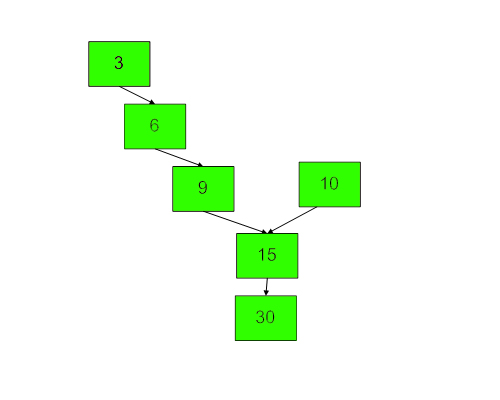
**Intersection Point in Y Shapped Linked Lists**

Given two singly linked lists of size **N** and **M,**write a program to get the point where two linked lists intersect each other.

  
Above diagram shows an example with two linked list having 15 as intersection point.  
**Note:** Expected time complexity is O(m + n) where m and n are lengths of two linked lists.

**Input:**  
First line of input is the number of test cases T. Every test case has four lines. First line of every test case contains three numbers, **x** (number of nodes before merge point in 1st list), **y**(number of nodes before merge point in 2nd list) and **z** (number of nodes after merge point). Next three lines contain x, y and z values.

**Output:**  
Print the data of the node in the linked list where two linked lists intersects.

**Your Task:**  
The task is to complete the function **intersetPoint**() which finds the point of intersection of two linked list. The function should return data value of a node where two linked lists merge. If linked list do not merge at any point, then it should return **-1**.

**Challenge**: Try to solve the problem without using any extra space.

**Expected Time Complexity:** O(N+M)  
**Expected Auxiliary Space:** O(1)

**Constraints:**  
1 <= T <= 50  
1 <= N <= 100  
1 <= value <= 1000

**Example:  
Input:**  
2  
2 3 2  
10 20  
30 40 50  
5 10  
2 3 2  
10 20  
30 40 50  
10 20  
**Output:**  
5  
10

**Explanation:  
Testcase 1:** The point of intersection of two linked list is 5, means both of them get linked (intersects) with each other at node whose value is 5.  
**Testcase 2:**The point of intersection of two linked list is 10. Hence, output is 10.