



线性表 | 线性表的应用

线性表的基本操作及实现

- 要求：分别采用顺序存储结构和链式存储结构的带头结点的单链表，实现线性表的基本操作，并在main函数中测试各个基本操作的函数是否正确。
- 目的：编程训练；
提高编程技术；
理解并掌握两种存储结构的实现过程。

线性表的简单应用（在线OJ检查）

1. 线性表倒置：把线性表(a_1, a_2, \dots, a_n) 变为(a_n, a_{n-1}, \dots, a_1)

(1) 函数原型1: `void InvertList_SeqStorage(List L);`

功能：将顺序存储结构存储的线性表里面 n 个数据倒置

(2) 函数原型2: `void InvertList_LinkedStorage(List L);`

功能：将链式存储结构存储的线性表的 n 个数据倒置

2. 线性表元素按照访问频度排序

问题：设计一个在线性表中实现Locate运算的函数，使得线性表中所有结点按访问频度递减的顺序排列，以使访问频繁高的结点总是靠近表头 `ElemType Locate(List L, int key);`

线性表的简单应用-讨论

- 题目：大整数加减计算器的设计与实现
- 设计内容：设计并实现一个简单计算器程序，该程序能实现任意大整数的加、减-运算。
- 具体描述为：利用线性数据结构实现长整数的存贮，线性数据结构的每个结点含一个整型变量。输入、输出形式按简体中文表示习惯，每四位一组，组间用逗号分隔。
- 程序要求：
 - (1) 要求可以存储两个无限大的整数；
 - (2) 能够实现任意两个无限大的整数加减运算

线性表的简单应用-讨论

- 讨论内容：
 - 如何能够存储任意大的数据？
 - 采用哪种存储结构？为什么？
 - 基于选择的存储结构，如何实现2个任意大整数的加减法操作？进位和借位怎么处理？画出算法流程图并最终编程实现解决任意大整数的存储与加减法计算功能。