《数据结构》实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学号 | 2016010207 | 姓名 | 孙博文 | 专业 | 石油工程 | 时间 | 2018.4.1 |

1. **实验题目**

顺序表基本操作

1. **实验目的**

1. 熟练掌握线性表的链式存储方式下，基本操作的实现算法，巩固和体会单链表操作特点；

2. 理解体会动态内存申请与释放；

3. 通过本次实验，熟练掌握C语言指针的使用。

**三、算法设计分析**

（一）数据结构的定义

这次实验我没有参考课本编排，独立实现了一遍。

在这个自建的单链表数据结构中，我定义了结构体LinkTable作为每一个节点的类型。

LinkTable中含有两个成员，分别为数据Data和指针Next。

为了便于实现，我没有重定义Data的数据类型，而是直接用了最简便的int型；指针Next采用了重命名，将LinkTable型指针重命名为Next，作为指向链表下一个元素的桥梁，Next与逻辑结构相符合，便于理解。

最后，我将struct LinkTable重定义为LT，减少码字量。

（二）总体设计

实验由主函数、单链表建立函数、插入函数、删除函数、查找函数、销毁函数组成。

1. 主函数：调用各个函数，达成操作

void main()；

1. 线性表建立函数：创建一个空的线性表

CreatLinkTable(PtoLT &List);

1. 插入函数：在头节点处插入指定个数个数据。

InsertElement(PtoLT &head);

1. 删除函数：删除指定位置的数据。

DeleteElement(PtoLT &head);

1. 查找函数：按关键字查找指定元素

SearchbyKey(PtoLT head);

（6）打印函数：计算表长并输出单链表

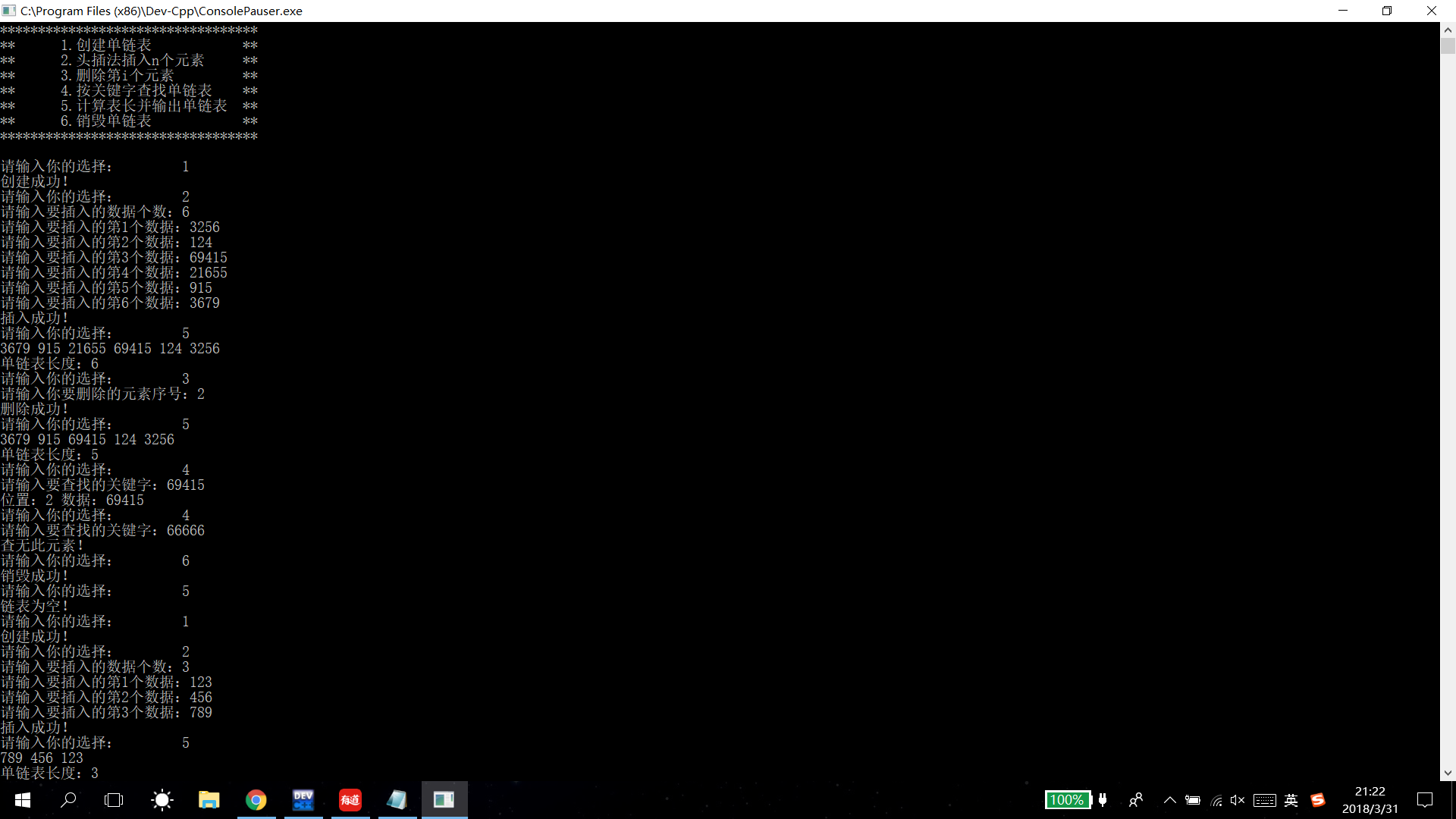
Print(PtoLT head);

