数据结构与算法综合应用选题

1. 医药物流配送网络优化（可能涉及图，最短路径算法等）

2. 门诊挂号模拟系统（可能涉及队列，检索，排序等）

3. 药品销售模拟系统（可能涉及栈，检索，排序等）

4. 医院网络布线建设优化（可能涉及图，最小生成树算法等）

5. 病人信查询与管理系统（可能涉及线性表，检索，排序等）

* 综合作业安排规则：

按组号MOD 5，得余数就是该组的题目，比如第1组第6组— “题1”，第2组第7组—“题2”。（如果上述题目不合适的，可以与老师协商后自选题目，要求涉及到树、图、排序及检索知识点相关的均可，应用场合要涉及医药相关）

作业上交要求：

1. 实验报告，模板见下面附件。
2. 实验代码，演示操作过程（视频3分钟内）
3. 汇报PPT及汇报视频（PPT总体美观，要有题目名称、组员姓名学号，要列汇报大纲，视频时间限制在5分钟）。

（一共上交5个文件）

以上作业相关资料以组号命名上交到学委汇总，分类整理之后再发到tandy@gzucm.edu.cn邮箱。

**一元多项式的表示及运算（实验报告）**

（格式要求：

实验标题黑体三号

内容标题：仿宋 小三

内容：仿宋 小四号,行距1.5）

班级：\*\*\*专业（）级

组长学号：2000\*\*\*\*\*\*\*\* 组长姓名：\*\*\*\*\*\*\*

完成日期： 年 月 日

|  |  |
| --- | --- |
| 姓名 | 分工 |
| \*\* | 总体设计，需求分析 |
| \*\* | 算法设计，流程图绘制 |
| \*\* | 编写代码 |
| \*\* | 系统测试 |
| \*\* | 文档编写 |

一.问题的需求分析（备注：对于复杂一些的问题，需要详细描述）

1.问题描述

编制一个程序，实现两个一元多项式的相加、相减、相乘。

2.需求分析

（1）建立一元多项式

（2）计算出两个一元多项式的和、差、积。

（3）一元多项式运算结果输出

二.抽象数据类型的设计

ADT 一元多项式的表示及运算

ADT Poly is

operations

PolyNomail createPoly()

PolyNomail add(PolyNomail ahead,PolyNomail bhead)；相加

PolyNomail sub(PolyNomail ahead,PolyNomail bhead)；相减

PolyNomail mult(PolyNomail ahead,PolyNomail bhead)；相乘

Void output(PolyNomail dpoly);

end ADT Poly

三.算法与数据结构的设计

1 数据结构设计

在对上述问题的数据进行抽象之后,确定应用以下存储结构:

typedef struct term\* PolyNomail;

typedef struct term

{

double coef;//变量代表的意义需要加注释

int expn;

struct term \*next;

}PolyNode;

进行运算的函数：

PolyNomail add(PolyNomail ahead,PolyNomail bhead)；相加

PolyNomail sub(PolyNomail ahead,PolyNomail bhead)；相减

PolyNomail mult(PolyNomail ahead,PolyNomail bhead)；相乘

2 算法设计

（下面以多项式相加为例）

一元多项式的相加算法图示

**ah**

-1

7 0

3 1

9 8

5 17

∧

**bh**

-1

8 1

22 7

-9 8

∧

22 7

-1

7 0

11 1

5 17

∧

**ah**

一元多项式加法算法主要步骤

分别对两个链表ah 、bh进行扫描，设p 、q 分别指向线性链表ah 、 bh的当前进行比较的某个结点：

p-> expn < q-> expn ： p 所指结点应为和多项式中的结点

p-> expn = q-> expn ：将p所指结点的系数“加”到q所指结 点的系数上相加；

p-> expn > q-> expn x ： 从表bh中删除q 所指结点，并将其 插入到ah表p所指结点之前；

四．算法的精化与程序的实现（程序实现的核心算法）

（一）核心算法一

* 流程图
* 算法实现代码

（二）核心算法二

* 流程图
* 算法实现代码

createPoly

从终端输入数据建立多项式链表：

PolyNomail createPoly()

{

PolyNode \*head=0;

int expn;

double coef;

head=(PolyNode\*)malloc(sizeof(PolyNode));

if(head==0)

return 0;

head->coef=0.0;

head->expn=0;

head->next=0;

do

{

printf("Input Coef and Expn (0 0):");

//输入系数及指数，以输入0 0结束

scanf("%lf%d",&coef,&expn);

if((long)coef==0&&expn==0)

break;

insert\_node(head,coef,expn);

