数据结构作业

张笑辰

机1901

201930310107

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

typedef struct Node {

float xishu;

int zhishu;

struct Node\* Next;

}Node;

typedef struct Node\* PNode;

void insertNewPoint\_link(PNode head, PNode qNode)

{

PNode p = head;

PNode h = head;

PNode q;

while (p->Next != NULL)

{

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

{

qNode->Next = p->Next;

p->Next = qNode;

break;

}

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

{

float sum = p->Next->xishu + qNode->xishu;

if (sum != 0)

{

p->Next->xishu = sum;

}

else

{

q = p->Next;

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

free(q);

q = NULL;

}

break;

}

p = p->Next;

}

if (p->Next == NULL)

{

p->Next = qNode;

}

}

//输出多项式

void printLinkeLink(PNode head)

{

PNode temp = head->Next;

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

temp = temp->Next;

while (temp != NULL)

{

if (temp->xishu > 0)

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

else if (temp->xishu < 0)

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

temp = temp->Next;

}

}

void add\_poly(Node\* pa, Node\* pb)

{

Node\* p = pa->Next;

Node\* q = pb->Next;

Node\* pre = pa;

Node\* u;

while (p != NULL && q != NULL)

{

if (p->zhishu > q->zhishu)

{

pre = p;

p = p->Next;

}

else if (p->zhishu == q->zhishu)

{

float x = p->xishu + q->xishu;

if (x != 0)

{

p->xishu = x;

pre = p;

}

else

{

pre->Next = p->Next;

free(p);

}

p = pre->Next;

u = q;

q = q->Next;

free(u);

}

else

{

u = q->Next;

q->Next = p;

pre->Next = q;

pre = q;

q = u;

}

}

if (q)

{

pre->Next = q;

}

free(pb);

}

void main()

{

float xishu;

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;

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)

{

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

tem->xishu = xishu;

tem->zhishu = zhishu;

tem->Next = NULL;

insertNewPoint\_link(head1, tem);

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

}

printf("多项式一为：\n");

printLinkeLink(head1);

printf("\n");

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)

{

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");

add\_poly(head1, head2);

printf("\n多项式相加后的结果为：\n");

printLinkeLink(head1);

printf("\n\n");

}

实验总结：利用链表进行储存并遍历拿出来进行运算。首先比较Pa和Pb中的需要比较的qa和qb中指数的大小。如果qa的指数小，则会ha和qa指针都后移，继续比较。如果qa和qb的指数相等，则会把qa和qb的系数进行相加并赋值给qa的系数。从而实现相同指数的系数相加，加完后将qb所指结点删除，同时qa后qb指针都后移。如果qa的指数比qb大，则把qb所在的结点链接到qa的前面。qb指针继续后移。进行循环。