共100分，第一题30分，第二题30分，第三题40分。

第一题：判断不减得20分，判断不为空得10分。

第二题：功能未实现直接0分，用链表而非顺序表得0分，一个复杂度不对扣10分，方法非O(n)复杂度扣10分。

第三题：正确实现求交集的功能得20分，2个复杂度各占10分。

表述不清楚要酌情扣分。

第二章 线性表习题

2.1 请说明下面程序段的作用。其中，L 是一个单链表，结点有数据域 data 和指针域 next。

pre = L -> next;

if (pre) {

while (pre -> next) {

p = pre->next;

if (p-> data >= pre-> data) {

pre = p;

}

else {

return false;

}

}

return true;

}

else return false;

判断L是否不减且不为空

* 1. 已知一顺序表 A有n个元素，其元素值递减有序排列，编写一个算法删除顺序表中多余的值相同的元素。（伪代码就行，要求：空间复杂度为O(1)，分析代码的时间复杂度）

用一个变量记录当前拷贝位置p，遍历A，若其值与前一个不相同，则将这个值赋予位置p，并使得p++。

时间复杂度 O(n)，空间复杂度O(1)

int p=1;

for(int i=1; i<A.size(); i++)

{

if (A[i] != A[i-1])

{

A[p] = A[i];

p++;

}

}

最后一并返回新的顺序表的长度

上面也可用A[p-1]代替A[i-1]，也可设置bias偏置，用i-bias来定位A[p]。

* 1. 已知两个单链表 A 和 B 分别表示两个集合，其元素严格递增排列，编写算法求出 A 和 B 的交集 C，要求 C 同样以元素递增的单链表形式存储。（伪代码就行，要求：分析代码的时间复杂度和空间复杂度）

Node\* pa = A.head;

Node\* pb = B.head;

List C;

While(pa != NULL && pb != NULL)

{

If(pa->val == pb->val)

{

C.insert(pa->val);

pa = pa->next;

pb = pb->next;

}

else if(pa->val > pb->val)

{

pb = pb->next;

}

else if(pa->val < pb->val)

{

pa = pa->next;

}

}

时间复杂度O（m+n），空间复杂度O(m+n)或O(A交B)或O(min{m,n})。

如果是把交集结果直接存在原有的链表中，空间复杂度可优化到O(1)。