、自己动手做练习	(21)
四、综合应用实例	(21)
<b>实验 6 串实验</b>	(23)
一、实验目的	(23)
二、基本实验练习	(23)
三、自己动手做练习	(24)
四、综合应用实例	(24)
五、注意事项及小结	(28)
<b>实验 7 二叉树实验</b>	(29)
一、实验目的	(29)
二、基本实验练习	(29)
三、自己动手做练习	(31)
四、综合应用实例	(31)
<b>实验 8 图实验</b>	(33)
一、实验目的	(33)
二、基本实验练习	(33)
三、自己动手做练习	(36)
四、综合应用实例	(37)
<b>实验 9 查找实验</b>	(39)
一、实验目的	(39)
二、基本实验练习	(39)
三、自己动手做练习	(41)
四、综合应用实例	(42)
<b>实验 10 排序实验</b>	(44)
一、实验目的	(44)
二、基本实验练习	(44)
三、自己动手做练习	(46)
四、综合应用实例	(46)

综合实验 1 线性结构综合性实验 .....	(48)
一、实验目的 .....	(48)
二、实验要求 .....	(48)
三、详细实验步骤要求 .....	(48)
四、提交作业要求 .....	(49)
综合实验 2 树形结构综合性实验 .....	(50)
一、实验目的 .....	(50)
二、实验要求 .....	(50)
三、详细实验步骤要求 .....	(50)
四、提交作业要求 .....	(50)
第 1 章课后习题解析 .....	(53)
第 2 章课后习题解析 .....	(56)
第 3 章课后习题解析 .....	(61)
第 4 章课后习题解析 .....	(67)
第 5 章课后习题解析 .....	(71)
第 6 章课后习题解析 .....	(83)
第 7 章课后习题解析 .....	(95)
第 8 章课后习题解析 .....	(106)
自测练习题 .....	(113)
参考文献 .....	(118)



Part 1  
SHIYAN  
ZHIDAO

第1篇

实验指导





## 一、实验目的

掌握结构体类型变量的定义和使用。



## 二、基本实验练习

有  $n$  个学生的信息(包括学号、姓名、成绩),要求用键盘输入上述各项数据,按照成绩从低到高的顺序输出各学生的信息。

### 1. 定义一个结构体类型

```
typedef struct student
{
    int no;          /* 学号 */
    char name[20];  /* 姓名 */
    double score;   /* 成绩 */
}STU;
```

### 2. 创建结构体数组

#### 分析

为将创建好的结构体数组带回到主调函数,这里使用数组名作为形式参数。

```
void CreateStu(STU s[],int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("请输入第%d个同学的学号:\n",i+1);
        scanf("%d",&s[i].no);
        printf("请输入第%d个同学的姓名:\n",i+1);
        scanf("%s",s[i].name);
        printf("请输入第%d个同学的成绩:\n",i+1);
        scanf("%lf",&s[i].score);
    }
}
```

### 3. 对结构体数组排序

#### 分析

这里采用冒泡排序对结构体数组进行排序,flag 是数组元素交换的标志位,当一趟排序未出现两两元素交换则结束排序。

```
void SortStu(STU s[],int n)
{
    int i,j,flag;
    STU t;
    for(i=0;i<n-1;i++)
```

```

{
    flag=1;
    for(j=0;j<n-1-i;j++)
        if(s[j+1].score<s[j].score)
        {
            flag=0;
            t=s[j];
            s[j]=s[j+1];
            s[j+1]=t;
        }
    if(flag)
        return;
}
}
}

```

#### 4. 输出结构体数组

```

void PrintStu(STU s[],int n)
{
    int i;
    printf("   学号   姓名   成绩\n");
    for(i=0;i<n;i++)
    {
        printf("%6d%9s%10.1lf\n",s[i].no,s[i].name,s[i].score);
    }
}
}

```

### 三、自己动手做练习

在上述建立的学生信息结构体数组中,编写一个函数 `int SearchStu(STU s[],int n,int no)` 来实现在长度为 `n` 的学生信息结构体数组 `s` 中查找学号为 `no` 的学生,若找到了,输出查找到学生的完整信息,并返回学生所在的数组下标;否则,提示查找失败,返回 `-1`。实现后的程序运行界面如图 1-1 所示。

