**实验1 线性表的应用：稀疏一元多项式运算器**

PB17050941 李喆昊

1. 问题描述

题目：稀疏一元多项式运算器

基本功能：

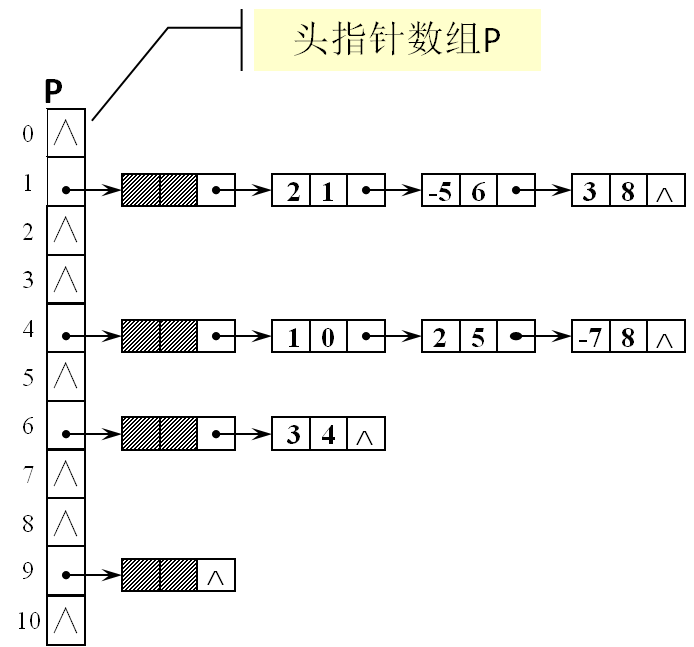
1. 输入并创建多项式

多项式各项按指数升序或降序排列，系数为浮点型，指数为整型(可以为负)

