**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 | 左康田 姜 来 蔡杰铭 韦清秀 童明安 | | | |
| 专 业 | 软件工程 | | 年 级 | 2020级 |
| 班 级 | 软工1班 | | 实验时间 | 2021.3.30~2021.4.5 |
| 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。**  **/\*神仙打架\*/**  **//童明安**  **#include <stdio.h>**  **#include <stdlib.h>**  **#include "LinkList.h"**  **int main(int argc, char \*argv[]) {**  **//函数声明{**  **LinkList createlink\_tail(LinkList L);**  **void printfLink(LinkList L);**  **int findLink(LinkList L,int i);**  **int insertLink(LinkList L,int x,int location);**  **int delLink\_x(LinkList L,int x);**  **//函数声明}**  **LinkList L;**  **int i,s,location2,location1,x;**  **L =createlink\_tail(L);//调用表尾输入法**  **printfLink(L);//调用输出函数**  **printf("\n请输入您要添加的元素和位置");**  **scanf("%d %d",&x,&location1);**  **insertLink(L,x,location1); //调用增加函数**  **printfLink(L);//调用输出函数**  **printf("\n请输入您要删除的元素");**  **scanf("%d",&location2);**  **delLink\_x(L,location2);//调用删除函数**  **printfLink(L);//调用输出函数**  **return 0;**  **}**  **/\*左康田\*/**  **LinkList createlink\_tail(LinkList L){**  **L=(LinkList)malloc(sizeof(LinkList));**  **L->next=NULL;//头结点**  **LinkList tail,p;//缔造新节点**  **tail=L;**  **int i,n;**  **printf("\n请输入您要保存的数的个数：");**  **scanf("%d",&n);**  **for(i=1;i<=n;i++){//用n次循环输进n个数**  **p=(LinkList)malloc(sizeof(LNode));//创造一个像LNode那么大的空间**  **scanf("%d",&p->data);**  **tail->next=p;//将表尾接向P**  **tail=p;//更新表尾**  **}**  **tail->next=NULL;//将最后的表尾指向NULL**  **return L;**  **}**  **/\*姜来\*/**  **void printfLink(LinkList L)**  **{**  **LinkList p;**  **p=L->next;**  **while(p!=NULL){**  **printf("%6d",p->data);**  **p=p->next;**  **}**  **}**  **/\*姜来\*/**  **int findLink(LinkList L,int i)**  **{ int x=1,s=0;//S即为Flag**  **LinkList p;**  **p=L->next;**  **while(p!=NULL)**  **{**  **if(x==i) {s=1;break;}**  **x++;**  **p=p->next;**  **}**  **return s;//用于判断链表是否能能达到该位置**  **}**  **/\*韦清秀 &(童明安) \*/**  **int insertLink(LinkList L,int x,int location)**  **{**  **int k,j;**  **LinkList q,p;**  **q=(LinkList)malloc(sizeof(LNode));**  **p=L->next;**  **q->data=x;**  **k=findLink(L,location-1);//调用寻找函数**  **if(k==1){**  **q->next=p->next;//将想插入元素指向插入位置前一个元素的初始后续地址**  **p->next=q;//将 前一个元素的地址指向想插入的元素**  **return L;**  **}else printf("程序无法接入该位置");**  **}**  **/\*蔡杰铭\*/**  **int delLink\_x(LinkList L,int x)**  **{**  **LinkList p,q;//定义两个指针**  **p=L->next;//第一个指针**  **int flag=0;**  **while(p!=NULL){**  **if(p->data==x){//判断是否找到**  **flag=1;**  **break;}**  **q=p;//将所需删除元素的上一个指针赋 给q**  **p=p->next;**  **}**  **if(flag==1){**  **q->next=p->next;//删除x**  **free(p);//释放p**  **}**  **else ("没有找到您想删除的元素");**  **return L;**  **}**    **本次作业由姜来同学负责思考题目需求，得出单链表的框架结构，进而得出头文件内容，并撰写简单的main函数。而后童明安同学分配编写函数任务—左康田同学负责单链表的输入、姜来同学负责编写单链表的输出函数和寻找函数、韦清秀同学负责编写单链表的增加元素功能、蔡杰铭同学负责编写单链表删除元素的函数。最后童明安同学负责主函数main的调用修补，整合整个项目的源代码，并对各个函数进行检查以及进一步注释，寻找代码的不足和错误，最后进行修改并完成实验报告project2。**  **错误：**   1. **使用NULL没有使用对应头文件** 2. **tail（用于接入表尾所定义）没有接入表尾**      1. **增加元素的函数未成功编写**   **A（没有增加函数的位置或没有调用寻找函数）**  **B（编写出一个死循环）**    **改进：**   1. **输入、输出、寻找编写的函数完全没有注释，非常不方便整合项目代码** 2. **输入、输出、寻找函数几乎没有任何改进。** 3. **整个作业编写成员态度散漫，耗时长，代码早早完成发现改进点也毫无改进想法** 4. **寻找函数调用时不完善，当想插入元素位置为1时**     **只需在查找函数中额外判定i=0的情况**       1. **团队任务分配不合理。**   **第三次小组数据结构经验小结**  **第三次作业完成，整个作业耗时更长，质量缺更差。团队分配任务固定化（组员依赖性极高），缺少自我责任。就代码完成质量情况看，长久以往，作业难度增加则作业质量越差，对该次作业知识内容不懂的仍然不懂，不利于个人发展。**  **就本次代码编程，可以先考虑各个需要完成的功能函数之前的联系（如本次寻找函数和增加元素函数），优先完成可多次利用的函数，方便后续函数的编写。** | | | | |