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 | 2024/3/12 | |

|  |  |
| --- | --- |
| 实 验 名 称 | 实验三 实现链表 |

代码：

#include<iostream>

#include<malloc.h>

using namespace std;

typedef struct LNode{

int data;

struct LNode \*next;

}LNode,\*local;

void init(local &head){

head = (local)malloc(sizeof(LNode));

head->next=NULL;

}

void length(local head){

int count = 0;

local p = head->next;

while(p){

count++;

p=p->next;

}

cout<<"链表长度是:"<<count;

}

void clear(local &head){

head->next=NULL;

local p=head;

}

void insert(local &head,int a,int x){

local p = head;

for(int count = 0;count<a-1;count++){

p=p->next;

}

local s = (local)malloc(sizeof(LNode));

s->data = x;

s->next=p->next;

p->next = s;

}

void getlocal(local &head,int i){

local p = head->next;

int count=0;

int x;

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

p=p->next;

count++;

}

x=p->data;

cout<<"第"<<i<<"号位置的元素是:"<<x;

cout<<endl;

}

void remove(local &head,int i){

local p = head;

int x;

for(int count = 0;count<i-1;count++){

p=p->next;

}

local s = (local)malloc(sizeof(LNode));

s=p->next;

x=s->data;

p->next=s->next;

free(s);

cout<<"第"<<i<<"号位置被删除的元素是:"<<x;

cout<<endl;

}

void merge(local &A,local &B,local &C){

local p1=A->next;

local p2=B->next;

local p3=C;

while(p1&&p2){

if(p1->data == p2->data){

p3->next=p1;

p1=p1->next;

p2=p2->next;

}else if(p1->data < p2->data){

p3->next=p1;

p1=p1->next;

}else{

p3->next=p2;

p2=p2->next;

}

p3=p3->next;

}

if(p1){

p3->next=p1;

}else{

p3->next=p2;

}

}

void write(local head){

local p=head->next;

int count=0;

while(p){

int num=p->data;

cout<<num<<" ";

p=p->next;

count++;

}

if(p==NULL&&count==0){

cout<<"这是一个空表!";;

}

cout<<endl;

}

int main(){

local head;

init(head);

for(int i=0;i<10;i++)

insert(head,i+1,i+1);

length(head);

cout<<endl;

getlocal(head,2);

write(head);

remove(head,3);

write(head);

clear(head);

write(head);

local A,B,C;

init(A);

init(B);

init(C);

for(int i=0;i<10;i++)

insert(A,i+1,i\*2);

cout<<"合并前A表:";

write(A);

for(int i=0;i<10;i++)

insert(B,i+1,i+1);

cout<<"合并前B表:";

write(B);

merge(A,B,C);

cout<<"合并表后:";

write(C);

cout<<endl;

return 0;

}

运行结果