（7）销毁函数：销毁整个线性表

DestroyLinkTable(PtoLT &head);

**四、实验测试结果及结果分析**

（一）测试结果（此处给出程序运行截图）



1. 结果分析

首先输入1，创建一个单链表，Next指向NULL，初始存储数据为-1；

然后输入2，插入指定个数个元素，在此处我输入了6，接下来输入了6个数据：3256、124、69415、21655、915、3679。由于采用了头插法，所以头结点中也存放了数据，而且逆序存放。

然后我输入5，打印出了所有数据，和链表长度。

然后我输入3，准备删除某个位置上的元素。我输入2，删除2位置上的元素（表头视为位置0），那么应该会删除21655。显示删除成功。

我输入5，打印出所有数据和表长，显示21655确实被删除了，表长-1。

我输入4，然后输入69415，在整个表中查找关键字为69415的节点。然后输出了位置2，数据69415，正确。

然后我输入了一个不存在的数据66666，返回“查无此元素！”，正确。

然后我输入了6，返回“销毁成功”。

然后我输入了5，尝试打印原先建立的单链表，输出“链表为空！”，打印失败。正确。

然后我创建了一个新表，继续尝试上述基本操作，发现仍然可以实现，说明单链表对象建立无误。

1. **实验总结**

一般情况下，链表头节点中不存放数据。但由于我第一次使用头插法，为了便于实现，所以我在头节点中存放了数据。时间仓促，又没有参考课本，所以如果有小错误，请老师不吝赐教。

**附录 实验程序代码(该部分请加注释)**

#include <iostream>

#include <algorithm>

using namespace std;

typedef struct LinkTable \*PtoLT;

typedef struct LinkTable{

int Data;

PtoLT Next;

}LT;

void CreatLinkTable(PtoLT &List); //创建单链表

void InsertElement(PtoLT &head); //头插法插入n个元素。注：我不清楚头插法头指针存不存放数据，为了便于实现，我没有按惯例让头指针数据为空，而是用它存放了数据，序号为0

void DeleteElement(PtoLT &head); //删除第i个元素

void SearchbyKey(PtoLT head); //按关键字查找单链表

void Print(PtoLT head); //计算表长并输出单链表

void DestroyLinkTable(PtoLT &head); //销毁单链表

int main() {

int n=1;

PtoLT List=(PtoLT)malloc(sizeof(LT));

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\* 1.创建单链表 \*\*\n");

printf("\*\* 2.头插法插入n个元素 \*\*\n");

printf("\*\* 3.删除第i个元素 \*\*\n");

printf("\*\* 4.按关键字查找单链表 \*\*\n");

printf("\*\* 5.计算表长并输出单链表 \*\*\n");

printf("\*\* 6.销毁单链表 \*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

while (n) {

point:;

printf("请输入你的选择： ");

scanf("%d",&n);

switch(n) {

case 1: { //创建一个新的单链表

List=(PtoLT)malloc(sizeof(LT));

CreatLinkTable(List);break;

}

case 2: InsertElement(List);break; //插入元素

case 3: DeleteElement(List);break; //删除元素

case 4: SearchbyKey(List);break; //查找元素

case 5: Print(List);break; //打印表长和所有元素值

case 6: DestroyLinkTable(List);break; //销毁线性表

default: printf("请重新输入!\n");goto point;

}

}

return 0;

}

void CreatLinkTable(PtoLT &List) {

List->Next=NULL;

List->Data=-1;

printf("创建成功！\n");

}

void InsertElement(PtoLT &head) {

int times,t=0;

printf("请输入要插入的数据个数：");

scanf("%d",&times);

while (t++,times--) {

int temp;

PtoLT TP=(PtoLT)malloc(sizeof(LT));

printf("请输入要插入的第%d个数据：",t);

scanf("%d",&temp);

TP->Data=temp;

TP->Next=head;

head=TP;

}

printf("插入成功！\n");

}

void DeleteElement(PtoLT &head) {

int i;

printf("请输入你要删除的元素序号：");

scanf("%d",&i);

PtoLT TP=head;

while (i-1>0)

i--,TP=TP->Next;

PtoLT TP0=TP->Next;

TP->Next=TP0->Next;

free(TP0);

printf("删除成功！\n");

}

void SearchbyKey(PtoLT head) {

int key,des=0;

printf("请输入要查找的关键字：");

scanf("%d",&key);

while (head!=NULL) {

if (head->Data!=key)

++des,head=head->Next;

else break;

}

if (head!=NULL)

printf("位置：%d 数据：%d\n",des,key);

else

printf("查无此元素！\n");

}

void Print(PtoLT head) {

if (!head) {

printf("链表为空！\n");

return;

}

int len=0;

PtoLT TP=head;

while (TP->Next!=NULL) {

printf("%d",TP->Data),len++;

if (TP->Next->Next!=NULL) printf(" ");

else printf("\n");

TP=TP->Next;

}

printf("单链表长度：%d\n",len);

}

void DestroyLinkTable(PtoLT &head) {

while (head!=NULL) {

PtoLT T0=head;

head=head->Next;

free(T0);

}

printf("销毁成功！\n");

}

**实验报告雷同者均视为未做。抄袭请慎重！**

【注意】上机实验报告只提交电子版，要求在课堂上完成并提交。