**专题一：结构化程序设计与递归函数**

问题1： 多项式算数

问题2： 共享子串问题

组号：21 教师：张引、田沈晶、熊海辉

成员：

学号 3170105860 姓名 彭子帆

学号 3160103848 姓名 陈稼诚

**报告提交时间：2018.5.6**

# 1.问题1名称（完成的其他问题也按此结构模板撰写）

## 1.1 问题描述

一元多项式 ：

## 主要运算：多项式相加、相减、相乘等采用链表结构来存储多项式的非零项。每个链表结点存储多项式中的一个非零项，包括系数和指数两个数据域以及一个指针域应让使用则从控制台输入多项式，输入格式为每行两个数字，第一个数字是系数，第二个数字是幂

## 例如： 1 4 表示 注意两个输入的数字之间以空格隔开用-1 -1来表示输入终止（只考虑x的正整数次幂）

## 1.2 问题解决思路和关键点

### 1.2.1模块化

将几个不同功能单独编写，将加法、减法、乘法、排序、合并同类项的函数分别编写，然后根据不同需要调用不用函数。

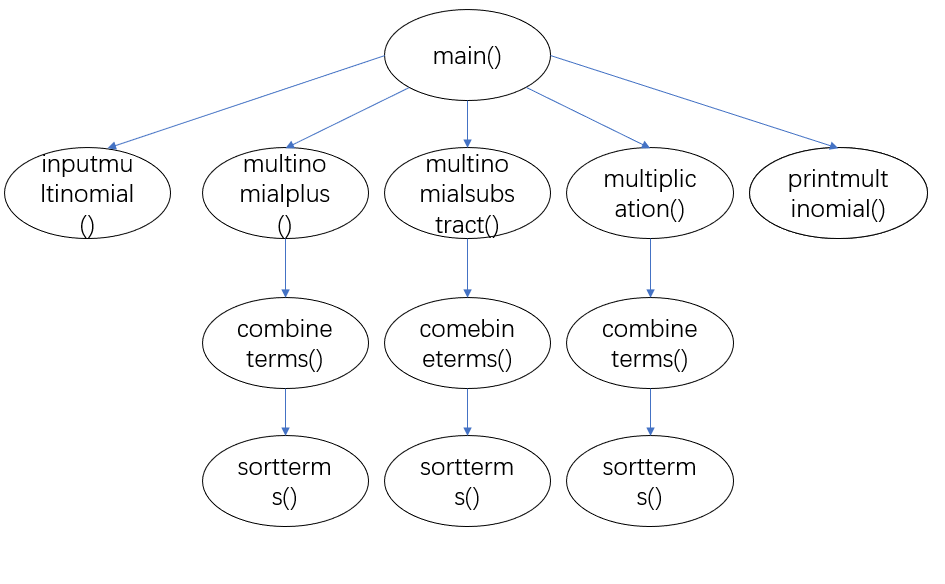
### 1.2.2 排序

链表排序不像数组排序简单易操作，数组排序可以用选择冒泡等等一系列。由于链表的复杂性，经过思考后，链表我们采用了冒泡排序，若需交换，则将相邻的单元里的所有值交换即可。

## 1.3 程序结构

### 1.3.1 函数调用关系图

画出main函数及其下属各层调用的函数



### 1.3.2 函数说明

1）函数原型：struct terms \*Inputmultinomial(void)

功能描述；输入多项式

参数描述： void

返回值描述：返回一个结构指针

重要局部变量定义：head,tail,p

重要局部变量用途描述：记录链表头，尾和临时指针变量申请空间

函数算法描述：while循环一个一个输入系数和指数，直到同时为-1

2）函数原型：struct terms \*multinomialplus(struct terms \*multihead1,struct terms \*multihead2)

功能描述；两函数相加

参数描述： 两个多项式的头结构指针

返回值描述：返回两多项式相加后的新的头结构指针

重要局部变量定义：临时结构指针

重要局部变量用途描述：用于连接两个链表

函数算法描述：将两个多项式相连，尔后合并同类项即可

3）函数原型:struct terms \*multinomialsubtract(struct terms \*head1,struct terms \*head2)

