

高级语言程序设计

课程设计报告

学 院 班 级

姓 名 学 号

成 绩 指导老师

年 月 日

**课程设计答辩记录表**

|  |  |
| --- | --- |
| **课程设计题目及要求** | |
| **多项式的合并运算：**  **要求：**  键盘输入多项式如下所示：  7X^500+9X^100+2X^3+2X^2+100 与  3X^3000+35X^2000+18X^100-2X^3+1000  输出合并的结果：  3X^3000 +35X^2000 +7X^500 +27X^100 +2X^2 +1100  注意：输入多项式时，并不一定按照幂降次的次序排列，并且有可能出现输入格式错误，程序应能检查其输入格式是否正确。  【要求】使用链表结构实现多项式的存储。 | |
| **完成情况：（答辩时由老师填写如下内容）** | |
| 基本数据结构设计 |  |
| 算法设计及流程图 |  |
| 输入输出设计 |  |
| 代码规范及注释 |  |
| 数据测试及调试 |  |
| 代码理解与说明 |  |
| 评语 |  |
| 综合评分 |  |

**目 录**

[一、功能模块结构 1](file:///C:\Users\15421\Desktop\C语言课程设计报告(示例).doc#_Toc25142)

[二、数据结构设计及用法说明](file:///C:\Users\15421\Desktop\C语言课程设计报告(示例).doc#_Toc8965) 3

[三、程序结构（流程图） 4](file:///C:\Users\15421\Desktop\C语言课程设计报告(示例).doc#_Toc22821)

[四、各模块的功能 8](file:///C:\Users\15421\Desktop\C语言课程设计报告(示例).doc#_Toc14026)

[五、试验结果（包括输入数据和输出结果）](file:///C:\Users\15421\Desktop\C语言课程设计报告(示例).doc#_Toc8384) 9

[六、体会 1](file:///C:\Users\15421\Desktop\C语言课程设计报告(示例).doc#_Toc4448)2

[七、参考文献 14](file:///C:\Users\15421\Desktop\C语言课程设计报告(示例).doc#_Toc20069)

# 一、功能模块结构

多项式合并系统由五大功能模块组成：输入模块，链表链接，输出模块，提示菜单，释放空间。其中，输入模块包括对输入的内容进行提取系数与指数，对格式错误进行判断等；链表链接模块包括对链表的创建，对相同指数的节点进行合并等；输出模块包括对常数项、一次项、系数为一的项进行不同的输出格式的选择。

具体如下图所示：

多项式合并系统

输入模块

链表链接

输出模块

提取系数

提示菜单

释放空间

提取指数

提示错误

节点合并

链表创建

选择格式

输出

释放空间

帮助系统

图1 输入模块示意图

下图为具体功能模块：

输入

模块

系数

指

数

其他字符

整数部分

状态

小

数

部

分

整

数

部

分

状态

小数部分

判定为错误

图2 数据维护模块功能示意图

链表链接

不同

相同

找到大小位置插入

找到相同节点相加

图3 数据查询模块功能示意图

# 二、数据结构设计及用法说明

**多项式合并系统**

对多项式进行合并，主要包括读取多项式，合并多项式，存储多项式。

高校基本信息参考：

表1 多项式基本信息

|  |  |  |
| --- | --- | --- |
| **中文字段名** | **类型及长度** | **举例** |
| 系数 | double | 1 ; 100 ; 1.05； |
| 指数 | double | 1 ; 100 ; 1.05； |
| 输入字符 | char | ‘0’~‘9’ ； ‘x’ ； ‘X’ ； ‘^’ ； ‘ ‘ ；’\*‘ ； |