要求多项式用带头结点的单链表表示，多项式的项数存放在头结点

需要用指针数组存放N个多项式的头指针

存储结构示意图：



1. 输出多项式

要求输出时按指数升序或降序排列

需要输出的参数有：项数、每项的系数和指数

直接输出和按照代数格式输出均可

1. 多项式的求和

要求可以在1中保存的N个多项式中任选两个进行求和，并输出结果

1. 多项式的求差

要求可以在1中保存的N个多项式中任选两个进行求差，并输出结果

1. 多项式的求值

要求可以在1中保存的N个多项式中任选一个进行求值，并输出结果

自变量值可以任意输入，其类型为浮点型

1. 多项式的销毁

要求可以在1中保存的N个多项式中任选一个进行销毁并释放内存空间

销毁后，多项式列表中剩下N-1个多项式

1. 多项式的清空

要求可以在1中保存的N个多项式中任选一个清空其内容

清空后，多项式列表中仍然剩下N个多项式

1. 多项式的修改

要求可以在1中保存的N个多项式中任选一个进行操作

操作包括：①插入新的结点、②删除已有结点、③修改已有结点的系数和指数

拓展功能：

1. 多项式的微分

要求可以在1中保存的N个多项式中任选一个进行操作

求该多项式的n阶微分，并输出结果

n可以任意输入，其类型为整形

1. 多项式的不定积分

要求可以在1中保存的N个多项式中任选一个进行操作

求该多项式的不定积分，并输出结果

注意：多项式的不定积分后不一定为多项式

1. 多项式的定积分

要求可以在1中保存的N个多项式中任选一个进行操作

求该多项式在区间[a,b]上的定积分，并输出结果

a，b可以任意输入，其类型为浮点型

4. 多项式的乘法和乘方

要求可以在1中保存的N个多项式中任选两个进行求积，并输出结果

5. 多项式的除法

要求可以在1中保存的N个多项式中任选两个进行除法，并输出结果

分别输出商式和余式两个多项式

1. 多项式的四则运算

可以利用输入“编号+运算符”的形式进行多项式的四则运算，

编号为1中保存的N个多项式的编号

要求可以处理加、减、乘、除、括号5种运算符

多项式除法运算可以忽略余式（检查时保证数据不会产生余式）

举例：

多项式列表中保存着：

多项式1：x^2

多项式2：x^3

多项式3：1+x

多项式4：x

输入：(1-2\*3)/4

输出：x-x^2-x^3

1. 计算器的仿真界面

自由发挥

要求：可以在不使用键盘（包括屏幕键盘）只使用鼠标的情况下完成1~14的全部操作

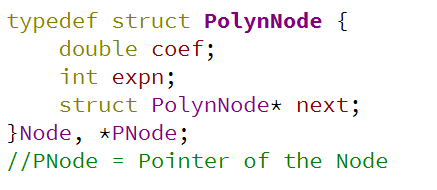
输入：测试文件中包括：多个多项式及一系列操作

输出：操作后的多项式

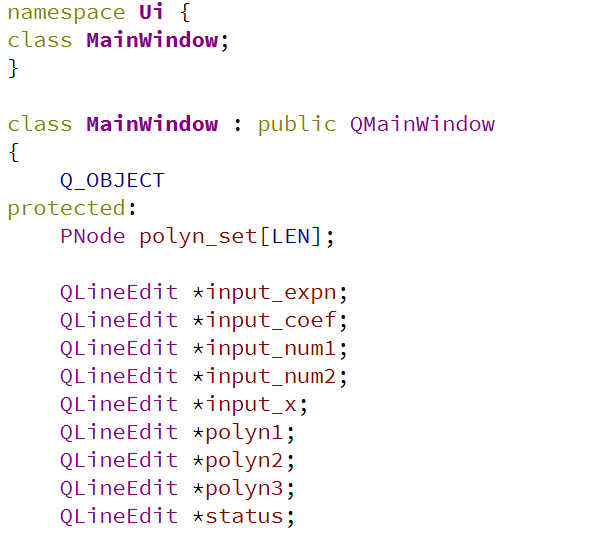
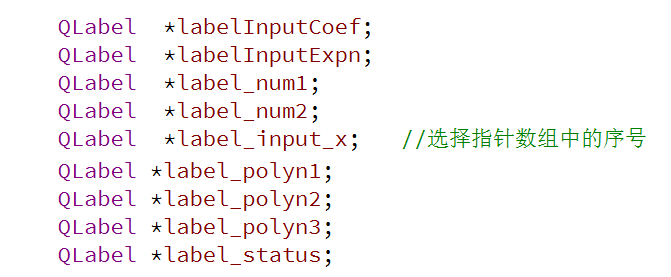
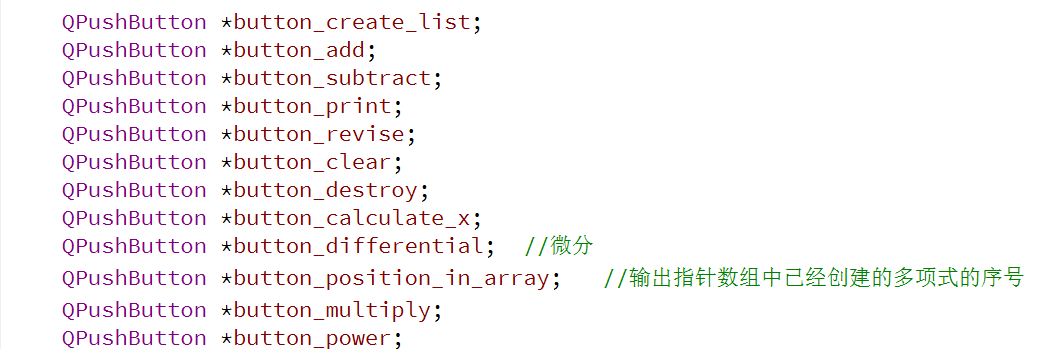
1. 算法描述
2. 数据结构描述

采用线性链表作为存储结构，逻辑结构为线性结构。

主要变量：



由于使用Qt制作图形界面，还定义了一些Qt内置变量

1. 程序结构描述

①Qt 相关的主要函数

槽函数：



以及主窗口布局函数，功能是对窗口控件进行布局。



②多项式操作函数



接口：p为指向多项式结点的指针，第一个int形参为次数，第二个double形参为系数



接口：p为指向多项式结点的指针，temp\_expn为待查找元素的系数，position为返回的元素的位置



接口：p为指向待插入多项式结点的指针，将p指向的结点插入Inserted\_Node指向的结点后面



功能：比较a,b的大小，a>b返回1，a==b返回0，a<b返回-1



功能：创建、插入兼具

接口：L为指向创建后的多项式链表的头结点的指针，PNode&形参表示待插入的结点



接口：L为指向待销毁的多项式链表的头结点的指针

功能：对多项式进行修改



接口：num1，num2是待相加or相减的多项式的序号，PolynSet[ ]是存放多项式指针的指针数组，result是指向运算结果多项式的头结点的指针，if\_sub为0，该函数执行相加，否则执行相减。



