**Move all zeros to the front of the linked list: -**

Given a linked list, the task is to move all 0’s to the front of the linked list. The order of all another element except 0 should be same after rearrangement.

**Input:**  
The first line of input contains an integer **T**denoting the number of test cases. For each test case, the first line contains an integer **N**denoting the number of elements in the Linked List and the second line contains N-space separated integer inputs i.e. elements in the Linked List.

**Output:**  
For each test case, the output is the modified linked list having all 0's in front.

**User Task:**  
The task is to complete the function **moveZeroes**() which should move all the 0's to the front of the list.

**Constraints:**  
1 <= T <= 100  
1 <= N <= 50  
**Note:**List is from back to front.

**Example:  
Input:**  
2  
10  
0 1 0 1 2 0 5 0 4 0  
7  
0 0 0 3 2 1 1

**Output:**  
0 0 0 0 0 4 5 2 1 1  
0 0 0 1 1 2 3

**Explanation:  
1.**Original list was 0->4->0->5->0->2->1->0->1->0->NULL.  
     After processing list becomes 0->0->0->0->0->4->5->2->1->1->NULL.  
**2.** Original list was 1->1->2->3->0->0->0->NULL.  
    After processing list becomes 0->0->0->1->1->2->3->NULL.