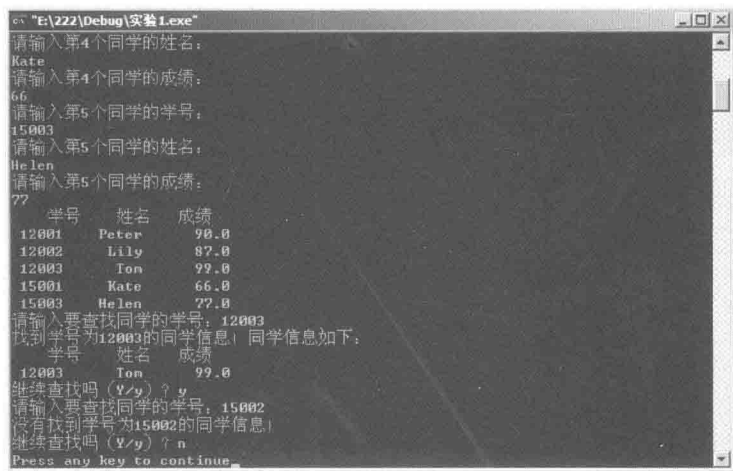


图 1-1 实验 1 练习运行界面

## 四、综合应用实例

在上述建立的学生信息结构体数组中,调用 int SearchStu(STU s[],int n,int no)函数在长度为 n 的学生信息结构体数组 s 中查找学号为 no 的学生,若找到了,删除该学生的信息。

**分析** 在数组中删除指定的元素后,需要将删除元素之后的元素向前移动一个位置。

学生信息结构体数组的删除算法如下。

```
void DelStu(STU s[],int n,int t)
/*在长度为 n 的学生信息结构体数组 s 中删除学号为 no 的学生信息*/
{
    int i;
    for(i=t;i<n-1;i++)
    {
        s[i]=s[i+1]; /*数组元素前移*/
    }
}
```

## 实验 2 顺序表实验

### 一、实验目的

掌握顺序表中基本运算的实现,并能够利用基本运算进行综合应用。

### 二、基本实验练习

现有一组学生的信息,每个学生的信息包含 2 项内容,分别为学号(int 型)和成绩(float 型),要求对这组学生信息进行如下操作。

#### 1. 建立顺序表

将学生信息建立一个顺序表。

**分析** (1)为了增加程序的通用性,学生人数及学生信息都由用户输入,可以用于不同人数的情况。

(2)建立顺序表,即将数据存在数组中,并且注意保存顺序表的实际长度。其具体代码如下。

```
#define MaxLen 100
typedef struct {
    int num; /* 学生学号 */
    float score; /* 学生分数 */
} stu; /* 学生信息数据类型说明 */
typedef stu DataType;
typedef struct {
    DataType data[MaxLen];
    int length;
} SeqList; /* 顺序表数据类型说明 */
void CreatList(SeqList *L)
{
    int i, k, n;
    printf("请输入学生人数(小于等于%d):\n", MaxLen);
    scanf("%d", &n);
    i = 1;
    while(n > MaxLen)
        /* 若输入的人数超过申请的存储空间,对用户进行提示。 */
        {
            if(i < 3) /* 给用户 3 次输入机会 */
            {
                printf("存储空间不足,请重新输入学生人数:");
                scanf("%d", &n);
                i++;
            }
        }
}
```

```

    }
    else
    {
        printf("输入数据不正确,程序将退出!");
        exit(1);
    }
}/*while循环增强程序的健壮性,使运行界面更友好。*/
printf("请输入学生学号(int)和成绩(float):\n");
for(i=0;i<n;i++)
    scanf("%d%f",&L->data[i].num,&L->data[i].score);
L->length=n;
return;
}

```

## 2. 输出顺序表信息

将建立的顺序表信息输出。

```

void PrintList(SeqList *L)
{
    int i;
    printf("学生学号和成绩分别为:\n");
    for(i=0;i<L->length;i++)
        printf("%d %5.2f\n",L->data[i].num,L->data[i].score);
    return;
}

