



---

# 数据结构-实验四





# 实验四：线性表

---

## 一、实验目的

- 1、复习线性表的逻辑结构、存储结构及基本操作；
- 2、掌握顺序表和（带头结点）单链表；
- 3、了解有顺表。



# 预备知识

顺序表:

```
#define LIST_INIT_SIZE 100           //初始容量
```

```
#define LISTINCREMENT 10           //容量增量
```

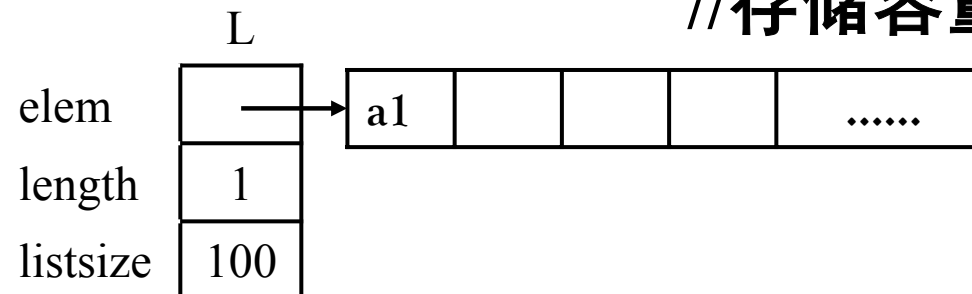
```
typedef struct {
```

```
    Elemtype *elem;           //存储基址
```

```
    int length;               //当前长度
```

```
    int listsize;             //存储容量
```

```
} Sqlist;
```

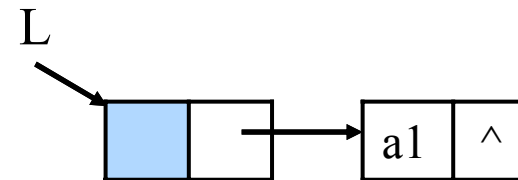
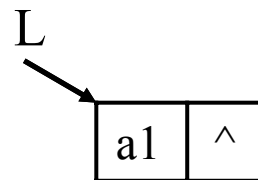




# 预备知识

(带头结点) 单链表-1:

```
typedef struct Lnode {  
    ElemType data;           //数据域  
    struct Lnode *next;      //指针域  
} Lnode, *LinkList;
```



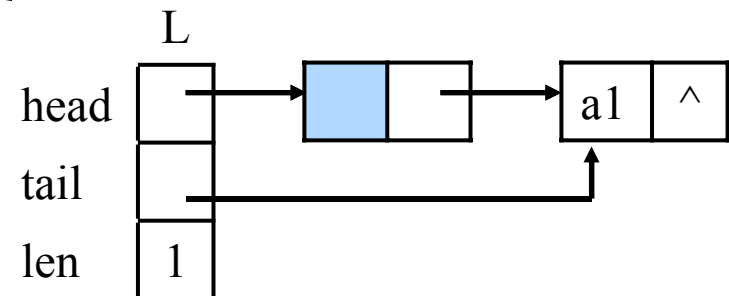
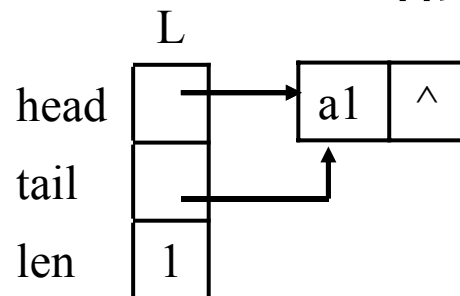


# 预备知识

(带头结点) 单链表-2:

```
typedef struct Lnode { //结点类型
    ElemType data;      //数据域
    struct Lnode *next; //指针域
} Lnode, *Link;
```

```
typedef struct LinkList { //链表类型
    Link head, tail;      //分别指向头结点和尾结点
    int len;              //当前长度
} LinkList;
```





# 预备知识

---

思考1：线性表的基本操作应该包括哪些？

思考2：与单链表相比，带头结点的单链表有什么好处？

思考3：如何构造有序表？



# 实验四：线性表

## 二、实验内容

1、（必做题）假设有序表中数据元素是正整数，请采用顺序表或（带头结点）单链表实现：

（1）OrderInsert(&L, e, int (\*compare)(a, b))

//根据有序判定函数compare，在有序表L的适当位置插入元素e；

（2）OrderInput(&L, int (\*compare)(a, b))

//根据有序判定函数compare，并利用有序插入函数OrderInsert，构造有序表L；

（3）OrderMerge(&La, &Lb, &Lc, int (\*compare)())

//根据有序判定函数compare，将两个有序表La和Lb归并为一个有序表Lc。



# 实验四：线性表

---

## 二、实验内容

2、（选做题）请实现：

(1) 升幂多项式的构造，升幂多项式是指多项式的各项按指数升序有序，约定系数不能等于0，指数不能小于0，例如， $2-100x^{14}+3x^{100}-16x^{1000}$ ；

(2) 两个升幂多项式的相加。

3、（选做题）数据结构MOOC的第二周编程实训题第3题：2-3 两个有序链表序列的交集（在拼题A网站<https://pintia.cn/>上完成）。