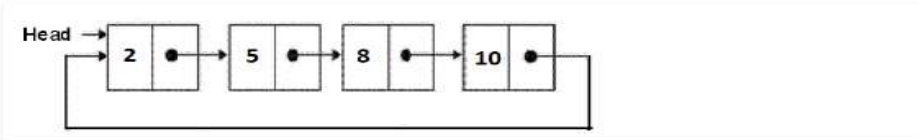
19CSE 212: Data structures and Algorithms
Lab Sheet 4
Circular Linked List and Doubly Linked List

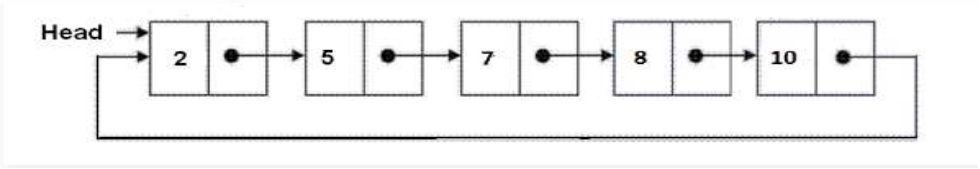
---

1. Implement the following in **a circular singly linked list.**
   a. Insert at head.
   b. Insert at last.
   c. Insert after a node.
   d. Delete a node with given data item.
   e. Delete a node at a given position.
   f. Searching an element.
2. Implement the following in a **doubly linked list.**
   a. Insert at a position, after a given node.
   b. Delete a node with given data.
   c. Sort the list.
   d. Reverse first k elements.
3. Implement a procedure **Sorted _insert ()** for **circular singly linked list.**

   The function **Sorted _insert()** should insert a new value in a sorted **Circular Linked List** (CLL). For example, if the input CLL is following.



After insertion of **7**, the above CLL should be changed to following



4. Implement the **SortedMerge()** function that takes two **doubly-linked lists**, each of which is sorted in increasing order, and merges the two together into one list which is in increasing order. **SortedMerge()** should return the new list. The new list should be made by splicing together with the nodes of the first two lists. For example, if the first list a is **5->10->15**

and the other list b is **2->3->20**, then **SortedMerge()** should return a pointer to the head node of the merged list **2->3->5->10->15->20.**

5. Implement the **sumof pair()** function that takes a sorted doubly linked list of positive distinct elements and find pairs in a doubly linked list whose sum is equal to the given value x.

   Example:

   Input : : 1 <-> 3<->4 <-> 5 <-> 7 <-> 8 <-> 9

   x = 8

   Output: (1,7), (3,5)