

实验二 单链表的插入

【实验目的】

- (1) 掌握 C 语言中结构体类型和指针类型。指针是逻辑关系的映像。
- (2) 掌握线性链表的操作特点和动态产生单链表的方法。
- (3) 掌握单链表的插入操作的特点和方法。

【实验内容】

设 $X = (x_1, x_2, \dots, x_n)$ 和 $Y = (y_1, y_2, \dots, y_m)$ 是两个链式存储的线性表，试把它们合并为一个线性表 Z ，而结点的物理存储位置不变，新的线性表为：

$$Z = \begin{cases} (x_1, y_1, x_2, y_2, \dots, x_n) & \text{当 } n > m \text{ 时} \\ (x_1, y_1, x_2, y_2, \dots, y_m) & \text{当 } n \leq m \text{ 时} \end{cases}$$

【实验要求】

从键盘输入两组整数（每组 7~8 个数据），分别动态产生带头结点的单链表 X 和 Y ，并输出结点值。然后合并为线性表 Z ，并观察输出结果。要求在不求出两链表长度的情况下，正确的合并两表。

【实验提示】

涉及动态生成一个结点。因此文件开头应有 `# include "malloc.h"`。此头文件中包含 `malloc(x)` 函数。在 TC 中可能是 `alloc.h`

以下为动态生成一个数据域值为 a 的结点的典型语句：

```
struct node *t;
t = (struct node *)malloc(sizeof(struct node));          /* t 为新生成的结点的指针 */
t->data = a;
```

涉及到两个链表 X 和 Y 的创建，因此可设计一个创建单链表的函数。

涉及到三个线性表 X, Y, Z 的打印，可设计一个打印单链表的函数。

首先分别生成链表 X, Y （可以用尾插法），分别打印之；然后将两表合并，再打印合并后的链表。