招生信息管理系统数据结构用法：该系统的数据结构是采用链表。

链表的模型如下：

系数1

指数1

系数2

指数2

系数n

指数n

NULL

head

**…**

图4 多项式合并系统链表的模型

链表说明：以从指数大到小的方式创建链表，删除功能一个个删除节点，输出功能遍历链表输出。

# 三、程序结构（流程图）

1、主函数流程图

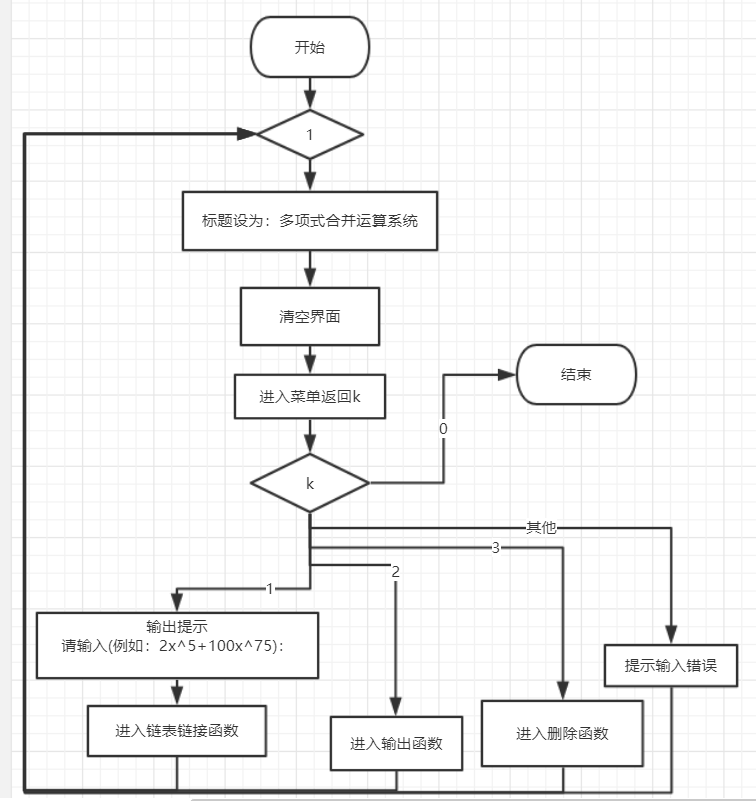


