

实验报告

实验课程：数据结构

专业：计算机科学与技术

班级：16060104

姓名：杨旭

学号：16060104117

实验学时：2个

指导老师：潘煜

成绩：\_\_\_\_\_\_\_\_

2017年12月12日

XI`AN TECHNOLOGICAL UNIVERSITY

西安工业大学实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 专业 | 计算机科学与技术 | 班级 | 16060104 | 姓名 | 杨旭 | 学号 | 16060104117 |
| 实验课程 | 数据结构 | 指导老师 | 潘煜 | 实验日期 | 2017.12.4 | 同实验者 | 无 |
| 实验项目 | 一元多项式求和 | | | | | | |
| 实验地点 | 工1-408实验室 | | | | | | |

1. 实验目的

1. 复习并熟练掌握数据结构所使用的程序设计语言——C 语言 ；

2. 学会单步跟踪、调试自己的程序；

3. 加深对线性表特别是链表知识的理解和掌握，并能够运用相关知识来解决相关的具体问题，如一元多项式相加等；

4.对一元多项式能实现输入、输出，以及两个一元多项式相加及结果显示。

1. 实验原理

一元多项式的表示在计算机内用链表来实现，同时为了节省存储空间，只存储其中非零的项，链表中的每个节点存放多项式的系数非零项。它包含三个域，分别存放多项式的系数，指数，以及指向下一个项的指针。根据一元多项式相加的运算规则：对于两个一元多项式中所有指数相同的项，对应系数相加，若其和不为零，则构成“和多项式”中的一项，对于两个一元多项式中所有指数不相同的项，则分别复抄到“和多项式”中去。

1. 程序流程：

1.定义一元多项式链表结构体类型；

2.输入多项式项数以分配存储空间；

3.输入多项式每项的系数和指数，将其插入当前多项式链表。同时判断是否有与当前节点指数相同的项，若存在，则将两项系数相加合并。此外，若存在系数为0的项，将其存储空间释放；

4. 多项式各项分别求和；

注意：进行完一次运算以后，应该及时销毁无用多项式以释放空间以便再次应用。

1）定义一元多项式链表结构体类型

typedef struct polynode

{

int coef; //多项式的系数

int exp; //指数

struct polynode \*next;

}node;

1. 代码实现及运行结果

#include<stdio.h>

#include<stdlib.h>

typedef struct polynode

{

int coef; //多项式的系数

int exp; //指数

struct polynode \*next;

}node;

node \*create() //用尾插法建立一元多项式的链表

{

node \*h,\*r,\*s;

int c,e;

h=(node\*)malloc(sizeof(node));

r=h;

printf("输入系数coef:");

scanf("%d",&c);

printf("输入系数exp: ");

scanf("%d",&e);

while(c!=0) //输入系数为0时，多项式的输入结束

{

s=(node\*)malloc(sizeof(node));

s->coef=c;

s->exp=e;

r->next=s;

r=s;

printf("输入下一个系数coef:");

scanf("%d",&c);

printf("输入下一个指数exp: ");

scanf("%d",&e);

}

r->next=NULL;

return(h);

}

void print(node \*p) //输出函数，打印出一元多项式

{

while(p->next!=NULL)

{

p=p->next;

printf("%d\*x^%d ",p->coef,p->exp);

}

}

void polyadd(node \*ha, node \*hb)//一元多项式相加函数，用于将两个多项式相加，然后将和多项式存放在多项式ha中，并将多项式hb删除

{

node \*p,\*q,\*pre,\*temp;

int sum;

p=ha->next;

q=hb->next;

pre=ha;

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

{

if(p->exp<q->exp)

{

pre->next=p;

pre=pre->next;

p=p->next;

}

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

{

sum=p->coef+q->coef;

if(sum!=0)

{

p->coef=sum;

pre->next=p;pre=pre->next;p=p->next;

temp=q;q=q->next;free(temp);

}

else //如果系数和为零，则删除结点p与q，并将指针指向下一个结点

{

temp=p->next;free(p);p=temp;

temp=q->next;free(q);q=temp;

}

}

else

{

pre->next=q;

pre=pre->next;

q=q->next;

}

}

if(p!=NULL) //将多项式A中剩余的结点加入到和多项式中

pre->next=p;

else

pre->next=q;

}

void main()

{

node \*ha,\*hb;

printf("请输入第一个多项式ha的系数与指数:\n");

ha=create();

printf("第一个多项式是：") ;

print(ha);

printf("\n");

printf("请输入第二个多项式hb的系数与指数:\n");

hb=create();

printf("第二个多项式是；") ;

print(hb);

printf("\n");

printf("多项式的和是:\n");

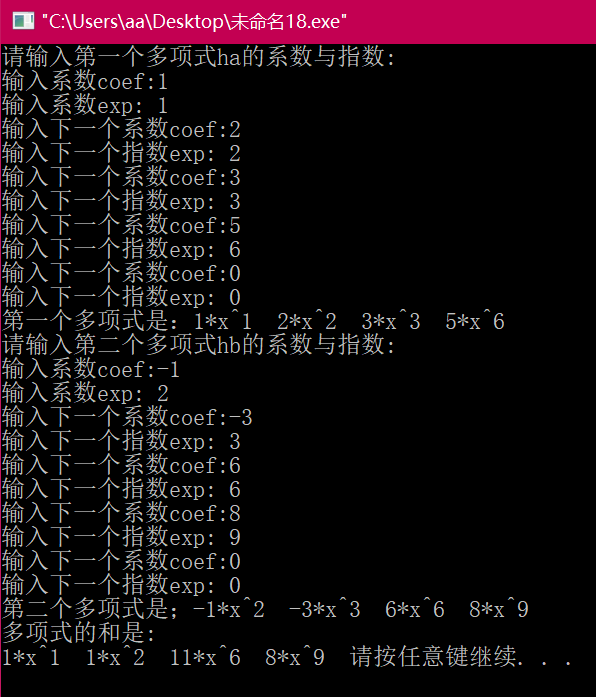
polyadd(ha,hb);

print(ha);

printf("\n");

}

运行结果：



1. 实验小结

通过这次的上机实验，是我对书本上的一些抽象的知识有了更加深刻地理解；本次实验使我对链式存储结构的应用有了更深刻的领会；