实验一 一元多项式运算器设计与实现

1. 问题的描述

（1）设计目标

本实验旨在使用数组和线性表等结构，设计和实现一个稀疏一元多项式运算器。该运算器可以完成的功能包括 1.创建一元多项式；2.显示一元多项式；3.一元多项式求和；4.对一元多项式求微分，计算它的 N 阶导数；5.求一元多项式的不定积分。

（2）输入数据

输入数据可以分为三部分。第一是用户通过菜单选择操作类型。第二是用户使用运算器功能时，比如求和、求导等输入的索引值与相关参数（比如求导次数）。第三是创建多项式时，用户输入多项式的项数、每一项的系数（浮点型）和指数（整型）。

（3）输出结果

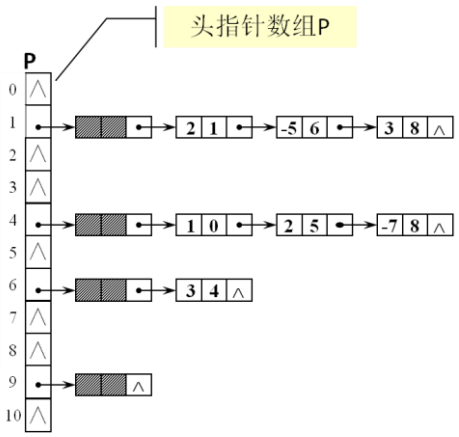
输出结果包括创建的多项式，求和、求导和求不定积分的结果，会显示在屏幕上，输出形式如：X^3–X^2+2X-2，也就是系数为 0 的不会显示，系数为 1 的不会显示系数，其余系数正常显示；指数为 1 的不显示指数项，指数为 0 的那一项就是常数，其余指数正常以^指数的形式显示。

2.算法的描述

(1)数据结构的描述

- ✓ **逻辑结构和存储结构**

总的来说，本程序建立了一个容量为 10 的头指针数组 polyns[]，该数组存放一元多项式的头指针。而具体解释多项式的存储结构的话，则是多项式用带头结点的单链表表示，单链表的 data 域存放系数和指数，单链表的指针域存放指向下一个链表节点的指针。如下图所示：



- ✓ **存储类型定义**

首先定义一个结构体 ElemType 存储多项式的系数（浮点型）和指数（整型）。再定义一

个链表结构体存储刚才构建的结构体 ElemType 以及指向链表的指针。如下图所示：

```
//定义多项式的存储结构
typedef struct ElemType{
    float coef;
    int expn;
}ElemType;
//定义链表存储多项式
typedef struct LNode{
    ElemType data;
    struct LNode *next;
}LNode,*LinkList;
//将指针名字叫做polynomial
typedef LinkList polynomial;
```

✓ **主要变量和数组的说明**

polyns[]：存储多项式的头指针数组。polynomial：指向多项式链表的头指针。select: 用户使用的功能。index: 数组的索引。

(2)程序结构的描述

✓ **函数原型、功能和接口的描述**

针对用户交互，我设计了一个显示菜单的函数。

void showmenu()：该函数不需要传入任何参数，在主函数中调用该函数能够直接显示功能菜单。

针对初始化一个带头结点的单链表，方便后续功能函数的使用，我定义了一个初始化函数。

int InitList(LinkList &L)：该函数需要传入一个链表指针作为引用参数，能够初始化一个只带头结点的单链表。在主函数和功能函数中直接调用该函数即可

针对运算器要执行的五个功能，我分别设定了一个函数。

void CreatePolyn(polynomial &p)：这个函数需要传入一个链表头指针作为引用参数，能够根据用户输入创建一个一元多项式并存储在数组中。通过在主函数中输入 1 调用该函数。

void ShowPolyn(polynomial p,int m)：这个函数需要传入一个链表头指针作为引用参数，以及一个整型的值作为参数，这个整型参数代表是显示不定积分还是显示非不定积分。因为考虑到不定积分要输出常数 C，所以加入了该参数以作区分。这个函数的功能是在屏幕上打印输出要显示的一元多项式。通过在主函数中输入 2 调用该函数。

void AddPolyn(polynomial pa, polynomial pb, polynomial &pc)：这个函数需要传入两个链表头指针作为参数，代表要求和的两项，以及一个新的链表头指针作为引用参数，代表求和结果存放的链表。这个函数的功能是对两个一元多项式求和，并将求和结果存放到一个新的链表中。通过在主函数中输入 3 调用该函数。

void DiffPolyn(polynomial p, int n, polynomial &result)：这个函数需要一个链表头指针作为参数，代表要微分的多项式，和一个整型值 n 作为参数，代表微分的次数，以及一个新的链表头结点作为引用参数，代表微分后的结果。这个函数的功能是对一个一元多项式求 n 次微分，并将微分结果存放到一个新的链表中。通过在主函数中输入 4 调用该函数。

void IntegPolyn(polynomial p, polynomial &result)：这个函数需要一个链表头指针作为参数，代表要积分的多项式，以及一个新的链表头结点作为引用参数，代表积分后的结果。这个函数的功能是对一个一元多项式求不定积分，并将积分结果存放到一个新的链表中。通过