图5 主函数流程图

2、菜单流程图（提示）

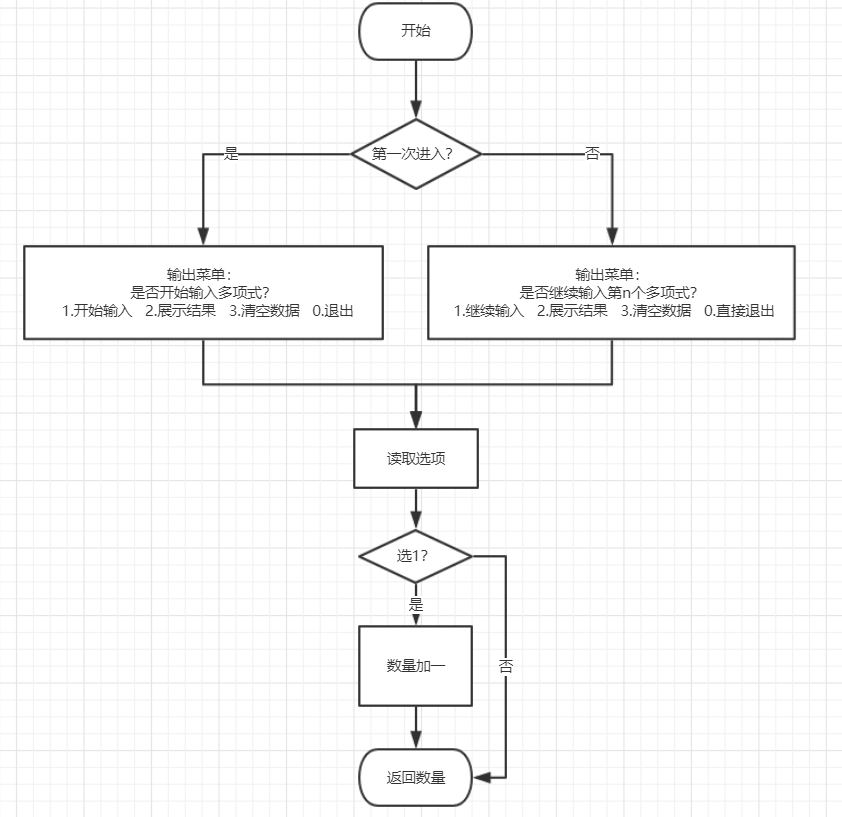


图6 菜单流程图

3、链表链接流程图

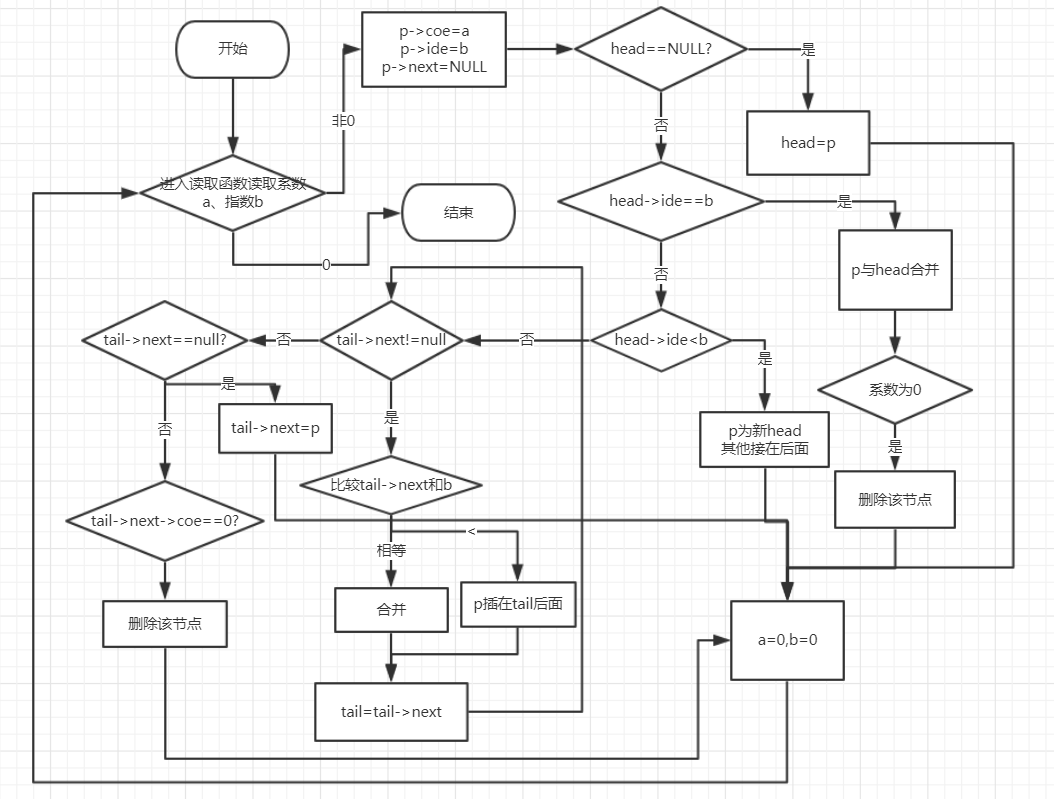


图7 链表链接流程图

4、输入模块流程图

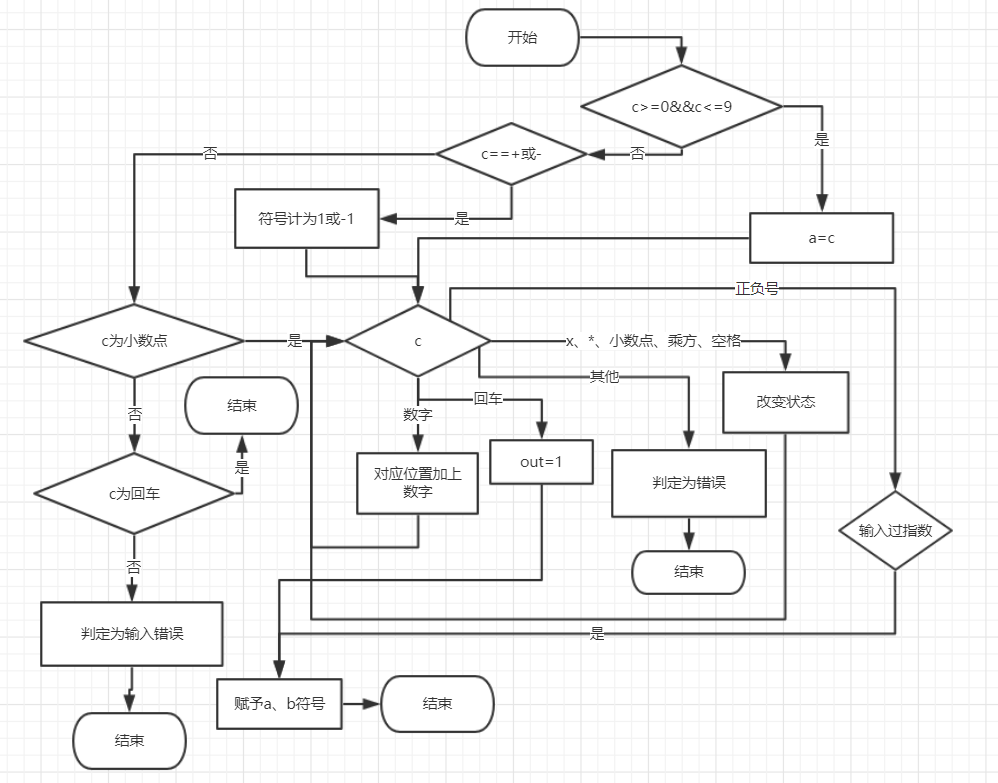


图8 输入模块流程图

