苏州城市学院实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院、系 | 计算科学与人工智能学院 | | 年级专业 | | 22物联网 | | 姓名 | 王子超 | 学号 | 2200443011 |
| 课程名称 | | 数据结构 | | | | | | | 成绩 |  |
| 指导教师 | |  | | 同组实验者 | | 无 | | 实验日期 | 2024/3/4 | |

|  |  |
| --- | --- |
| 实 验 名 称 | 实验二 实现顺序线性表 |

代码：

#include <iostream>

using namespace std;

#define N 100

typedef int array;

typedef struct SeqList {

array a[N];

int length;

} SeqList;

// 初始化

void init(SeqList &L,int a[],int len) {

L.length=len;

for(int i=0;i<L.length;i++){

L.a[i]=a[i];

}

}

//判空

void ifnothing(SeqList &L){

if(L.length==0)

cout<<"这是一个空表!\n";

else

cout<<"这个表有点东西!\n";

}

//清空

void clean(SeqList &L){

L.length=0;

}

//插入元素

bool insert(SeqList &L,int i,array x){

if(i<1 || i>L.length+1 || L.length>=N)

return false;

for(int j=L.length-1;j>=i-1;j--){

L.a[j+1]=L.a[j];

}

L.a[i-1]=x;

L.length++;

return true;

}

//删除元素

bool remove(SeqList &L,int i,array &x){

if(i<1 || i>L.length)

return false;

x=L.a[i-1];

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

L.a[k]=L.a[k+1];

}

L.length--;

return true;

}

//合并表

void hebing(SeqList &L1,SeqList &L2,SeqList &L3){

int i;

for(i=0;i<L1.length;i++){

L3.a[i]=L1.a[i];

}

for(int j=0;j<L2.length;j++,i++){

L3.a[i]=L2.a[j];

}

L3.length=i;

}

// 打印

void traverse(SeqList &L) {

for (int i = 0; i < L.length; i++) {

cout << L.a[i] << " ";

}

}

int main(void) {

SeqList A,B,F;

array x=0;

array a[5]={1,2,3,4,5};

array b[5]={7,8,9,10,11};

init(A,a,5); //初始化一下

ifnothing(A); //判空

clean(A); //清空

ifnothing(A); //判空(第二次)

init(A,a,5); //初始化(第二次)

insert(A,6,6); //在5之后插入元素{1,2,3,4,5,6}

remove(A,2,x); //删去 元素2{1,3,4,5,6}

init(B,b,5); //初始化B

hebing(A,B,F);

cout<<endl<<x<<endl; //打印删去的元素

traverse(F); //打印最终元素

cout<<endl;

system("pause");

return 0;

}

运行结果