在主函数中输入 5 调用该函数。

3.调试分析

（1）测试数据及方法

测试数据就是输入不同的一元多项式，包括指数为 0，1，2，系数为 0，为正或者为负的项进行创建、显示、求和、求微分、求不定积分的过程，观察输出结果是否有问题。

（2）遇到的问题及解决方法

①在生成新结点时，直接定义一个新指针来操作比如 LinkList qc = pc->next，（在此之前，pc 之后都是 NULL）而不是用 malloc 分配内存，导致程序运行到这一步时，产生 bug，程序运行崩溃。解决办法是用 malloc 函数对新结点分配内存，比如 LinkList newp = (LinkList) malloc(sizeof(LNode));，用 newp 这个结点来存放 qc 的值，再进行链表的连接。

②在执行功能函数时，最开始对输入的索引值 index 没有进行条件判断，而是直接调用相应函数，在程序正常使用时不会报错，但是一旦输入值有问题，比如超过了正常索引范围，或者要显示或求和、微分积分的数组索引位置还没有存放任何一个多项式。这时程序就会发生异常，运行崩溃。解决办法是对 index 加入判断，判断索引有没有超过范围，判断该位置有没有存放多项式。值得一提的是，在判断该位置是否为空时，我最开始写的是 polyns[index]->next !=NULL，这样写在判断这句话时因为 polyns[index]可能就为空，导致程序运行出错，最后也是改成了 polyns[index]!=NULL 才完全解决了这个问题。

③第一次实现的程序在求微分和积分后结果都会覆盖原多项式，导致不能重复使用原创建的多项式。解决办法是在执行功能函数时，将原多项式进行复制操作，保证原有多项式不被覆盖。相比于第一版他的可能缺点是浪费了内存，因为又申请了新的内存存储结果。

④打印不定积分时没有输出常数 C，从数学的角度看这是不正确的。为了保持严谨性，我在 ShowPolyn 这个函数中条件新的参数来区分是否打印 C，这个解决办法是可行的。

4.算法的时空分析

（1）时间复杂度

创建多项式：O(n^2)，其中 n 为多项式的项数。
显示多项式：O(n)，其中 n 为多项式的项数。
求和：O(n)，其中 n 为两个多项式中项数较多的那个。
求导：O(n * m)，其中 n 为多项式的项数，m 为求导次数。
求不定积分：O(n)，其中 n 为多项式的项数。

（2）空间复杂度

创建多项式：O(n)，其中 n 为多项式的项数。
显示多项式：O(1)。
求和：O(n)，其中 n 为两个多项式中项数较多的那个。
求导：O(n)，其中 n 为多项式的项数。
求不定积分：O(n)，其中 n 为多项式的项数。

5.测试结果及分析

（1）创建多项式

输入：（3，2）（1，3）（2，1）（6，0）和（-2，0），（2，1），（-1，2），（1，3）分别存放到索引为 0 和 1 的位置处，程序正常运行。

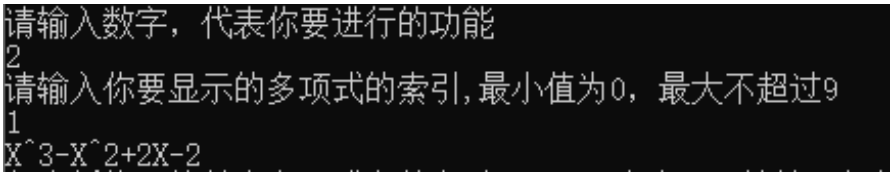
```
请输入数字，代表你要进行的功能
1
请输入你要创建的多项式的索引,最小值为0，最大不超过9
0
请设置欲输入的项数(准备输入几次)：
4
请输入第1项的系数和指数：
3
2
请输入第2项的系数和指数：
1
3
请输入第3项的系数和指数：
2
1
请输入第4项的系数和指数：
6
0
```

```
请输入数字，代表你要进行的功能
1
请输入你要创建的多项式的索引,最小值为0，最大不超过9
1
请设置欲输入的项数(准备输入几次)：
4
请输入第1项的系数和指数：
-2
0
请输入第2项的系数和指数：
2
1
请输入第3项的系数和指数：
-1
2
请输入第4项的系数和指数：
1
3
```

