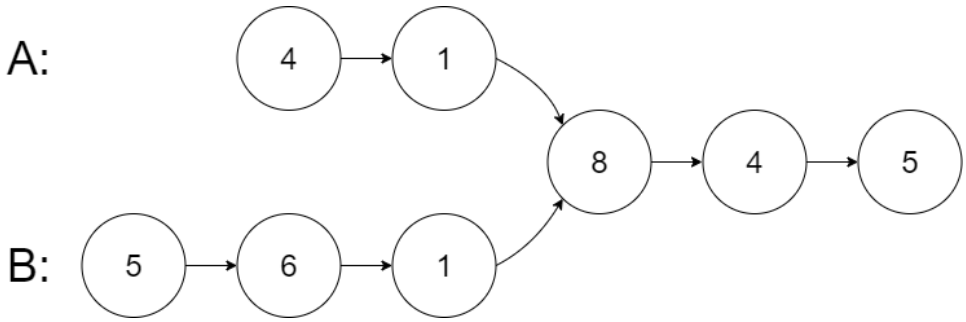
Problem 2

Problem Statement:

Given two singly connected linked lists, determine if both the linked lists intersect. If they intersect Return the pointer to the node of the linked list where lists A and B intersect. For example, in the given example, A and B intersect at node with value 8, hence you should return the pointer to node with value 8.



Function Description:

Function signature:

node* FindCommonList(node *headA, node *headB)

You will have to complete the function FindCommonList which takes as parameters pointers to the starting node of two singly connected linked lists. The function should return the pointer to the intersecting node. If the two lists do not intersect then return NULL. You only have to code with this function any other part of the template given to you should not be changed. The input format is given to help you understand the test case. You can download the template C program from this link

https://iiitbac-my.sharepoint.com/:u:/g/personal/vivek_yadav_iiitb_ac_in/ESHDUWKtK1tAo2fWd6IRKilByJV3CHoR4KSeIqBLMfvQ_A?e=9lWHQ7

Input:

First line contains a single integer t, the total number of nodes in both the linked lists. The next line contains an array *A* of *t* space separated integers, the ith integer *Ai* denotes the data stored in the ith node. The next line of the input contains the description of in the First linked list as space separated integers, ith integer denoting the index of the ith node in the input array *A*. The next line contains the description of second linked list in the format similar to the first linked list.

Output:

Output a single line with space separated integers, the first integer *k* denotes the length of the common intersection of both the linked lists. Then *k* space separated integers denote the common part linked list between the two linked lists given in the input. If the two linked lists do not intersect output *k = 0*.

Input	Output
8 4 1 8 4 5 5 6 1 1 2 3 4 5 6 7 8 3 4 5	3 8 4 5