功能：在L指向的多项式中插入系数为coef和次数为expn的结点



功能：在L指向的多项式中删除系数次数为expn的结点



功能：a和b指向的多项式相乘

三．调试分析

（1）测试数据使用的是助教提供的测试数据。

（2）程序调试中遇到的问题：

① 前端Qt图形界面与后端多项式处理函数交互的问题

解决方法：使用QString变量获取用户输入，再通过QString的toInt()、tofloat()等函数转换为可以被多项式处理函数处理的数据类型。如下图。



② 创建多项式时有错误。

解决方法：冷静分析CreatePolyn函数流程，发现是由于链表连接不当导致结点丢失。

③ 计算多项式值的时候结果出错。

解决方法：仔细检查后发现是对double类型变量x判断其是否为0一步出错。

不能直接使用x==0，而要使用fabs(x-0.0)<EPS，如下图。

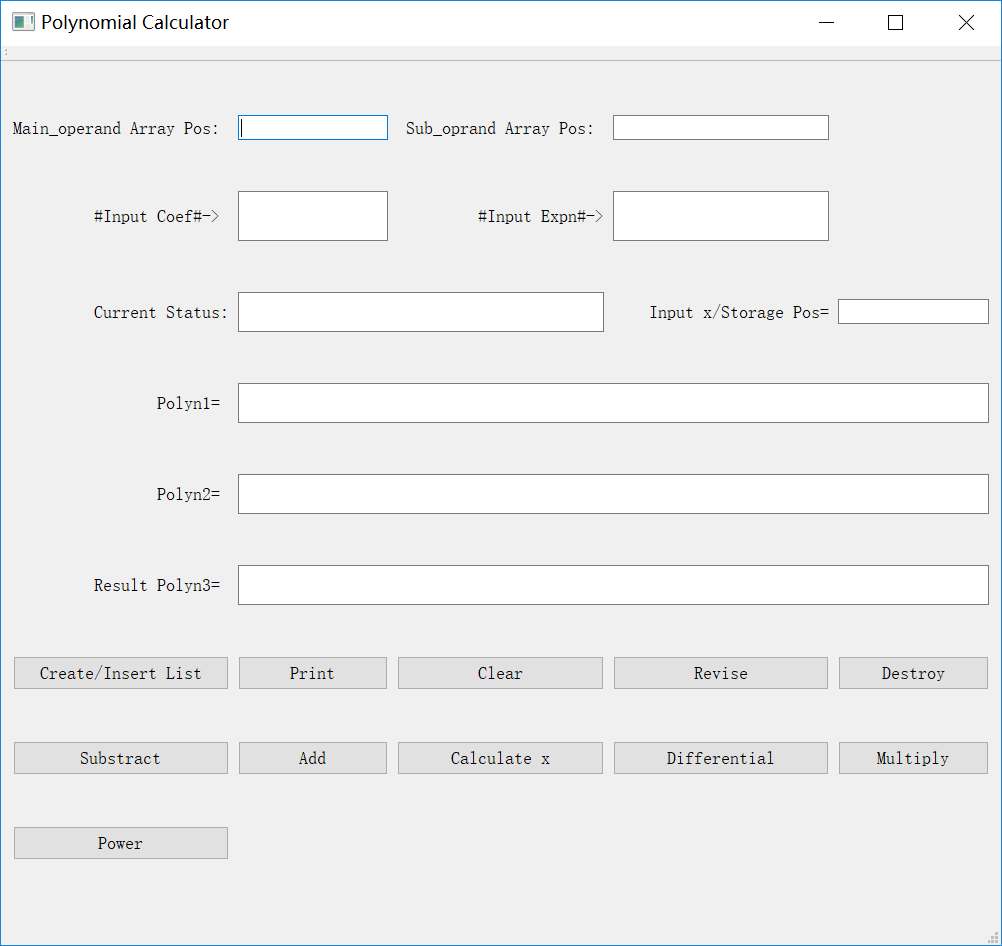


四．算法时空分析

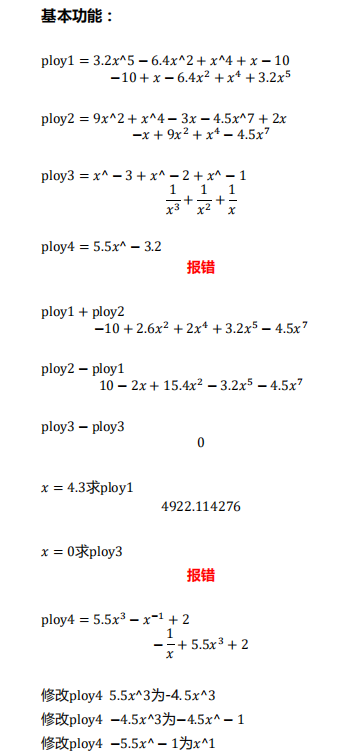
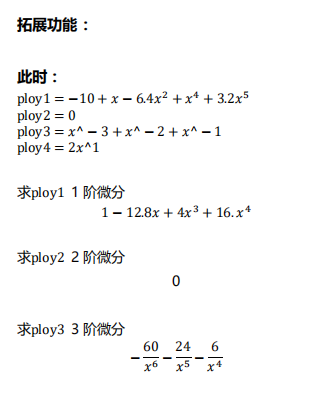
由于程序中函数很多，在此只分析几个重要函数：

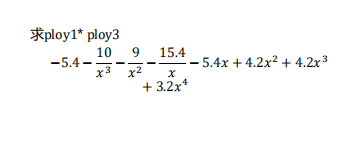
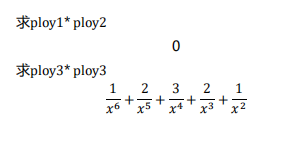
1. 创建多项式：时间和空间复杂度都为O(n)。
2. 多项式加/减法：对a,b多项式遍历，时间复杂度为O(m+n)，空间复杂度因为只开辟固定临时空间，为O(1)。
