**数据结构课程设计**

**-两个有序链表序列的交集**

# 题目描述

1. 项目内容

已知两个非降序链表序列S1和S2，设计函数构造出S1和S2的交集新链表S3。

1. 项目功能要求

输入说明：输入分2行，分别在每行给出由若干个正整数构成的非降序序列，用-1表示序列的结尾（-1不属于这个序列）。数字用空格间隔。

输出说明：在一行中输出两个输入序列的交集序列，数字间用空格分开，结尾不能有多余空格；若新链表为空，输出NULL。

# 总体思路

项目要求已经明确要求使用链表作为数据结构。在第一个项目中已实现自定义的链表类型，可以直接复用代码。且此功能中仅用到链表的尾插入和删除节点的操作。求交集时，用两个指针分别遍历链表，选取链表一作为结果链表。每遍历一个元素时，在第二个链表中找到等于它的最后的元素保留，如找不到则删除该元素。

# 代码类设计

源文件包含了输入两个非降序序列，和些许输入提示。

头文件mylist.h包含了对链表节点node类和链表list类的定义，与第一个项目中链表不同的是还添加了游标iterator类在list类中以方便元素的遍历查找。

以下是数据类型和链表相关类的定义：

# 具体实现

1. 输入处理

两个链表分别输入，共用一个输入函数，传参不同链表，输入函数input中遇到输入非法字符和输入内容小于前一个数字则清空当前链表，返回false，重新输入，否则持续输入至遇到-1。

1. 取交集

包装为unit函数，双指针遍历两链表，我们设两链表为和，两指针分别为和，遍历终止条件为其中一个指针遍历到了链表末尾。遍历过程中，如果小于，则删除指向的元素，后移；如果相等，两指针均后移；如果大于，只需要后移。其中需要一个辅助整型变量，用以标记删除的位置序号。最后将中如果有的剩余元素全部清空。

1. 链表类实现

在项目一种list类的基础上保留了尾插入和删除节点的功能，并添加了游标类用以遍历链表。对前自增后自增、等于不等于、外加括号和指针取值进行了重载。

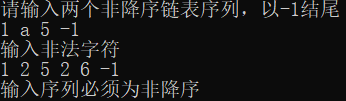
# 项目小结

性能分析

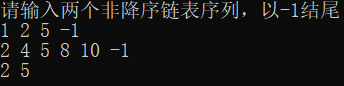
内存开销仅取决于链表的长度，因此空间复杂度为。时间上最坏对链表每个元素都遍历一遍，时间复杂度最多为，最好的情况就是长度短的链表是长度长的链表的最前的元素，如此时间上仅对长度短的链表遍历一遍。

# 程序功能测试

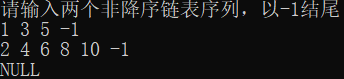
1. 输入非法字符或不满足非降序



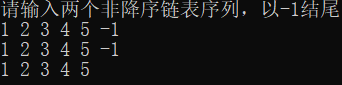
1. 一般情况



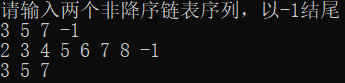
1. 交集为空



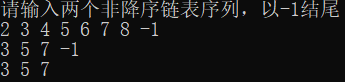
1. 完全相交



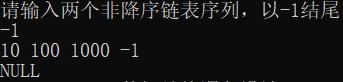
1. 其中一个序列为另一个的子集
2. 是的子集



1. 是的子集



1. 其中一个序列为空
2. 为空



1. 为空