功能描述；两多项式相减

参数描述： 两多项式的头结构指针

返回值描述：结果的结构指针

重要局部变量定义：临时结构指针

重要局部变量用途描述：连接两个链表，并且将第二个多项式的系数变为原来的相反数

函数算法描述：将第二个系数变为原来的相反数，而后将两个多项式相连，尔后合并同类项即可

4）函数原型：struct terms \*multiplication(struct terms \*head1,struct terms \*head2)

功能描述；两多项式相乘

参数描述：两个多项式链表的头结构指针

返回值描述：结果的头结构指针

重要局部变量定义：三个临时指针变量ptr1,ptr2,p，新的结构指针的头head和尾tail

重要局部变量用途描述：循环使其相乘，p申请空间，head和tail记录新链表

函数算法描述：for循环记录所有的相乘结果而后调用combineterms合并同类项

5）函数原型：struct terms \*sortterms(struct terms \*head)

功能描述；排序，把多项式x幂从小到大排序

参数描述： 需要排序的多项式的头结构指针

返回值描述：排序好的多项式的头结构指针

重要局部变量定义：临时结构指针ptr1,ptr2,p

重要局部变量用途描述：用于循环遍历所有并使用冒泡排序交换左右值

函数算法描述：链表的冒泡排序

6）函数原型：struct terms \*combineterms(struct terms \*head)

功能描述；合并同类项并去除系数为零的项

参数描述： 所需合并的多项式的头结构指针

返回值描述：合并同类项并去除零项后的链表头结构指针

重要局部变量定义：ptr1,ptr2,p

重要局部变量用途描述：临时结构变量用于合并同类项和去除零项

函数算法描述：先排序，for循环寻找两个相同的项，其合并而后去除新的系数里的零项

7）函数原型：void printmultinomial(struct trems \*head)

功能描述；打印多项式

参数描述： ptr

返回值描述：无

重要局部变量定义：ptr

重要局部变量用途描述：输出各项系数和指数

函数算法描述：for循环打印

### 1.3.3 程序文件结构

1）文件函数结构

Sortterms.c 包含sortterms函数

combineterms.c 包含comebineterms函数

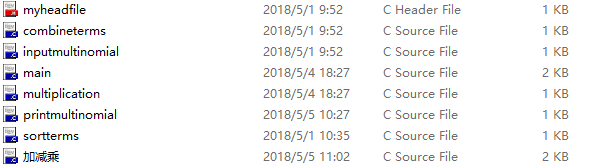
printmultinomial.c 包含printmultinomial函数

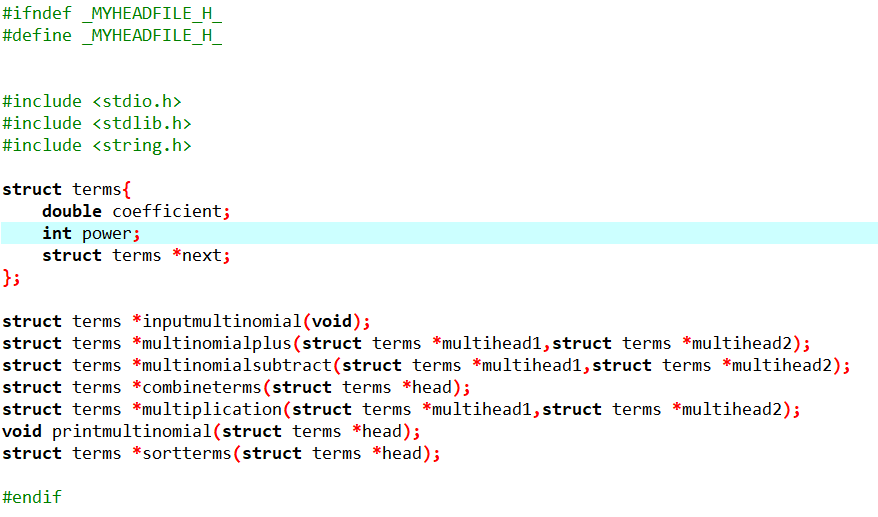
加减乘.c 包含multinomialplus、multinomialsubtract、multiplication函数