```

## 3. 修改成绩

在这组学生中找到成绩为 59 分的同学,并且将成绩改为 60。

### 分析

(1)在教材的基本运算中已经实现了顺序表中查找值为  $x$  的元素的序号,只要调用该函数找到第  $i$  个元素,将其成绩改为 60 即可。

(2)下面的自定义函数只能将找到的第 1 个 59 的成绩改为 60,思考如何将顺序表中所有的 59 改为 60。

```

void AlterList(SeqList *L)
{
    int i;
    i=LocateList(L,59);
    /*注意查找函数的第2个参数的数据类型,不是 DataType,而是 stu的第2个成员的数据类型,即 float型。*/
    L->data[i-1].score=60;
}

```

## 三、自己动手做练习

在上述建立的学生信息顺序表中,找到成绩最高的学生,并且将学生的完整信息输出。

### 分析

只要找到成绩最高学生的序号,将其对应的学号、成绩输出即可。

要求实现后的程序运行界面如图 2-1 所示。

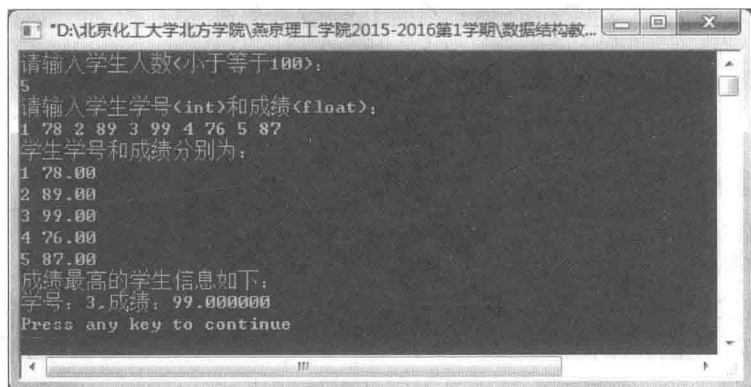


图 2-1 实验 2 练习运行界面

## 四、综合应用实例

实现顺序表的就地逆置,即利用顺序表原来的存储空间将线性表 $(a_1, a_2, \dots, a_n)$ 逆置为 $(a_n, a_{n-1}, \dots, a_1)$ 。

**分析** (1)因为顺序表中元素是按逻辑关系顺序存储的,所以要实现逆置,只要将顺序表中对称位置元素交换。

(2)顺序表中元素的个数即顺序表长度为 $L \rightarrow \text{length}$ 。

顺序表逆置算法具体如下。

```
void InverseList(SeqList *L)
{
    int i, n;
    DataType x;
    n=L->length;
    for(i=0; i<n/2; i++)
    {
        x=L->data[i];
        L->data[i]=L->data[n-i-1];
        L->data[n-i-1]=x;
    }
}
```

## 一、实验目的

- (1) 掌握单链表中基本运算的实现,并能够利用基本运算进行综合应用。
- (2) 了解链式存储结构与顺序存储结构运算实现的区别。
- (3) 理解同一问题在不同的数据存储结构下有不同的实现方法。其中,实验3和实验2的实验题目是完全一样的,但因为采用了不同的存储结构,所以实现方法不同。

