实验一 线性表的应用：稀疏一元多项式运算器

崔士强 PB22151743

1 问题描述

1.1 设计目标

本实验的设计目标是创建一个稀疏一元多项式运算器。这个运算器需要实现对稀疏一元多项式的基本操作，包括创建、输出、求和、求差、求值、销毁、清空和修改多项式，另外实现多项式的微分，定积分，不定积分功能。

1.2 输入及输出

每次输入多项式时，逐个输入各项的系数以及指数，程序将这些项按升序排列。经过运算后的多项式按要求存放在列表中，通过函数`void PrintPolyn(Polynomial P)` 输出到屏幕上

2 算法描述

2.1 数据结构描述

本程序使用链表存储多项式，用一个数组存放所有多项式：

```
typedef struct term{        // 多项式的项
    float  coef;              // 系数
    int    exp;               // 指数
}term, ElemType;

typedef struct LNode{       // 结点类型
    ElemType      data;
    struct LNode  *next;
}*Link, *Position;

typedef struct LinkList{    // 链表类型
    Link   head, tail;        // 分别指向线性链表的头结点和最后一个结点
    int    len;               // 指示线性链表中数据元素的个数
}LinkList;

typedef LinkList polynomial; //用带表头结点的有序链表表示多项式
```

2.2 程序结构描述

函数原型及功能说明如下：

```
void CreatePolyn(polynomial &P, int m );
  // 输入m项的系数和指数，建立表示一元多项式的有序链表P
void DestroyPolyn(polynomial &P);
  // 销毁一元多项式P
void PrintPolyn(polynomial P);
  // 打印输出一元多项式P
void ClearPolyn(polynomial p);
  // 将一元多项式P置空
int PolynLength(polynomial P);
  // 返回一元多项式P中的项数
void AddPolyn(polynomial &Pa, polynomial &Pb);
  // 完成多项式相加运算，即：Pa=Pa+Pb，并销毁一元多项式Pb
void SubstractPolyn(polynomial &Pa, polynomial &Pb);
  // 完成多项式相减运算，即：Pa=Pa-Pb，并销毁一元多项式Pb
float EvaluatePolyn(polynomial P, float f);
  // 算出一元多项式P(x)在x=f处的值
void DiffPolyn(polynomial &P);
  // 求一元多项式P的1阶导数
void Integral(polynomial &p);
  // 求一元多项式P的不定积分
void IntPrintPolyn(polynomial P);
  // 打印输出P的不定积分
float IntEvalPolyn(polynomial P, float f);
  // 算出一元多项式P的不定积分在x=f处的值
float DefIntegral(polynomial P, float a, float b);
  // 求一元多项式P在[a, b]上的定积分
```

3 调试分析

3.1 测试数据

选择含正、负、零次项的多项式进行测试，以保证正确性

3.2 问题及解决方法

在测试过程中发现对于 -1 次项的积分，由于其原函数为对数形式，需要单独处理.

解决方法：输出不定积分时，对每一项判断是否是零次项，如果某一项是零次项则输出对数形式。求定积分时同理.

4 算法的时空分析

各项功能的时间复杂度如下所示：

操作	相加/相减	求值	微分	不定积分	定积分
时间复杂度	$O(N+M)$	$O(N)$	$O(N)$	$O(N)$	$O(N)$

5 测试结果及分析

$$P(x) = x^{-1} + 1 + x + 2x^2 + 3x^3 + 4x^4 + 5x^5$$

$$Q(x) = 5x + 4x^2$$

对两个多项式执行以下操作（每次操作后复原）：

1. $P(x) + Q(x)$，并输出序号为 1 的多项式
2. $P(x) - Q(x)$，并输出序号为 1 的多项式
3. 计算 $P(x)$ 在 $x = 4$ 处的值
4. 计算 $P(x)$ 的 3 阶导数
5. 计算 $P(x)$ 的不定积分，并求其在 $x = 1$ 处的值
6. 计算 $P(x)$ 从 -1 到 4 的定积分
7. 将 $P(x)$ 置空，并输出序号为 1 的多项式
8. 将 $P(x)$ 销毁，并输出序号为 1 的多项式

程序输出的结果均符合预期.

6 实验体会和收获

在编写链表相关的函数并进行运用的过程中熟练掌握了链表的相关结构及算法，在编写多项式相关操作算法的过程中增强了对指针操作的把握程度。