**Task 1**Write a menu drive program to insert, delete, update and show the singly linked list. Implement the following functions.

1. Insert at Tail
2. Insert at Head
3. Insert after specific data
4. Insert before specific data
5. Insert after specific node (e.g. 3rd Node)
6. Insert before specific node
7. Update specific data
8. Delete Specific Node
9. Delete 1st Node
10. Delete Last Node

**Task 2**Write a menu drive program to insert, delete, update and show the doubly linked list. Implement the following functions.

1. Insert at Tail
2. Insert at Head
3. Insert after specific data
4. Insert before specific data
5. Insert after specific node (e.g. 3rd Node)
6. Insert before specific node
7. Update specific data
8. Delete Specific Node
9. Delete 1st Node
10. Delete Last Node

**Task 3**

Create a function to split 2 singly Linked Lists according to following rules.

Split 1 linked list into 2 linked lists with alternate nodes.

**Input:**   
List: **1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10**

**Output** :   
List 1 : **1 -> 3 -> 5 -> 7 --> 9**List **2 : 2 -> 4 -> 6 --> 8 -> 10**

**Note: There is no use of Even and Odd (Data), Alternate nodes mean First linked list will have nodes starting from head with skipping next node.   
In other words, nodes on 1,3,5,7 … positions will part of link 1 and nodes on position 2,4,6,8 will part of list 2.**

**Task 4**

Create a function to merge 2 singly Linked Lists according to following rules.

Insert nodes of 2nd linked list into 1st linked list at alternate position of first list.  
**For Example :**   
List 1 : **5-> 7 -> 17 -> 13 -> 11**  
List 2: **12 -> 10 -> 2 -> 4 -> 6**   
  
**Output** :   
List 1: **5 -> 12 -> 7 -> 10 -> 17 -> 2 -> 13 -> 4 -> 11 -> 6**  
List 2: Empty   
**Note: The nodes of second list should only be inserted when there are positions available.**   
  
**For Example :**   
List 1: **1 -> 2 -> 3**   
List 2: **4 -> 5 -> 6 -> 7 -> 8**

**Output:**  
List 1: **1-> 4 -> 2 -> 5 -> 3 -> 6**  
List 2: **7 -> 8**