**单链表的操作**

实验环境：code::blocks 17.12

实验内容：

链表的建立

链表的打印

求链表的长度

链表的插入操作

链表的删除操作

单链表的逆置

程序代码：#include<malloc.h>

#include <stdio.h>

typedef struct LNode{

int data;

struct LNode \*next;

} LNODE,\*LinkList;

void CreateList(LinkList &L, int n) {

struct LNode \*p;

int i;

L = (LinkList) malloc (sizeof (LNode));

L->next = NULL;

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

{

p = (LinkList) malloc (sizeof (LNode));

scanf("%d",&p->data);

p->next = L->next; L->next = p;

}

}

void PrintList(LinkList L)

{

LinkList p;

p=L;

printf("Begin to print the LinkList...\n");

while(p->next!=NULL)

{

printf("%d->",p->next->data);

p=p->next;

}

printf("----------The LinkList ended!\n");

}

int Listlength(LinkList L)

{

int i=0;

LNODE \*p=L;

while(p->next!=NULL)

{

i++;

p=p->next;

}

return i;

}

int InsertList(LinkList &L,int i,int 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 (LNode));

s->data = e; s->next = p->next;

p->next = s;

return 0;

}

int DeleteList(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;

}

void ReverseList(LinkList &L)

{

struct LNode \*p;

struct LNode \*s;

p=L->next;

s=NULL;

L->next=NULL;

while(NULL!=p)

{

s=p->next;

p->next=L->next;

L->next=p;

p=s;

}

}

int main()

{

LinkList L;

int length;

int n,i,e;

printf("please input the number of elements:\n");

scanf("%d",&n);

CreateList(L,n);

PrintList(L);

length=Listlength(L);

printf("the length of the list is %d\n",length);

printf("input the delete position \n");

scanf("%d",&i);

DeleteList(L,i);

PrintList(L);

printf("Input the insert position and e:\n");

scanf("%d%d",&i,&e);

InsertList(L,i,e);

PrintList(L);

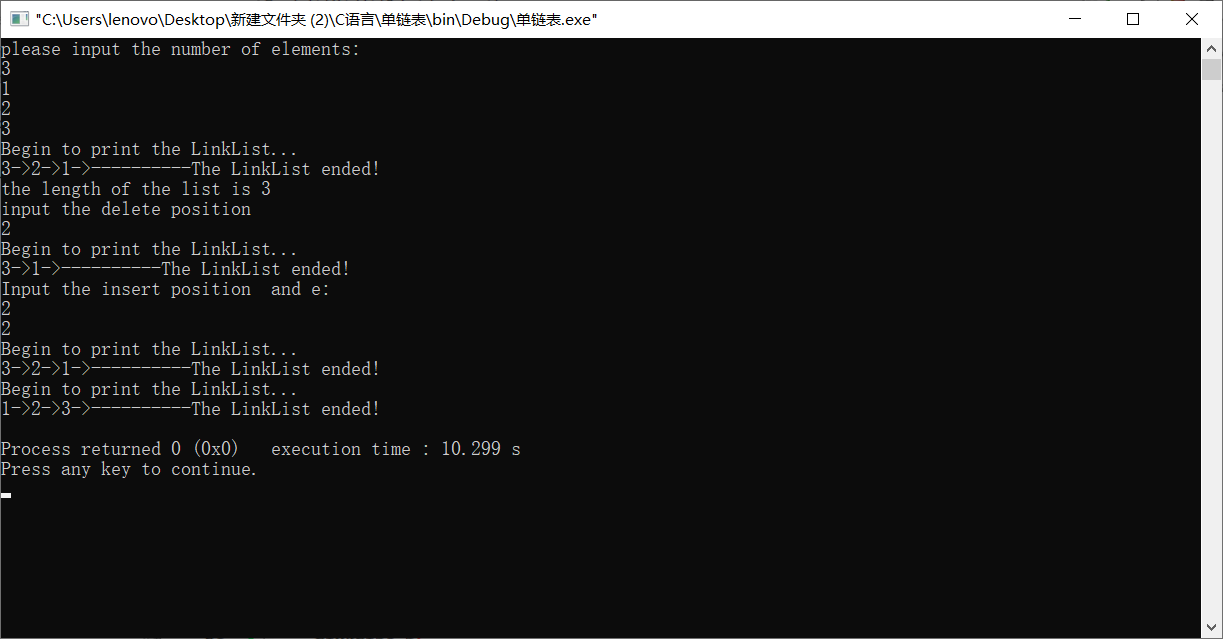
ReverseList(L);

PrintList(L);

getchar();

}

运行截图

****