链表的实现&熟悉链表的应用场景

基础部分

1. 用带头节点的单链表存储 1~100 的整数，并打印

   （本题主要考察链表的实现，可以参考书中 2.3 节线性表的链式表示和实现。

   1. 建立链表 node 结构，
   2. 建立链表创建函数。（输入：链表长度；输出：链表指针）

2. 实现函数 void insert(list &L, int i, ElemType e)，在带头节点的单链表中第 i 个元素之前插入元素 e。在本题中，调用函数，插入第 101 个节点，节点数据为 101，并打印（本题主要考察链表的基本操作）。
3. 实现函数 void delete(list &L, int i, ElemType e)，在带头节点的单链表中删除链表中第 i 个元素，并返回删除元素值。在本题中，调用函数，删除第 100 个节点，并打印。

进阶部分

1. 求两个带头节点的单链表相交的第一个节点

通过如下函数实现：

ListNode* FindFirstCommonNode(ListNode *pHead1, ListNode *pHead2)

思路：如果两个尾结点是一样的，说明它们用重合；否则两个链表没有公共的结点。先分别遍历两个链表得到它们的长度，并求出两个长度之差。在长的链表上先遍历若干次之后，再同步遍历两个链表，直到找到相同的结点，或者一直到链表结束。此时，如果第一个链表的长度为 m，第二个链表的长度为 n。

2. 将带头结点的单链表**就地**逆置（或者叫单链表的反转，即原来是 1~101，现在是 101~1。注意就地逆置的意思是该算法的空间复杂度为 O（1)），并打印输出 101~1。