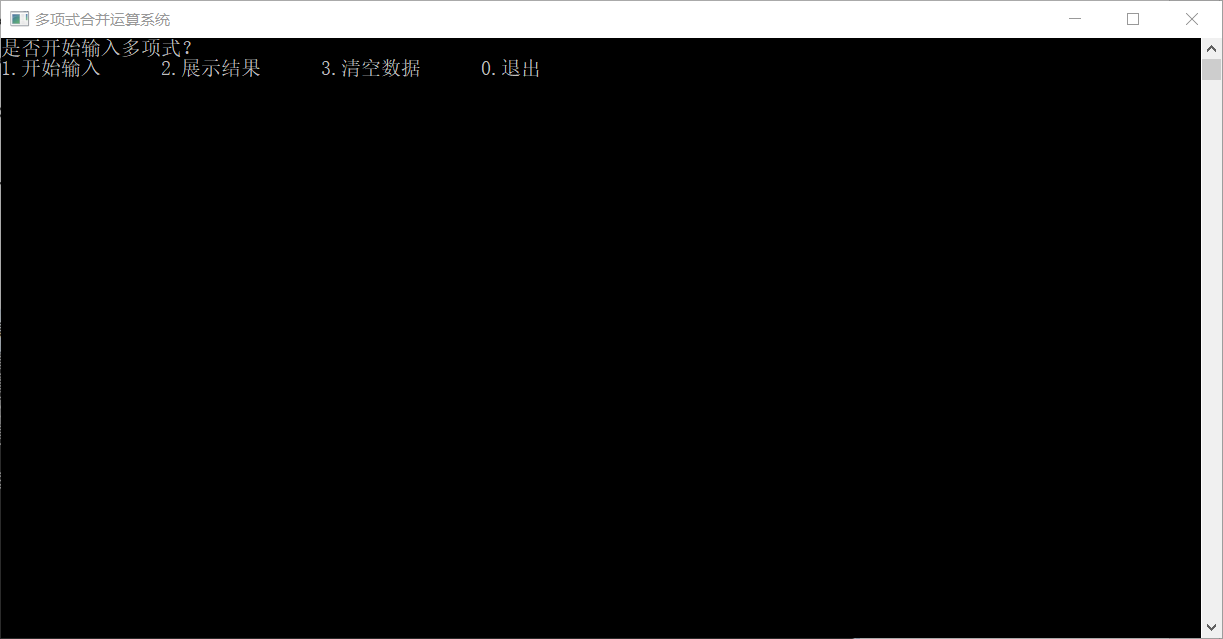
# 四、各模块的功能

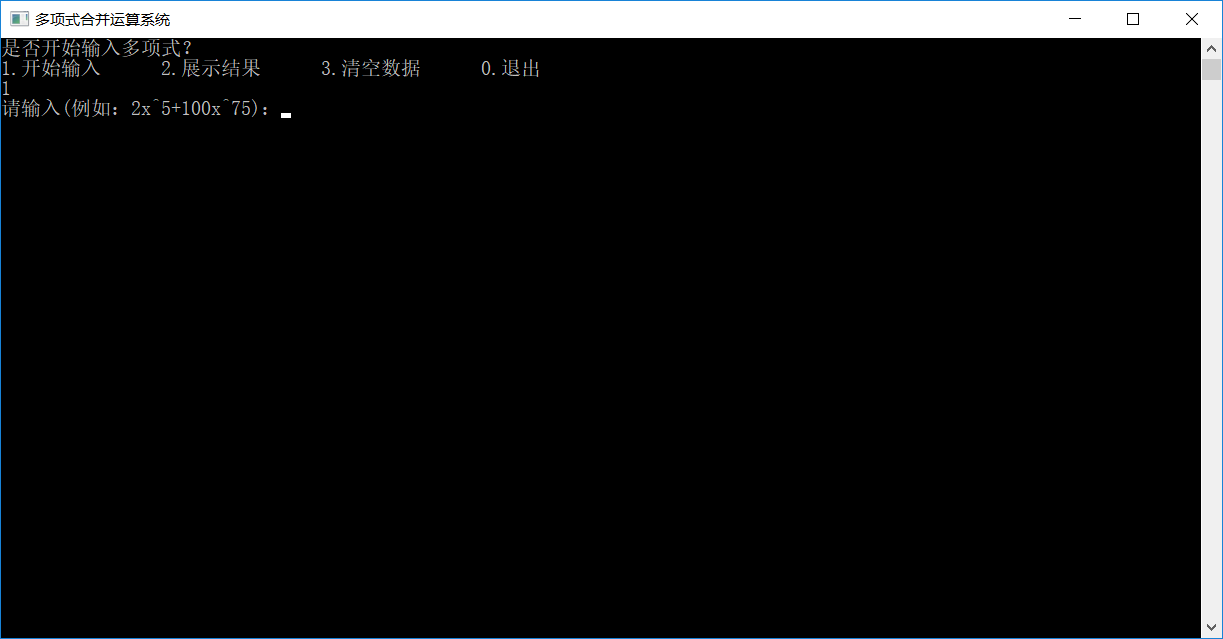
1. menu ：显示菜单，计数，选择分支。
2. add：添加多项式，创建链表，合并节点。
3. printall：输出每个节点。
4. correction：输入字符，提取系数、指数，判断格式错误。
5. deleteall：释放链表空间。

