数据结构作业

余健明

机1901

201930310050

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

//定义结构体类型

typedef struct Node {

float xishu; //定义系数为浮点数

int zhishu; //定义指数为整型

struct Node\* Next;

}Node;

typedef struct Node\* PNode; //定义指针类型

//多项式按照指数大小排序

void insertNewPoint\_link(PNode head, PNode qNode)

{

PNode p = head; //向head为首的链表中插入qnode结点，由p从head处开始遍历

PNode h = head;

PNode q; //用于销毁节点

while (p->Next != NULL) //若已输入多项式不止一项

{

//若新项的指数比某一项p大则将新项插在该项前面

if (p->Next->zhishu < qNode->zhishu)

{

qNode->Next = p->Next;

p->Next = qNode;

break; //插完之后即可退出

}

else if (p->Next->zhishu == qNode->zhishu)

{

float sum = p->Next->xishu + qNode->xishu;//计算合并后的系数

if (sum != 0) //若合并后系数不为零

{

p->Next->xishu = sum;//重新赋予p项新的系数

}

else //否则删去已有链表中的第一项

{

q = p->Next;

p->Next = p->Next->Next;

free(q);//销毁结点

q = NULL;

}

break;

}

p = p->Next; //若新项指数小于p当前指向的项则p移至下一项与下一项进行比较

}

if (p->Next == NULL) //若多项式只有一项则无需排序

{

p->Next = qNode;

}

}

//输出多项式

void printLinkeLink(PNode head)

{

PNode temp = head->Next; //建立临时指针用于遍历链表

printf("%fX^%d", temp->xishu, temp->zhishu); //先输出第一项（不用前置符号）

temp = temp->Next; //temp指针指向下一项

while (temp != NULL) //循环输出后续项

{

if (temp->xishu > 0) //若为正系数

printf(" +%fX^%d", temp->xishu, temp->zhishu);

else if (temp->xishu < 0) //若为负系数

printf("%fX^%d", temp->xishu, temp->zhishu);

temp = temp->Next;

}

}

//多项式的加法计算

void add\_poly(Node\* pa, Node\* pb)//pa,pb分别为多项式一和多项式二的头指针

{

Node\* p = pa->Next; //p为遍历指针，此时指向多项式一的第一项

Node\* q = pb->Next; //q为遍历指针，此时指向多项式二的第一项

Node\* pre = pa; //pre此刻指向多项式一的头指针，后续作为中间载体

Node\* u; //u指针做临时指针，用于释放节点

while (p != NULL && q != NULL)//若指针指向的内容都不为空

{

if (p->zhishu > q->zhishu)//若多项式一中的项系数大于对应多项式二中的项

{

pre = p;

p = p->Next;

}

else if (p->zhishu == q->zhishu)//若两项系数相等则合并同类项

{

float x = p->xishu + q->xishu;//x为合并后的系数

if (x != 0) //若合并后系数不为零

{

p->xishu = x; //将合并后的系数赋给多项式一中对应的项

pre = p; //pre指向p结点

}

else //若合并后系数为零

{

pre->Next = p->Next;//指向下一个结点

free(p); //释放p销毁结点

}

p = pre->Next;

u = q;

q = q->Next;

free(u);

}

else //若多项式一中的项系数小于对应多项式二中的项

{

u = q->Next;

q->Next = p;

pre->Next = q;

pre = q;

q = u;

}

}

if (q)

{

pre->Next = q;

}

free(pb);

}

//实现主函数

void main()

{

float xishu; //定义变量 系数和指数

int zhishu;

PNode head1 = (PNode)malloc(sizeof(struct Node));

PNode head2 = (PNode)malloc(sizeof(struct Node));

PNode tem = NULL;

head1->Next = NULL;

head2->Next = NULL;

//输入多项式一各项

printf("输入链表一的系数和指数，如：3,2 (以0,0结束输入)：\n");

scanf("%f,%d", &xishu, &zhishu);

while (xishu != 0) //当系数为零时停止输入

{

tem = (PNode)malloc(sizeof(struct Node)); //往链表中插入新的项

tem->xishu = xishu;

tem->zhishu = zhishu;

tem->Next = NULL;

insertNewPoint\_link(head1, tem); //插入新结点

scanf("%f,%d", &xishu, &zhishu); //输入下一项

}

printf("多项式一为：\n");

printLinkeLink(head1); //输出多项式一

printf("\n");

//输入多项式二各项

printf("\n输入链表二的系数和指数，如：3,2 (以0,0结束输入)：\n");

scanf("%f,%d", &xishu, &zhishu);

while (xishu != 0) //当系数为零时停止输入

{

tem = (PNode)malloc(sizeof(struct Node));

tem->xishu = xishu;

tem->zhishu = zhishu;

tem->Next = NULL;

insertNewPoint\_link(head2, tem);

scanf("%f,%d", &xishu, &zhishu);

}

printf("多项式二为：\n");

printLinkeLink(head2);

printf("\n");

//输出多项式相加的结果

add\_poly(head1, head2);

printf("\n多项式相加后的结果为：\n");

printLinkeLink(head1);

printf("\n\n");

}

实现思想：定义两个链表，将链表二中与链表一中相同的项相加，将链表一中没有的项插入进去，并按照指数继续降序排列。