《顺序表的一些基本操作-实验报告》 2018204506 高恺元 18软工创新

实验内容：实现顺序表的初始化，数据元素的遍历，增、删、改、查等功能。

源码：

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#define OVERFLOW -2

#define LISTINCREMENT 10

#define ListSize 50

typedef struct{//结构的定义

int \* elem; //数据域

int length; //当前表长

int listsize;

}SeqList;

void Create(SeqList &L,int n){//顺序表建立

L.elem=(int\*)malloc((ListSize+1)\*sizeof(int));

if (!L.elem){

exit(0);

}

L.listsize=ListSize;

printf("请输入 %d 组数据：\n",n);

for(int i=1;i<=n;i++){

scanf("%d",&L.elem[i]);

}

L.length=n;

}

void Print(SeqList L){//打印函数

printf("下面为你的顺序表：\n");

for(int i=1;i<=L.length;i++){

printf("%d ",L.elem[i]);

}

}

int Find(SeqList L,int x){//查找

for(int i=1;i<=L.length;i++){

if((L.elem[i])==x){

return i;

}

}

return 0;

}

void Insert(SeqList &L,int i,int e){//插入

int \*p;

int \*q;

int \*newbase;

if(i>=1&&i<=L.length+1){

if(L.length>=L.listsize){

newbase=(int\*)realloc(L.elem,(LISTINCREMENT+L.listsize)\*sizeof(int));

if(!newbase){

exit(OVERFLOW);

}

L.elem=newbase;

L.listsize += LISTINCREMENT;

}

q=&(L.elem[i]);

for(p=&(L.elem[L.length]);p>=q;--p){

\*(p+1)=\*p;

}

\*q=e;

++L.length;

}

else{

printf("操作失败.");

}

}

void Delete(SeqList &L,int i){//删除

int \*p;

int \*q;

if((i>=1)&&(i<=L.length)){

p=&(L.elem[i]);

q=L.elem+L.length;

for(++p;p<=q;++p){

\*(p-1)=\*p;

}

--L.length;

}

else{

printf("操作失败.");

}

}

int main()

{

SeqList L;

L.length=0;

int i,x;

int n;

system("cls"); //清屏

//1.创建顺序表

printf("请输入数据数量：\n");

scanf("%d",&n);

Create(L,n);

Print(L);

//2.查找

printf("\n请输入欲查找数据的位置：");

scanf("%d",&x);

i=Find(L,x);

printf("%d 位置的数据为 %d",x,i);

//3.插入

printf("\n请输入数据插入位置：\n");

scanf("%d",&i);

printf("\n请输入要插入的数据\n");

scanf("%d",&x);

Insert(L, i, x);

Print(L); //打印

//4.删除

printf("\n请输入要删除数据的位置：\n");

scanf("%d",&i);

Delete(L,i);

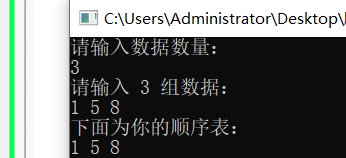
Print(L); //打印

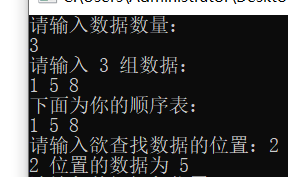
getchar();

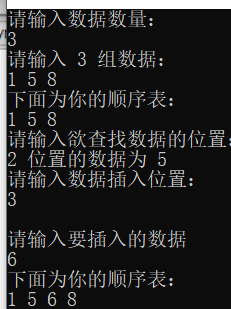
return 0;

}

运行截图

创建

查找

插入

删除