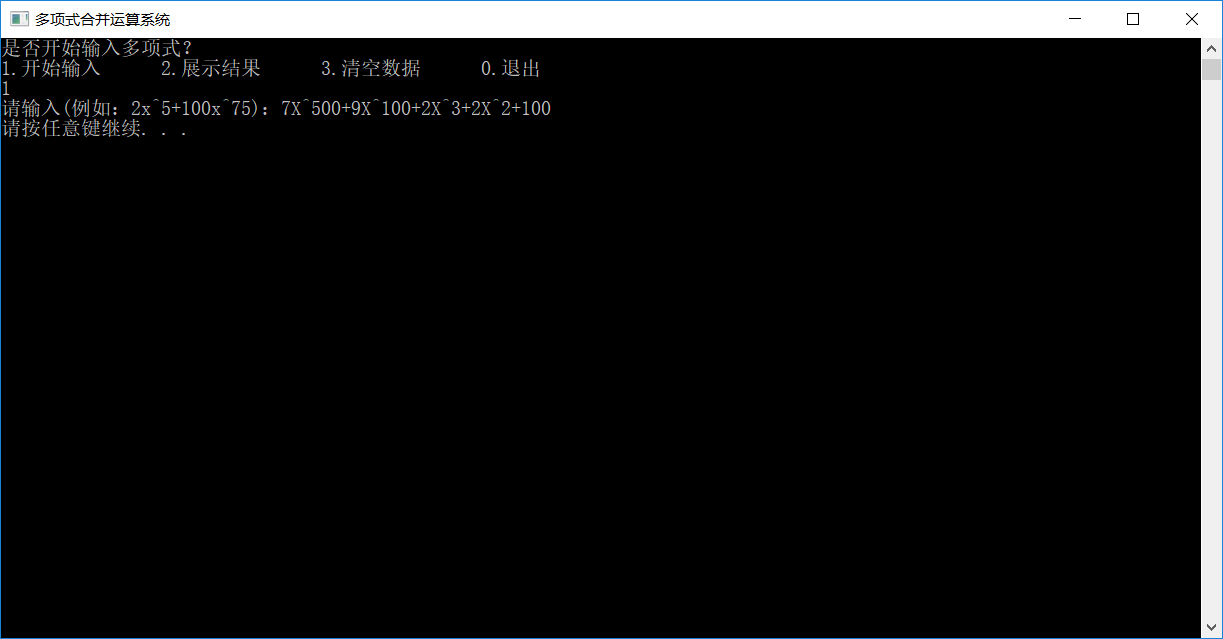
# 五、试验结果（包括输入数据和输出结果）

输入数据：

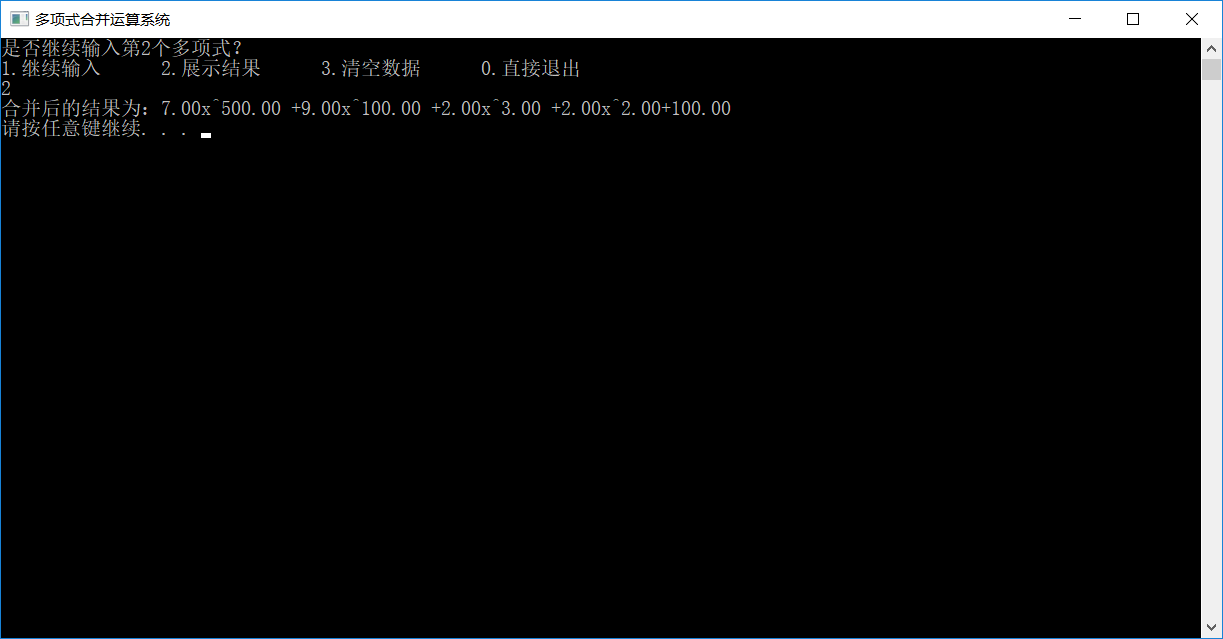
1. ：7X^500+9X^100+2X^3+2X^2+100
2. ：3X^3000+35X^2000+18X^100-2X^3+1000
3. ：2\*x5-36x67+5.5x54
4. 系统主界面



