



数据结构实验

实验四

1



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、了解有序表。

2



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;
```

3



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;
```

4

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;
```

5

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?

6

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类型是整型, 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。

7

实验四: 线性结构



二、实验内容

- 2、(必做题) 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;
 - (2) 两个升幂多项式的相加。

8

实验四：线性结构



二、实验内容

3、（选做题）约瑟夫环问题。

问题描述：将数字 $1, 2, \dots, n$ 环形排列；按顺时针方向从1开始计数，计满 k 时输出该位置上的数字，同时从环中删除该数字；然后从下一位置开始重新开始计数，直到环中所有数均被输出为止。

请使用顺序表或链表实现：

对输入的任何 n 和 k ，输出相应的出列序。