## 《单链表实验报告》2018204506 高恺元 18软工创新

实验内容：实现单链表的创建，长度计算，插入，删除，倒序输出等操作。

操作环境：C语言编程环境

操作系统：win10

源码

#include <malloc.h>

#include <stdio.h>

#include <stdlib.h>

typedef struct elemtype{

char number[10];

char name[20];

int age;

}person;

typedef struct LNode{

person data;

struct LNode \* next;

}Lnode,\*linklist;

//创建

void Create(linklist &L,int n){

struct LNode \*p;

int i;

L = (linklist)malloc(sizeof(struct LNode));

L->next = NULL;

for(i=n;i>0;--i){

p=(linklist)malloc(sizeof(struct LNode));

printf("输入第 %d 位的编号，姓名，年龄信息：\n",n-i+1);

scanf("%s%s%d",&p->data.number,&p->data.name,&p->data.age);

p->next = L->next;

L->next = p;

}

getchar();

}

//打印

void Print(linklist L){

linklist p;

p = L;

printf("下面是你的单链表：\n");

while(p->next!=NULL){

printf("[%s|%s|%d]->",p->data.number,p->data.name,p->data.age);

p = p->next;

}

printf("打印结束.\n");

}

//求长度

int Total(linklist L){

int t=0;

Lnode \*p = L;

while(p->next != NULL){

t=t+1;

p=p->next;

}

return t;

}

//插入

int Insert(linklist &L,int i,person e){

struct LNode \*p;

struct LNode \*s;

int j;

p = L;

j = 0;

while(p && j<i-1){

p=p->next;

++j;

}

s = (linklist)malloc(sizeof(struct LNode));

s->data = e;

s->next = p->next;

p->next = s;

return 0;

}

//删除

int Delete(linklist &L,int i){

struct LNode \*p;

struct LNode \*q;

int j;

p=L;

j=0;

while(p->next && j<i-1){

p = p->next;

++j;

}

q=p->next;

p->next = q->next;

free(q);

return 0;

}

//逆序输出

void rOutput(linklist &L){

struct LNode \* ohead;

struct LNode \* nhead;

ohead = L->next;

nhead = NULL;

L->next = NULL;

while(ohead != NULL){

nhead = ohead->next;

ohead->next = L->next;

L->next = ohead;

ohead = nhead;

}

Print(L);

}

int main(){

linklist L;

int length;

int n,i;

person e;

printf("请输入人数：");

scanf("%d",&n);

Create(L,n);

Print(L); //创建

length = Total(L);

printf("链表长度为 %d\n",length); //求长度

printf("输入要删除元素的位置：");

scanf("%d",&i);

Delete(L,i);

Print(L); //删除

printf("请输入要插入小e的位置及他的信息：");

scanf("%d%s%s%d",&i,&e.number,&e.name,&e.age);

Insert(L,i,e);

Print(L); //插入

printf("逆序输出链表：\n");

rOutput(L); //逆序输出

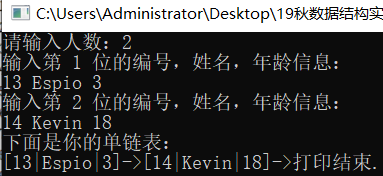
Print(L);

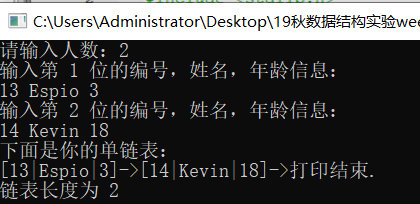
getchar();

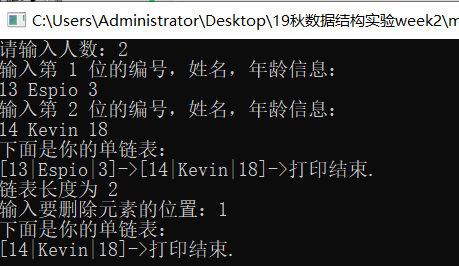
return 0;

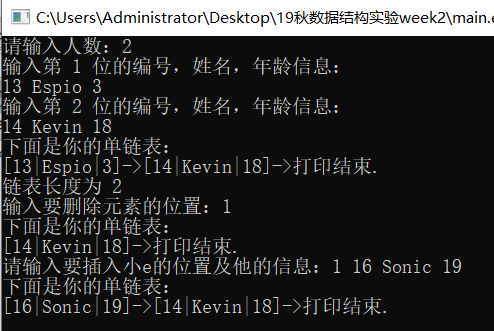
}

运行截图

创建

求长度

删除

插入

逆序输出