## 二、基本实验练习

现有一组学生的信息,每个学生的信息包含2项内容,分别为学号(int型)和成绩(float型),要求对这组学生信息进行如下操作。

### 1. 建立单链表

将学生信息建立一个单链表。

**分析** (1) 为了增加程序的通用性,学生人数及学生信息都由用户输入,可以用于不同的人数的情况。

(2) 每个学生信息存储于一个结点中,结点存储空间需动态申请。

其具体代码如下。

```
typedef struct {
    int num; /*学生学号*/
    float score; /*学生分数*/
} stu; /*学生信息数据类型说明*/
typedef stu DataType;
typedef struct node{
    DataType data;
    struct node *next;
}ListNode, *LinkList;
LinkList CreatList()
{ /*尾插法建立单链表*/
    DataType x;
    int i, n;
    ListNode *s, *r;
    printf("请输入单链表中元素的个数:\n");
    scanf("%d", &n);
    LinkList head= (ListNode *)malloc(sizeof(ListNode)); /*头结点*/
    if(head==NULL)
    {
        printf("分配空间失败!");
```

```

        exit(1);
    }
    r=head;
    printf("请输入%d个单链表中的元素,每个元素包括:学号(int),成绩(float)\n",
n);
    for(i=0;i<n;i++)
    {
        s=(ListNode *)malloc(sizeof(ListNode));/*为新插入的结点申请空间*/
        if(s==NULL)
        {
            printf("分配空间失败!");
            exit(1);
        }
        scanf("%d%f",&s->data.num,&s->data.score);
        r->next=s;
        r=s;
    }/*r为尾指针*/
    r->next=NULL;/*单链表的最后一个指针为空*/
    return head;
}

```

## 2. 输出单链表

将建立的单链表信息输出。

其具体代码如下。

```

void PrintList(LinkList head)
{
    ListNode *p;
    p=head->next;
    while(p!=NULL)
    {
        printf("%d %5.2f\n",p->data.num,p->data.score);
        p=p->next;
    }
    printf("\n");
}

```

## 3. 修改成绩

在这组学生中找到成绩为 59 分的同学,并且将成绩改为 60。

### 分析

(1)在课本的基本运算 LocateList()函数中已经实现了单链表中查找运算,并有两种返回值,一种是返回查找元素的指针,一种是返回查找元素的序号。本题中利用返回查找元素指针的自定义函数 ListNode \* LocateListp(LinkList L,DataType x)实现更方便。

(2)下面的自定义函数只能将找到的第 1 个 59 的成绩改为 60,思考如何将单链表中所有的 59 改为 60。



```
void AlertList(LinkList head)
{
    ListNode *p;
    p=LocateListp(head,59);/*注意 LocateListp 第 2 个参数数据类型,同实验 2*/
    p->data.score=60;
}
```

### 三、自己动手做练习

在上述建立的学生信息表中,找到成绩最高的学生,并且将学生的完整信息输出。

**分析** 只要找到成绩最高学生的指针,将该指针指向结点的学号、成绩输出即可。

要求实现后的程序运行界面如图 3-1 所示。

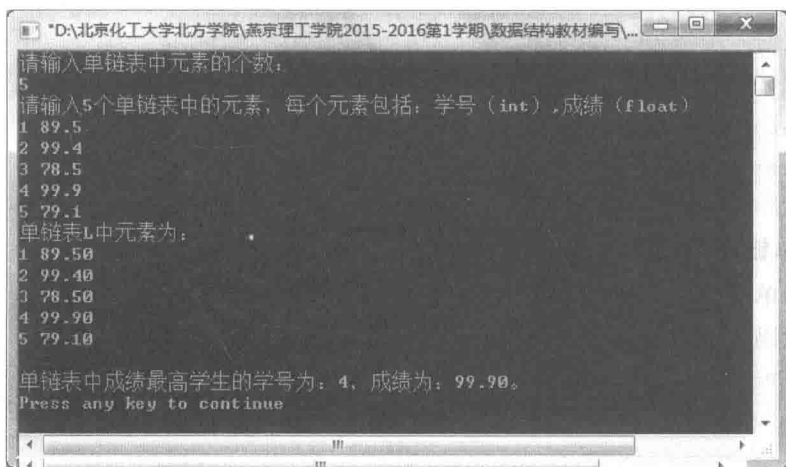


图 3-1 实验 3 练习运行界面

**此题注意:**因为要返回的是成绩最高学生的所有信息,所以需要使用一个指针记住最高成绩的地址并将其返回。

### 四、综合应用实例

实现单链表的就地逆置,即利用单链表原来的存储空间将线性表 $(a_1, a_2, \dots, a_n)$ 逆置为 $(a_n, a_{n-1}, \dots, a_1)$ 。

**分析** (1)因为单链表中基本的存储单元为结点,且元素之间的逻辑关系通过结点的后继指针来体现,所以想实现单链表的就地逆置,只要改变每个结点的后继指针即可。

(2)逆置时,可依次从原链表中取下结点,然后采用头插法重新建立新的单链表。

单链表逆置算法如下。