实验 4 单链表

4.1 实验目的

1. 熟练掌握单链表的类型定义和基本操作算法（以建立、插入、删除、遍历、排序和归并等操作为重点）的实现。

2. 通过实验加深对 C 语言的使用（特别是函数调用、结构体和指针的应用）。

3. 掌握模块化程序设计方法。

4.2 预备知识

1. 单链表类型定义

```
typedef  struct Node
{
    ElemType data;        //数据域
    struct Node *next;    //指针域：指向直接后继
}LNode, *LinkList;        //单链表类型
```

2. 单链表的基本操作

⑴ 初始化 InitList(&L)

构造一个空链表 L。

⑵ 销毁链表 DestroyList(&L)

释放链表 L 占用的内存空间。

⑶ 判定是否为空表 ListEmpty(L)

返回一个值表示 L 是否为空表。若 L 为空表，则返回 1，否则返回 0。

⑷ 求链表的长度 ListLength(L)

返回链表 L 的长度。实际上只需返回 length 成员的值即可。

⑸ 输出链表 DispList(L)

当表 L 不为空时，顺序输出 L 中各数据元素的值。

⑹ 求某个数据元素值 GetElem(L, i, &e)

该运算返回 L 中第 i(1≤i≤ListLength(L))个元素的值，存放在 e 中。

⑺ 按元素值查找 LocateElem(L, e)

顺序查找第 1 个值域与 e 相等的数据元素的序号。若这样的元素不存在，则返回值为 0。

⑻ 插入数据元素 ListInsert(&L, i, e)

在链表 L 的第 i 个位置(1≤i≤ListLength(L)+1)上插入新的元素 e。

⑼ 删除数据元素 ListDelete(&L, i, &e)

删除链表 L 的第 i(1≤i≤ListLength(L))个元素。

⑽ 清空链表 ClearList(&L)

删除链表 L 中的所有数据元素，但不释放已分配给链表的存储空间。

4.3 实例：单链表的构建和输出

用头插法构建一个非空单链表并输出链表中各元素。

参考代码：

```
#include <stdio.h>
#include <stdlib.h>
typedef int ElemType;  //数据元素类型
typedef  struct Node
{
    ElemType data;
    struct Node *next;  //指向直接后继
} LNode, *LinkList;     //单链表类型

//基本操作的函数声明
void InitList(LinkList &L);
void DispList(LinkList L);

//头插法建立非空单链表
void InitList(LinkList &L)
{
    int i, n; //n 为元素个数
    LinkList s;
    L=(LinkList)malloc(sizeof(LNode)); //给头结点分配内存空间
    L->next=NULL;
    printf("n=? ");
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        s=(LinkList)malloc(sizeof(LNode)); //给新结点分配内存空间
        printf("\n data %d=? ",i);
        scanf("%d",&(s->data)); //为新结点的数据域赋值
        s->next=L->next;         //为新结点的指针域赋值，将新结点插入到 L 中
```

```
        L->next=s;
    }
} //InitList

//输出单链表
void DispList(LinkList L)
{
    LinkList p;
    p=L->next;  //利用 p 遍历链表
    printf("\n 链表为: ");
    if(!p)
        printf("空");
    else
    {
        while(p)  //遍历链表，并输出各结点值
        {
            printf("%d ", p->data);
            p=p->next;
        }
    }
    printf("\n");
}//DispList

//主函数，测试上述定义的类型及基本操作
void main()
{
    int k;
    LinkList l;
    printf("\n\n=================================");
    printf("\n   1. 建立单链表" );
    printf("\n   2. 输出单链表");
    printf("\n   3. 结束程序运行");
    printf("\n=================================");
    do
    {
        printf("\n 请选择(1-3): ");
        scanf("%d", &k);
        switch(k)
        {
        case 1: InitList(l); break; //创建单链表 l
        case 2: DispList(l); break; //输出链表中各结点值
        } //switch
```

```
    }while(k!=3);
    printf("\n再见! \n");
}//main
```

4.4 实验：实现单链表的基本操作并进行测试

1. 用 C 语言实现带头结点单链表的 ADT（包括单链表的类型定义和基本操作）。

2. 在 1 基础上设计一个主程序完成如下功能：（假定数据元素的类型为整型）

⑴ 初始化单链表 L；

⑵ 输出链表 L；

⑶ 销毁链表 L；

⑷ 依次采用尾插法插入 12, 56, 7, 3, 89, 21, 123, 20 元素，并输出链表 L；

⑸ 输出 L 的长度；

⑹ 判断 L 是否为空；

⑺ 输出链表 L 的第 i 个元素；

⑻ 输出指定元素的位置；

⑼ 在第 i 个位置上插入某元素；

⑽ 删除链表中的指定元素；

以上为必做题。下面为选做题：

⑾ 将单链表 L 按数据元素值由小到大排序，并输出排序后的 L。

3. 其他要求

⑴ 单链表的类型定义和基本操作函数声明放在 LinkList.h 文件；

⑵ 基本操作的实现放在单独的 LinkList.cpp 文件；

⑶ 测试程序放在 LinkListTestApp.cpp 中。