1. 假设x指向一条列表中的某个结点，请分别给出每个代码块的注释及功能。

1）x->succ = x->succ->succ; // x不是指向尾结点

2）t->succ = x->succ;

x->succ = t;

3）**ListNode\*** fun(**ListNode\*** x){ //x指向列表的头结点

**ListNode\*** a = x;

**ListNode\*** b = NULL;

while (a != NULL){

**ListNode\*** c = a->succ;

a->succ = b;

b = a;

a = c;

}

return b;

}

2. 算法设计题，以下每道题请给出C++代码，并给出必要的注释。

1) 给定一个单向列表，请编写一个函数得到该列表中倒数第个节点，给出必要注释。要求：空间复杂度为O(1)，时间复杂度为O(n)，且只能遍历一次列表。

2）给定一个单向列表，请编写一个函数，删除列表的倒数第个节点。要求：空间复杂度为O(1)，时间复杂度为O(n)，且只能遍历一次列表。

3. 已知两个列表，请设计一种算法得到它们的第一个公共节点（请注意，输出节点，而非节点内容）。要求：用文字描述算法，并给出伪代码，空间复杂度为O(1)，时间复杂度为O(n)。

举例：下图中蓝色节点为输出节点。



4. 对于有序向量（升序），请写出对其进行二分查找的伪代码，并分析时间复杂度。

5. 对于有序单向列表（升序），请写出对其进行二分查找的伪代码，并分析时间复杂度。