main.c

myheadfile.h 包含stdio.h,stdlib.h，结构体terms的定义，以及函数声明

2）多文件构成机制





## 1.4 安装运行说明

直接运行exe文件，直接编译运行，无需安装

## 1.5 总结

### 1.5.1 实践过程中遇到的难点及解决方案

链表的冒泡排序以及合并同类项时遇到的越界问题，通过自学以及不断的debug最终成功

### 1.5.2 程序亮点或基本功能之外的拓展点

### 1.5.3 实践之不足

还行，实践挺足的吧

### 1.5.4 分工合作和各自的收获

我们写了不同的函数，收获了很多，对于链表的操作更加明了了。

# 2.共享子串

## 2.1 问题描述

## 英语单词可以用链表的形式存储。为了节省空间，可以将两个以相同子串结尾的英语单词，共享同一段链表空间。比如“loading”和“being”。

## 2.2 问题解决思路和关键点

### 2.2.1如何做到两个链表串联在一起？

先找出两个单词公共的部分。分别用三个链表存取两个单词不同的部分以及共享的部分。最后再拼接在一起。

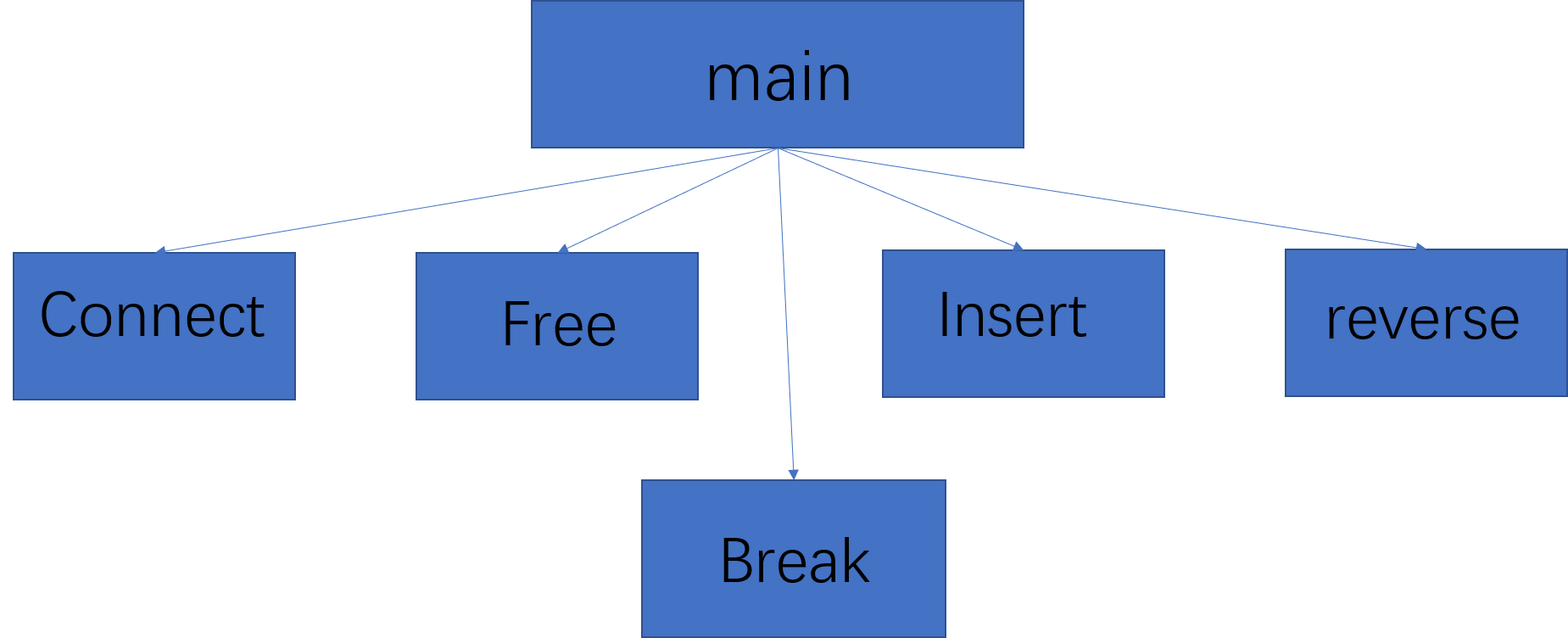
### 2.2.2如何找出两个单词的共享部分

将输入的两个单词的字符串倒置。一个一个对比字符是否相同。

## 2.3 程序结构

### 2.3.1 函数调用关系图

画出main函数及其下属各层调用的函数



### 2.3.2结构类型

在程序中，节点类型被定义为

typedef struct character{

char ch;

int tag;

struct character \*next;

}List;

### 2.3.3 函数说明

1）函数原型：List \*Insert(char c, int t, int i, List \*Ptrl);

功能描述：在链表中插入一个元素。

参数描述： 字符c，整数int，整型t，链表头Ptrl

返回值描述：本函数返回一个新的链表头节点。

重要局部变量定义：指针p；指针s；

重要局部变量用途描述：用于寻找第i-1个节点以方便插入节点的操作。

函数算法描述：首先判断插入位置是否为头结点。如果是即插入头结点。在本程序中只进行了插入头节点的操作。

2）函数原型： void reverse(char c[])

功能描述： 翻转字符串

参数描述： 字符数组c【】

返回值描述：本函数无返回值

重要局部变量定义：整型length，整型i

重要局部变量用途描述：length用于记录字符串长度，i用于循环。

函数算法描述：将数组的前半部分每个数与后半部分相加。再通过两次相减实现数据的交换。