3. 多项式乘法：时间复杂度为O(n\*n)。创建新链表存储结果，空间复杂度为O(n)。
4. 多项式求微分：需要遍历多项式，时间复杂度为O(n)。因为创建新链表存储结果，所以空间复杂度为O(n)

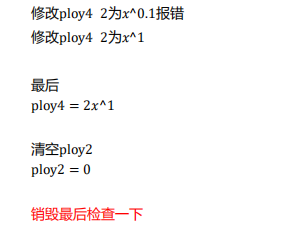
五．测试结果及分析

 最终界面：

例：测试数据为



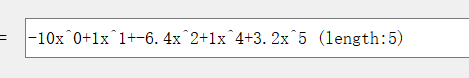




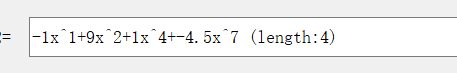
部分结果分析：

（1）创建链表

Polyn1



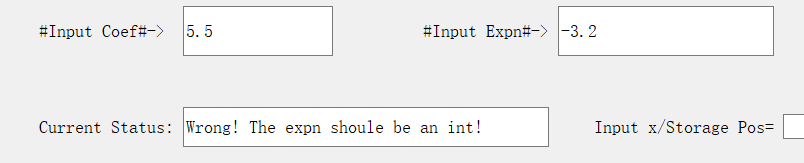
Polyn2



Polyn3

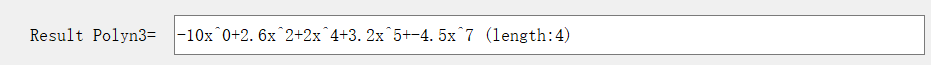


Polyn4，报错

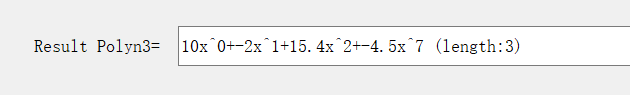


（2）运算

Polyn1+Polyn2，结果正确



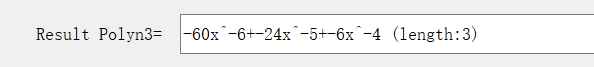
Polyn2-Polyn1，结果正确



Polyn3-Polyn3， 结果正确



求Polyn3的3阶微分，结果正确



Polyn1\*Polyn3，结果正确



六．实验体会和收获

1. 我对制作图形界面十分感兴趣，通过本次实验自学了Qt，了解了制作图形界面的基本方法。
2. 增进了对链表这种数据结构的理解以及多项式的各种复杂操作的链表实现，有效地锻炼了编程能力。