**Intersection of two sorted Linked lists : -**

Given two lists sorted in increasing order, create a new list representing the intersection of the two lists. The new list should be made with its own memory — the original lists should not be changed.

**Input:**  
You have to complete the method which takes 3 argument: the head of the first linked list , the head of second linked list and the head of the third link list which is to be created. You should not read any input from stdin/console. There are multiple test cases. For each test case, this method will be called individually.

**Output:**  
Complete the Function given to produce the desired list with intersectioned values.

**User Task:**  
The task is to complete the function **intersection**() which should find the intersection of two linked list and add all the elements in intersection to the third linked list.

**Expected Time Complexity** : O(n+m)  
**Expected Auxilliary Space** : O(n+m)

**Constraints:**  
1 <= T <= 100  
1 <= size of linked lists <= 5000  
1 <= Data in linked list nodes <= 1000

**Example:**  
**Input:**  
3  
5 4  
1 2 3 4 6  
2 4 6 8  
4 2  
10 20 40 50  
15 40

5 4

1 2 2 3 4

2 2 4 5

**Output:**  
2 4 6  
40

2 2 4

**Explanation:  
Testcase 1**: For the given first two linked list, 2, 4 and 6 are the elements in the intersection.