}while(1);

return head;

}

//add实现多项式链表相加：

PolyNomail add(PolyNomail ahead,PolyNomail bhead)

{

PolyNode \*head=0;

PolyNode \*ap,\*bp;

int expn=0; double coef=0.0;

ap=ahead->next; bp=bhead->next;

head=(PolyNode\*)malloc(sizeof(PolyNode));

if(head==0)

return 0;

head->coef=0.0; head->expn=0; head->next=0;

while(ap)

{

expn=ap->expn; coef=ap->coef;

insert\_node(head,coef,expn);

ap=ap->next;

}

while(bp)

{

expn=bp->expn; coef=bp->coef;

insert\_node(head,coef,expn);

bp=bp->next;

}

return head;

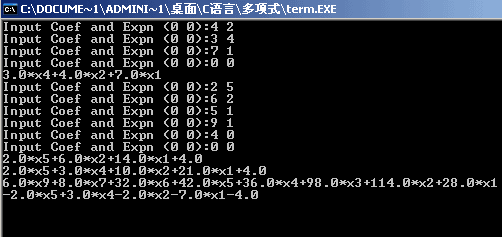
}

//sub实现多项式链表相减：

……

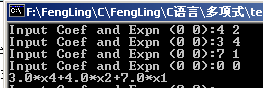
五．程序的调试与计算的结果**分析**

用户进入程序之后就输入第一个多项式的每一个项，并以（0，0）结束，之后输入第二个多项式，这样就能够计算出相加、相减、相乘的结果。



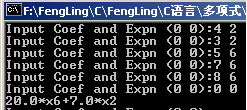
1.调试情况

（1）调试能否正确输入与显示多项式链表，结果如下：



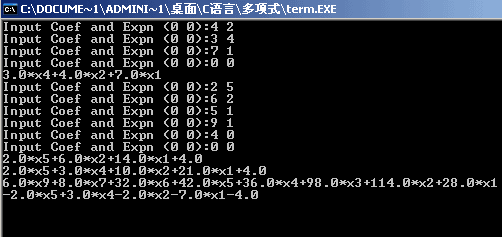
这样就能够证明能够按顺序输入每一个项之后也能够按顺序存储多项式链表

（2）调试输入相同次数的项能否自动合并同一个项，结果如下：



输入2次数的项有2个，6次数的项有3个，结果是存储了2个项

（3）计算两个多项式链表相加、相乘、相减，结果如下：



2.结果分析

（先把调试过程中出现的结果都分析一下）然后下结论：从调试情况看，本程序实现了两个一元多项式的相加、相减、相乘。（如果程序还没有实现相应的功能，就不能下结论）

六．时间代价的分析

本程序所有的函都要用到insert\_node函数，这个函数是实现插入一个数，会使这个数自动在相应大小的位置插入的。要实现这个功能必须每一次插入一个数据，平均运行次数是n/2(n是当前总个数)，所以要插入n个数据，平均运行次就是n\*(n+1)/4，时间复杂度为O(n)。而在相加函数中，平均运行次数是n\*(n+1)/2;在相乘函数中，平均次数是M\*（M+1）/4，其中M=n\*m,n是第一个多项式链表的项数，m是第二个多项式链表的项数;相减也是一样。

七、有待改进的问题，收获和体会

1、有待改进的问题

程序的容错性及健壮性需进一步加强（自己程序中还出现哪些问题都应写上）

2、收获和体会

本程序采用了结构化的做法，有一定的代表性和条理性；其中主要数据类型采用了最简单的单向链表，简明而有效。上机实验对动手能力的一个训练，平时看一下题目觉得自己会做，但没有动手，就难以学到一些非常基础的东西了。在编译的过程中，会发现许多被忽略的细节，这样不但可以巩固书本学来的知识，还可以让自己充分理解学习中的难点，获得编程的乐趣。特别是在编译链表的过程中，要是稍不注意，就会造成链表的脱节，这些可以训练自己的认真和谨慎。况且里面有好多思维逻辑方面的东西，要是思维一乱就不知道自己在做什么了。但即使多方面的思考，在编译过程中也难免会出错，因此改错的过程对自己来说又是一个挑战。由于本程序是用C++编成的，因此拥有了面向对象的功能，使其有很大的扩展空间。

最后，感谢谭老师给我们这样一个好的实践机会来体会程序设计中的种种问题，我希望在以后的深入学习中，这种机会将更多。