实验 1: 线性表的应用：稀疏一元多项式运算器

1.1 问题描述

通过程序实现一元多项式多种运算，包括：

1.1 输入并创建多项式 期望以 ax^b 的形式输入多项式 **1.2 输出多项式** 以降幂的格式输出所有多项式 **1.3 多项式求和** 选定两个多项式并求和，输出求和后的多项式 **1.4 多项式求差** 选定两个多项式并求差，输出求差后的多项式 **1.5 多项式求值** 对指定的多项式和用户给定的 x, 求 `f(x)` **1.6 多项式销毁** 销毁用户确定的多项式，清除空间。 **1.7 多项式清空** 讲选定的多项式清楚为 `0` **1.8 多项式修改** 1). 搜索指定多项式中的某一项, 并修改它 2). 删除某一项 3). 向多项式中插入一个新项 **1.9 多项式微分** 选定多项式，求 `f'(x)` **1.10 多项式不定积分** 选定多项式，求它的不定积分并输出 **1.11 多项式定积分** 选定多项式，求其定积分，并输出 **1.12 多项式乘法和乘方** 选定两个多项式，进行乘法/选定 1 个多项式，进行乘方操作，输出结果 **1.13 多项式除法** 选定两个多项式并相除，输出结果 **1.14 多项式四则运算** 输入以序号标注的多项式的运算，输出结果

2. 算法描述

2.1 数据结构描述

本实验通过链表实现几乎所有的操作，链表的定义位于 `include/main.h`, 所有的多项式被串在链表中，每一个多项式的存储也由链表实现。 在四则运算时使用了栈的操作，定义在 `include/stack.h`

本实验中所用的 **主要** 函数声明在 `include/main.h` 中，辅助的函数声明位于 `include/assistant.h` 中，`include/common.h` 中声明了一些通用的函数, 程序从用户输入读取选项后，通过 `DoOperation` 函数中的 `case` 调用对应的函数。

2.2 程序结构的描述

实现不同的功能，本程序使用了不同的函数。

2.2.1 输入并创建多项式 程序首先调用 `MakePoly()` 函数，`MakePoly()` 函数从用户输入读取相应的多项式，并调用 `ProcessStr()` 对输入的多项式进行切割，调用 `ProcessExpr()` 对单个表达式进行处理，最后通过 `InsertNode()` 找到对应的节点将 Node 添加到链表当中。

2.2.2 输出多项式 程序调用 `ShowList()` 函数，`ShowList()` 调用 `PrintPoly()` 输出每一个多项式。

2.2.3 多项式求和 多项式求差 程序都调用了 `DisAndCall()` 函数，传入了对应的函数指针 `AddPoly` `SubPoly`, `DisAndCall()` 调用 `ShowList()` 输出所有可以被操作的多项式并请用户选择，最后使用对应的函数 `AddPoly` 或 `SubPoly` 完成相应的功能。

2.2.4 求 f(x) 程序调用 `SolvePoly()` 请求被操作的函数和 `x`，随后进行对应的操作。

2.2.5 多项式销毁 调用 `DelPoly()` 对对应多项式进行 `free()` 操作

2.2.6 多项式清空 调用 `EmptyPoly()` 对多项式置 0.

2.2.7 多项式修改 调用 ChangePoly() 后使用 case 处理请求，分别调用 change_insert_node() change_delete_node() change_change_node() 进行不同的修改，最后使用 PrintPoly() 输出函数

2.2.8 多项式微分 调用 PolyDiF()，调用 poly_dif() 对多项式进行实际的微分操作，并输出多项式，最后询问是否把多项式存入。并不需要考虑 ln 出现的情形。

2.2.9 多项式不定积分 调用 PolyInfI()，调用 poly_infi() 进行实际积分操作，最后询问是否存入。需要考虑 x^−1 情形，方法是在头中定义变量 coffLn 存储 Ln 之前的系数。

2.2.10 多项式定积分 调用 PolyDefI(), 直接调用 **poly_infi()** 进行不定积分，随后进行求值并输出结果，考虑出现 Ln 时 x 不能小于0.

2.2.11 多项式乘法和乘方 乘法调用 DisAndCall() 传入 PolyMultip() 的函数指针，乘方调用 PolyPow() 并在其中调用 PolyMultip()

2.2.12 多项式除法 调用 DisAndCall(PolyDiv, head), 传入函数指针 PolyDiv(), PolyDiv() 调用递归的函数 do_div() 对两个函数递归地相除。

2.2.13 多项式四则运算 调用 PolyExpr(), 使用栈的操作存储元素并进行对应的运算。

3. 调试分析

使用 gdb 对每个功能进行调试，主要是对 x^−1 的定积分处理，多项式的除法和对 ln 的不定积分 x 的定义域。

4. 实验体会和收获

这次实验写了很多的函数来实现一些错综复杂的功能，写出了较为完成且可调试的程序。