（2）显示多项式

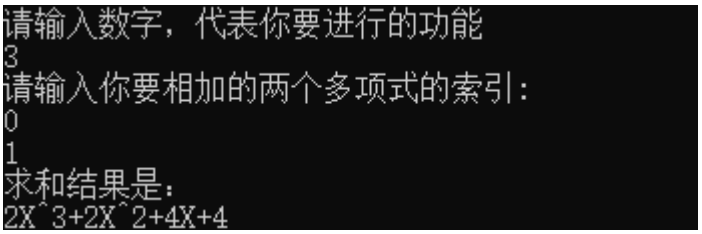
输入 index 值为 0 和 1，打印刚创建的多项式，输出结果为 X3+3X2+2X+6 和 X3-X2+2X-2。

```
请输入数字，代表你要进行的功能
2
请输入你要显示的多项式的索引,最小值为0，最大不超过9
0
X^3+3X^2+2X+6
```



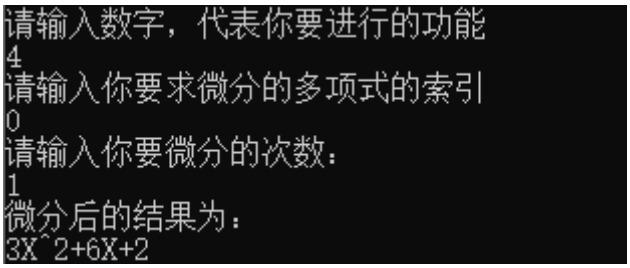
（3）多项式求和

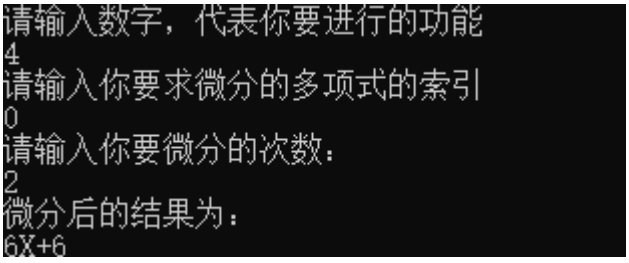
输入 index 值为 0 和 1，对这两个多项式进行求和操作，输出结果为 2X3+2X2+4X+4。



（4）多项式求微分

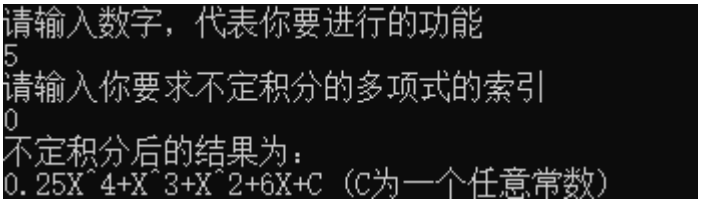
输入 index 值为 0，微分次数分别为 1 和 2，输出结果分别为 3X2+6X+2 和 6X+6。





（5）多项式求不定积分

输入 index 值为 0，输出结果为 0.25X4+X3+ X2+6X+C（C 为一任意常数）。



（6）健壮性检验

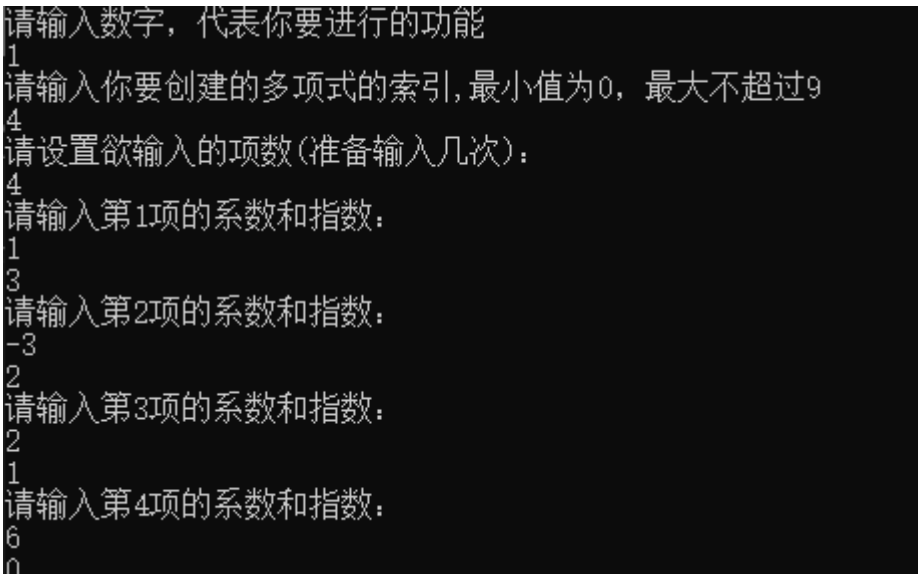
输入 X3-X2+2X-2 时，按 X3、-X2、+2X、-2 顺序输入 和 按-2、X3、-X2、+2X 顺序输入时，结果显示相同，都是 X3-X2+2X-2。