3）函数原型：void Connect(List \*head, List \*tail)

功能描述： 拼接两个链表

参数描述：head为要放在前段的链表，tail表示放在后端的链表。

返回值描述：本函数无返回值。

重要局部变量定义：指针p

函数算法描述：首先判断head是否为空。若为空则返回tail。若不为空则利用p来寻找head中指向空指针的链表结。将其指向tail的首节；返回新的head链表首节点。

4）函数原型：void Free(List \*p)

功能描述：释放先前malloc申请的空间。

参数描述：需要释放的链表头指针。

返回值描述：无意义

重要局部变量定义：指针p；

函数算法描述：释放空间知道链表指向空指针。

5）函数原型：void Break(List \*head, List \*Breakpoint)

功能描述：将一个链表在指定处截断

参数描述：链表头指针，断点。

返回值描述：无意义

重要局部变量定义：指针p；

函数算法描述：释放空间知道链表指向空指针。

### 2.3.4 程序文件结构

1）文件函数结构

文件分为共享子串.c, function.c, function.h.

function.c中:

List \*FindKth(int K, List \*PtrL);

List \*Insert(char c, int t, int i, List \*Ptrl)

void Connect(List \*head, List \*tail);

void reverse(char c[]);

void Free(List \*p);

void Break(List \*head, List \*Breakpoint)

Function.h中包含函数原型。

2）多文件构成机制

在function,h中用ifndef防止重复定义

## 2.4 安装运行说明

运行时现将main.c、function.c文件编译为.o文件，再将.o文件以及头文件串联起来生成可执行的exe文件.

## 2.5 总结

### 2.5.1 实践过程中遇到的难点及解决方案

Free函数在释放空间时遇到程序崩溃对的情况。经过检查发现是因为两个链表衔接到同一个链表。第一个链表释放玩空间后并未将指针变为空指针。导致第二次释放时会尝试free掉一个已经被释放空间的指针，导致程序崩溃。后在释放空前之前先把链表拆分为两个独立的链表就可以顺利释放空间。

### 2.5.2 程序亮点或基本功能之外的拓展点

无。

### 2.5.3 实践之不足

程序功能较为单一。没有附加的功能。

### 2.5.4 分工合作和各自的收获

我们交流了各自的代码，在撰写代码的过程中获得了乐趣。

# n. 感想