C++代码

#include<iostream>

using namespace std;

typedef struct Lnode

{

int data;

struct Lnode \*next;

}Lnode,\*LinkList;

int ListLength(LinkList L);

void Union(LinkList &La,LinkList Lb);

bool ListInsert(LinkList &L,int i ,int e);

Lnode \*LocateELem(LinkList L,int e);

struct Lnode \*creatlnode();

void print\_l(LinkList L);

int ListLength(LinkList L);

bool GetElem(LinkList L,int i,int &e);

struct Lnode \*creatlnode()

{

struct Lnode \*head ,\*tail,\*p;

int data;

int size=sizeof(struct Lnode);

head=tail=NULL;

cin>>data;

while(data!=-1)

{

p=(Lnode \*)malloc(size);

p->data=data;

p->next=NULL;

if(head==NULL)

head=p;

else tail->next=p;

tail=p;

cin>>data;

}

return head;

}

bool ListInsert(LinkList &L,int i ,int e)

{

LinkList p=L;int j=0;

while(p&&j<i-1){p=p->next;++j;}

if(!p||j>i-1) return false;

LinkList s=new Lnode;

s->data=e;

s->next=p->next;

p->next=s;

return true;

}

Lnode \*LocateELem(LinkList L,int e)

{

LinkList p=L->next;

while(p &&p->data!=e)

p=p->next;

return p;

}

bool GetElem(LinkList L,int i,int &e)

{

LinkList p=L; int j=1;

while(p&&j<i){

p=p->next; ++j;

}

if(!p || j>i) return false;

e=p->data;

return true;

}

void print\_l(LinkList L)

{

LinkList p;

p=L;

while(p)

{

cout<<p->data<<" ";

p=p->next;

}

cout<<endl;

}

int ListLength(LinkList L)

{

LinkList p;

p=L->next;

int a=0;

while(p)

{ a++;

p=p->next; }

return a;

}

int main()

{

int L1\_len,L2\_len,e;

LinkList L1,L2;

cout<<"输入第一个链表（-1停止）：";

L1=creatlnode();

cout<<"输入第二个链表（-1停止）：";

L2=creatlnode();

L1\_len=ListLength(L1);

L2\_len=ListLength(L2);

for(int i=1;i<=L2\_len+1;i++)

{

GetElem(L2,i,e);

if(!LocateELem(L1,e))

ListInsert(L1,++L1\_len,e);

}

print\_l(L1);

return 0;

}