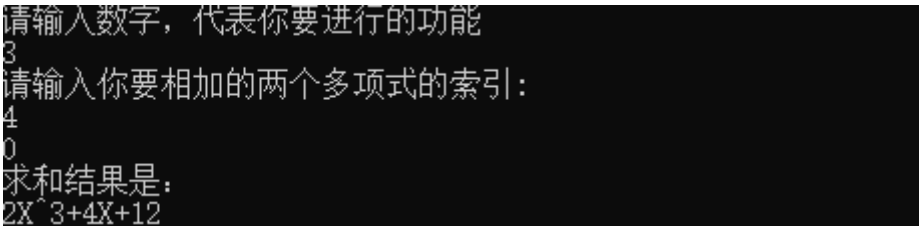
```
请输入数字，代表你要进行的功能
1
请输入你要创建的多项式的索引,最小值为0，最大不超过9
2
请设置欲输入的项数(准备输入几次)：
4
请输入第1项的系数和指数：
1
3
请输入第2项的系数和指数：
-1
2
请输入第3项的系数和指数：
2
1
请输入第4项的系数和指数：
-2
```

```
请输入数字，代表你要进行的功能
1
请输入你要创建的多项式的索引,最小值为0，最大不超过9
3
请设置欲输入的项数(准备输入几次)：
4
请输入第1项的系数和指数：
-2
0
请输入第2项的系数和指数：
1
3
请输入第3项的系数和指数：
-1
2
请输入第4项的系数和指数：
2
1
```

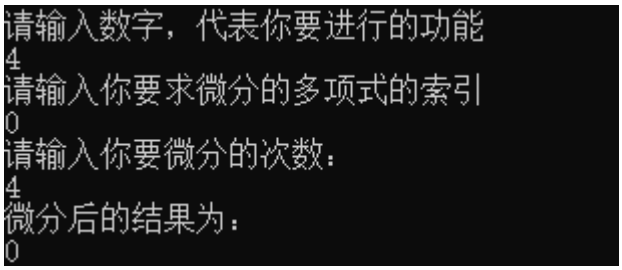
```
请输入数字，代表你要进行的功能
2
请输入你要显示的多项式的索引,最小值为0，最大不超过9
2
X^3-X^2+2X-2
程序暂停，按任意字母或者数字后再enter确认，可继续运行程序
y

@@@@@@@@@@@@@@@@@@@@@@@@@@@@
****0.退出此系统**********
****1.创建一元多项式*****
****2.显示一元多项式*****
****3.一元多项式求和*****
****4.求微分（N阶导数）**
****5.求不定积分**********
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
请输入数字，代表你要进行的功能
2
请输入你要显示的多项式的索引,最小值为0，最大不超过9
3
X^3-X^2+2X-2
```

输入 X3+3X2+2X+6 和 X3-3X2+2X+6，求和结果是 2X3+4X+12，系数为 0 的项被消除。





输入 index 值为 0，求 4 次微分，输出结果是 0。



（7）分析和说明

通过测试，确认程序能够正确执行相应功能，达到了我们实验的目标。创建多项式和显示多项式的功能正常，能够正确存储和显示多项式。求和功能正常，能够正确计算两个多项式的和。求导功能正常，能够正确计算多项式的 N 阶导数。求不定积分功能正常，能够正确计算多项式的不定积分。同时，健壮性检验也顺利通过。

6.实验体会和收获

通过本次实验，我深入理解了一元多项式的存储和操作方法，掌握了链表的基本操作和多项式操作的算法设计。在调试过程中，我遇到了链表结点操作不当、空指针引发的程序崩溃、原有多项式被结果覆盖、计算结果不正确等问题，通过逐步排查和调试，最终解决了这些问题。实验过程中，我对 C++的指针和动态内存分配有了更深入的理解，同时也提高了代

码调试和错误处理的能力。总体来说，本次实验让我在数据结构和算法设计方面有了很大的提升。