**理工学院计算机类课程实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | 数据结构 | | 实验名称 | 实验2 Project 单链表 |
| 小组名称及成员 | 吴警 | | | |
| 专 业 | 软件工程 | | 年 级 | 2019 |
| 班 级 |  | | 实验时间 | 第4周 |
| 教师评语（或成绩） | | 教师签字：  年 月 日 | | |
| **一、实验目的**  （1）了解C项目编写方式  （2）熟悉单链表的基本运算以及算法与程序之间的关系 | | | | |
| **二、实验设备（工具、材料、硬软件）及要求**  1、设备：计算机一台、Dev C++  2、具体要求：  实现单链表的基本算法：  （1）创建一个单链表  （2）单链表的查找  （3）在单链表中增加一个元素  （4）在单链表中删除一个元素 | | | | |
| **三、实验过程**  **要求：（1）列出实现的源代码、运行效果截图；（2）为源码添加必要的注释语句。另外，对实验进一步分析存在的问题提出解决办法，或总结实验所取得的经验均可写入报告中。能力强者也可使用java语言实现代码。**  **源代码：**  **main.c：**  **#include <stdio.h>**  **#include <stdlib.h>**  **#include "linklist.h"**  **int main(int argc, char \*argv[]) {**  **LinkList createLink\_tail();//尾插入**  **//LinkList createLink\_head();//头插入**  **int searchLink(LinkList L,int location,ElemType \*e);**  **int insertLink(LinkList L,int location,ElemType x);**  **int deleteLink(LinkList L,int location);**  **void printLink(LinkList L);**    **LinkList L;**  **//L=createLink\_head();**  **L=createLink\_tail();**  **printLink(L);**    **int t=0;**  **int e,location;**  **printf("请输入要查找的位置:");**  **scanf("%d",&location);**  **t=searchLink(L,location,&e);**  **if(t>0){**  **printf("找到了,是 %d\n",e);**  **}**  **else if(t==-1){//查找位置不合法**  **printf("没找到!\n");**  **}**    **int x;**  **printf("请输入要插入的位置及插入的数 \n");**  **scanf("%d %d",&location,&x);**  **t=insertLink(L,location,x);**  **if(t>0){**  **printLink(L);//打印插入元素x后的单链表L**  **}**  **else if(t==-1){**  **printf("输入的位置出错\n");**  **}**    **printf("请输入要删除的位置:");**  **scanf("%d",&location);**  **t=deleteLink(L,location);**  **if(t>0){**  **printLink(L);//打印删除元素x后的单链表L**  **}**  **else if(t==-1){**  **printf("输入的位置出错\n");**  **}**  **return 0;**  **}**  **linklist.h**  **//创建单链表结构体**  **typedef int ElemType;**  **struct LNode{**  **ElemType data;**  **struct LNode \*next;**  **};**  **typedef struct LNode LNode;**  **typedef struct LNode \*LinkList;**  **printLink.c**  **#include <stdio.h>**  **#include "linklist.h"**  **void printLink(LinkList L){**  **LinkList p;**  **p=L->next;**  **while(p!=NULL){//判断p是否为空**  **printf("%5d",p->data);**  **p=p->next;//遍历单链表L所有元素**  **}**  **printf("\n");**  **}**  **createLink\_head.c**  **#include <stdio.h>**  **#include <malloc.h>**  **#include "linklist.h"**  **LinkList createLink\_tail(){**  **LinkList L,p,tail;//tail是最后一个结点**  **int n;**  **L=(LinkList)malloc(sizeof(LNode));//为结点L申请空间**  **L->next=NULL;**  **tail=L;**  **printf("单链表长度:");**  **scanf("%d",&n);**  **int i,x;**  **printf("输入元素:");**  **for(i=1;i<=n;i++){**  **p=(LinkList)malloc(sizeof(LNode));//为新结点p申请空间**  **scanf("%d",&x);**  **p->data=x;//将数据放入新结点data**  **tail->next=p;//核心代码 将新元素p接入单链表L中**  **tail=p;//将新元素p接在L->next的位置**  **}**  **tail->next=NULL;**  **return L;**  **}**  **createLink\_head.c**  **#include <stdio.h>**  **#include <malloc.h>**  **#include "linklist.h"**  **//头插法**  **LinkList createLink\_head(){**  **LinkList L,p;**  **int n;**  **L=(LinkList)malloc(sizeof(LNode));//为结点L申请空间**  **L->next=NULL;**  **printf("单链表长度:");**  **scanf("%d",&n);//输入单链表长度n**  **int i,x;**  **printf("输入元素:");**  **for(i=1;i<=n;i++){**  **p=(LinkList)malloc(sizeof(LNode));//为新结点p申请空间**  **scanf("%d",&x);**  **p->data=x;//将数据放入新结点data**  **p->next=L->next;//核心代码 将新元素p接入单链表L中**  **L->next=p;//将新元素p接在L->next的位置**  **}**  **return L;**  **}**  **searchLink.c**  **#include <stdio.h>**  **#include "linklist.h"**  **int searchLink(LinkList L,int location,ElemType \*e)**  **{**  **int j=1;**  **LinkList p;**  **p=L->next;**  **while(p!=NULL&&j<location){//判断L中location位置上的元素合法性**  **p=p->next;**  **j++;**  **}**  **if(p==NULL||j>location){**  **return -1;//不合法**  **}**  **\*e=p->data;**  **return 1;**    **}**  **insertLink.c**  **#include <stdio.h>**  **#include <malloc.h>**  **#include "linklist.h"**  **int insertLink(LinkList L,int location,ElemType x){**  **LinkList p,q;**  **p=L;**  **int j=0;**  **while(p&&j<location-1){**  **p=p->next;**  **j++;**  **}**  **if(p==NULL||j>location-1)**  **return -1;**  **q=(LinkList)malloc(sizeof(LNode));//为新结点q申请空间**  **q->data=x;//将数据放入新结点data**  **q->next=p->next;//核心代码 将新元素q接入单链表L中**  **p->next=q; //将新元素q接在p->next的位置**  **return 1;**  **}**  **deleteLink.c**  **#include <stdio.h>**  **#include <malloc.h>**  **#include "linklist.h"**  **int deleteLink(LinkList L,int location){**  **LinkList p,q;**  **p=L;**  **int j=0;**  **while(p&&j<location-1){**  **p=p->next;**  **j++;**  **}**  **if(p==NULL||j>location-1)//判断删除位置是否合法**  **return -1;//插入的位置有误**  **q=p->next;**  **p->next=q->next;//核心代码 删除结点q**  **free(q);//释放结点q的空间**  **return 1;**  **}**  **运行结果：**    **实验过程中遇到的困难：**  **(1) 整体实验代码是参考老师的发的视频完成的，对.h文件的创建还不是特别理解。**  **(2) 插入和删除元素的过程，缺少了对插入和删除元素的指定位置进行判断。**  **改进措施：**  **(1) 课后及时对自己不懂的知识进行学习巩固。**  **(2) 对数组相关算法要考虑全面，考虑是否越界及合法。**  **收获：**  **(1) 掌握了二分查找的算法。**  **(2) 基本了解C项目的编写方式。**  **(3) 学会创建简单顺序表并能够在指定位置插入、删除元素及查找指定元素。** | | | | |