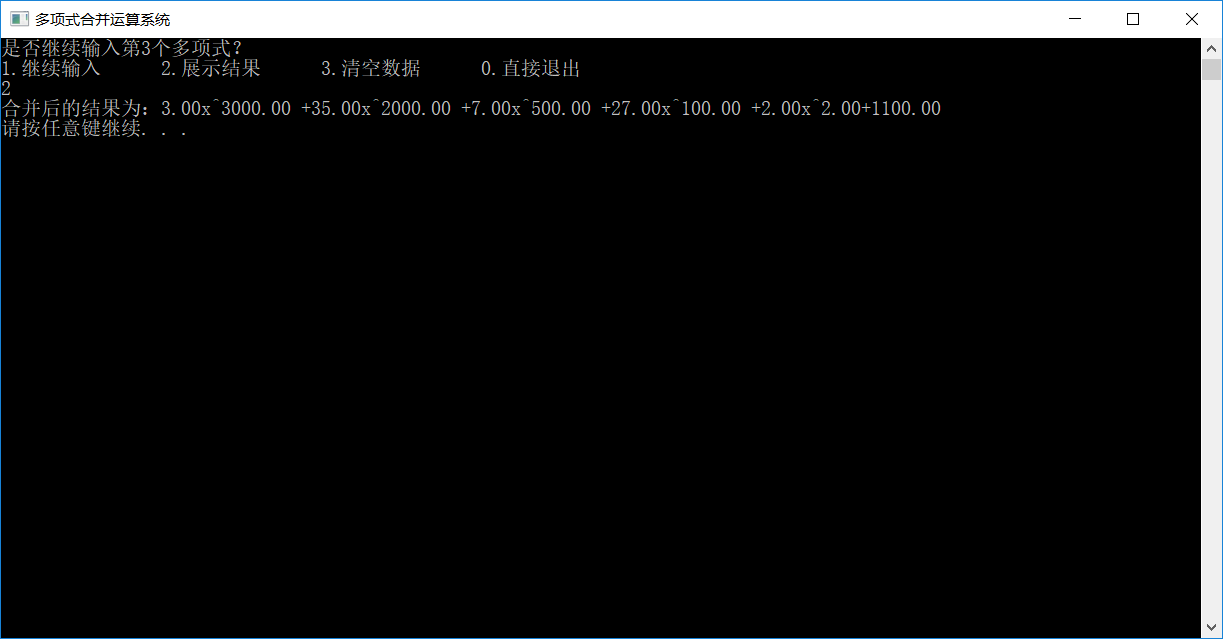
1. 输入界面
2. 输入后



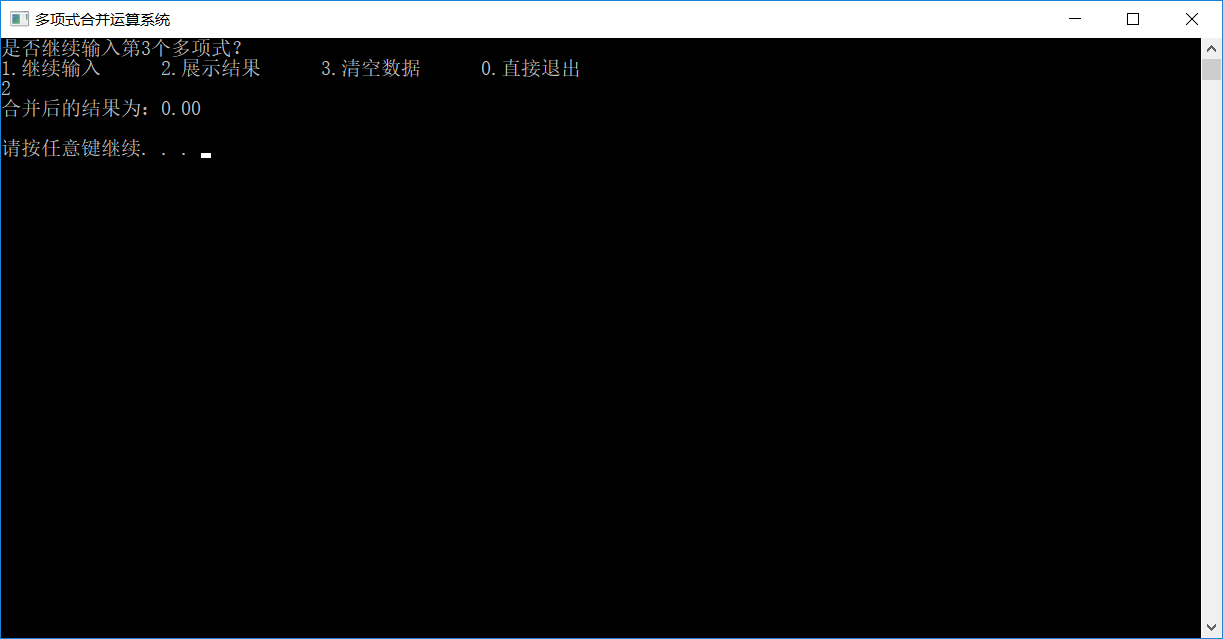
1. 展示结果



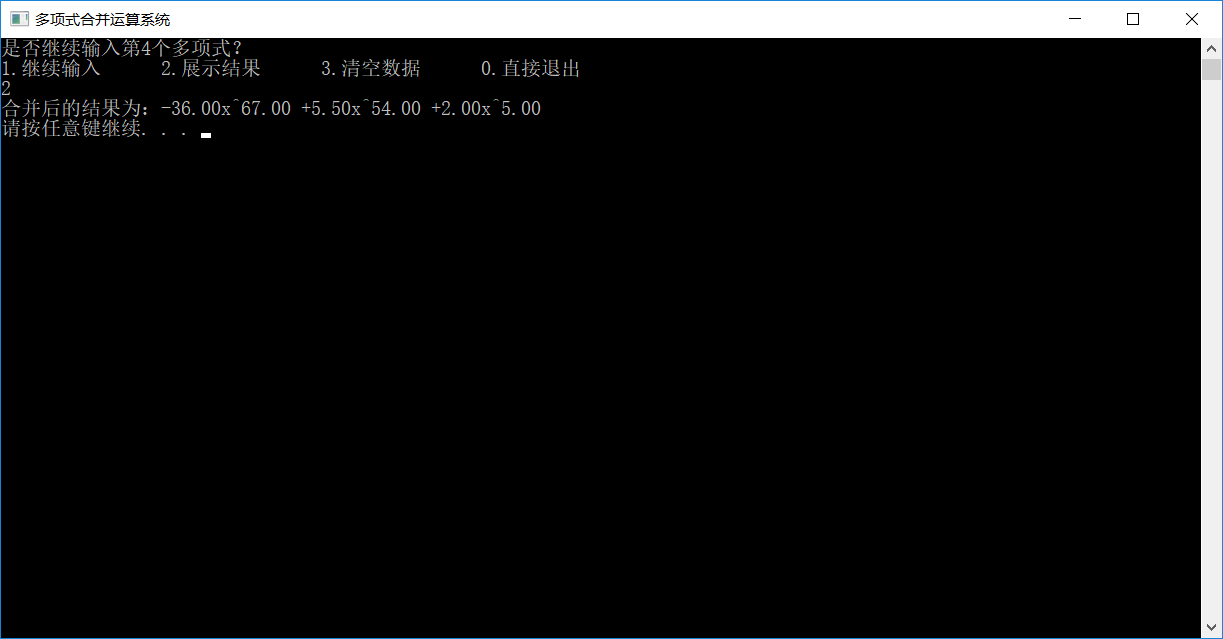
1. 继续输入后结果



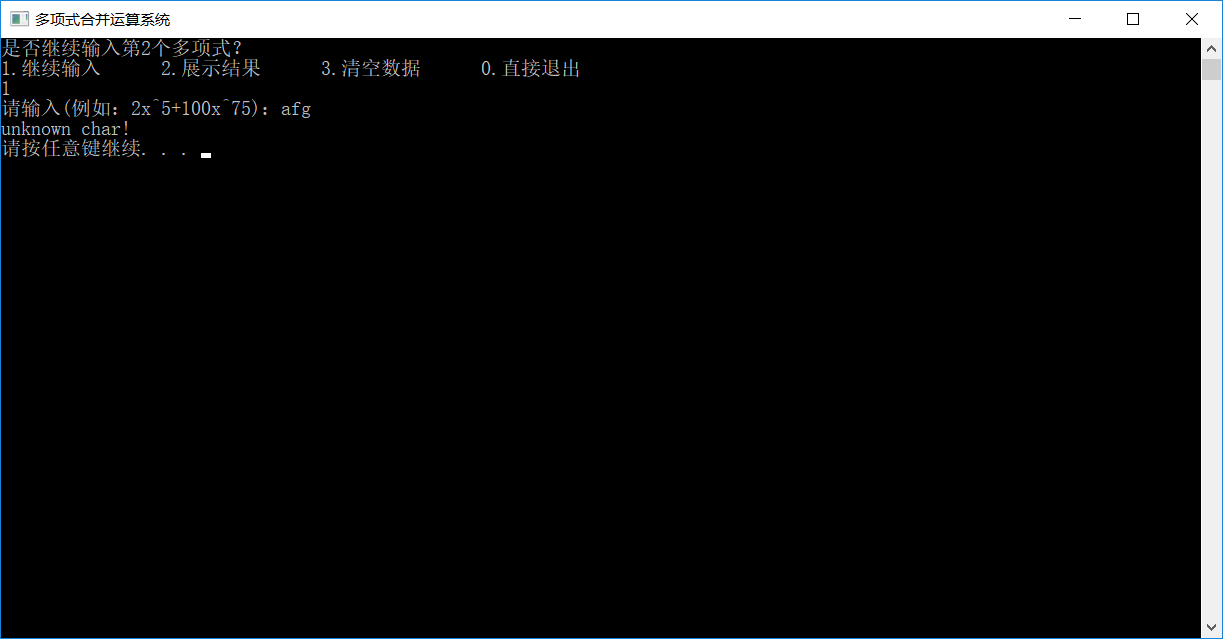
1. 清空数据后结果

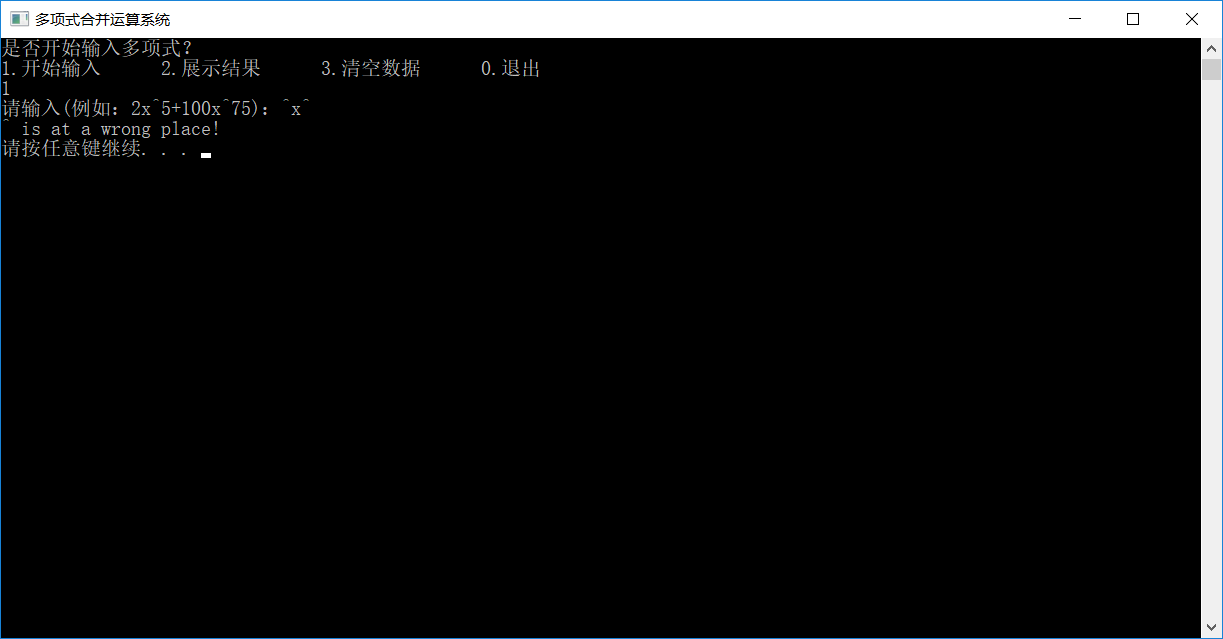


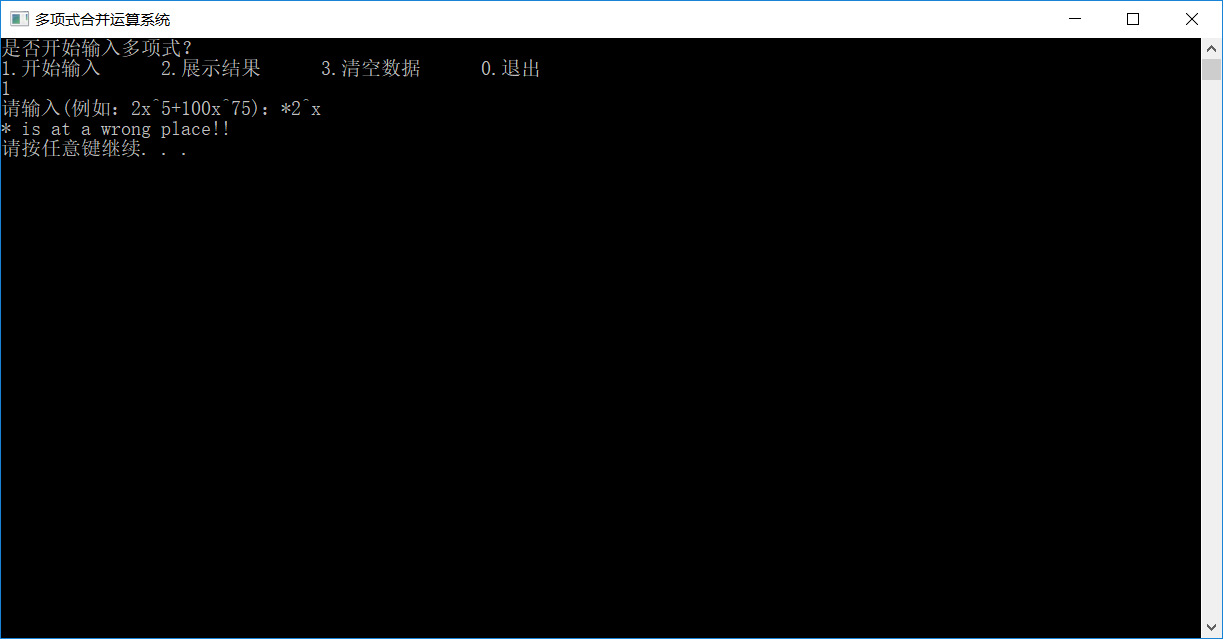
1. 继续输入后结果



1. 有错误字符







六、体 会

刚开始的时候，课设是一个新的概念，对于从未接触过的东西，人总有无尽的欲望去探索、发掘。后来，大致了解到，课设就是课堂所学知识的具现化。通过课设，我们能更好的消化知识，使技术不仅仅局限在理论层面。通过动手，能深刻理解计算机的思想。课设题目也许不是最难的，但却是最繁琐的。

题目的一些要求，例如使用链表，就是一个消化吸收的过程。课设基础中的基础就是链表，但运用链表并不是一个简单的活。也许是自己学的还不精，亦或是平常题量不足，导致课设的进程缓慢。于是乎，在“笨鸟先飞”敲打下，我提前学习了数据结构中的部分内容，对链表有了更深刻的了解。然后又重复研究了视频里的代码，初步掌握了链表的运用。在电脑上稍加练习后，也对如何应用链表有了一些领悟。而后就开始着手课设了。

主函数写起来非常简单，而后又把菜单单独剥离使程序更有条理性。然后就开始了链表的工作。链表节点的传入传出，链表的插入，释放一一完成后，一句“能找出格式错误”困扰到了我。于是乎，从简单的读入两个数变成了改错的过程。同样，我也把它单独拿出来，写了一个函数。这是最困难的一个函数，涉及到了输入的类型，输入的位置等。刚开始我是直接读入数字，后来发现不可行，因为位置的原因，还有系数为1能省略的情况。选择用字符类型读入，无疑增加了许多工作量。在临奔溃的边缘，终于把能想到的情况考虑了个遍。刚开始会有很多错误，例如赋值的左右不对等 等。终于都一一克服了之后，就开始了最困难的工作，测试修改。也许有一些小小的逻辑错误是暂时无法识别的。通过上百次的测试，错误终于减小到了一定的范围内。

做课设真的是一个学习的过程，C语言方面的进步还是次要的，主要是对毅力的磨练，我这种急性子都快疯了。第一次依靠自己的力量写出了这么长的代码并且成功运行，成就感还是有的。当然，最重要的收获当然是，我能耐下性子，去解决一些有些挑战的难题，能独立解决问题。希望在以后的生活中，我能有这种百折不挠，愈挫愈勇的精神。面对困难永不言弃，坦然面对自己的错误并改正它。

# 七、参考文献

**[1]**《C语言课程设计》 何钦铭，颜 晖 编著 高等教育出版社

**[2]**《C语言设计语言》 Brian W. Kernighan/Dennis M. Ritchie编著

徐宝文/李志译 译 尤晋元 审校 机械工业出版社