

DATA STRUCTURES AND ALGORITHMS
Unit-II-Assignment

Reg. No. : RA2311026010677 Name : Nandita Nahata

Year : 2024 . Branch: CSE AI/ML Section: AA-2

I. Choose the best answer:

1. Which of the following is not true about linked lists?
 - (a) It is a collection of linked nodes.
 - (b) It helps in dynamic allocation of memory space.
 - (c) It allows direct access to any of the nodes.
 - (d) It requires more memory space in comparison to an array.

2. Which node pointers should be updated if a new node B is to be inserted in the middle of A and C nodes of a singly linked list?
 - (a) NEXT pointer of A and NEXT pointer of C
 - (b) NEXT pointer of B and NEXT pointer of C
 - (c) NEXT pointer of B
 - (d) NEXT pointer of A and NEXT pointer of B

3. A circular linked list contains four nodes "A, B, C, D". Which node pointers should be updated if a new node E is to be inserted at end of the list?
 - (a) NEXT pointer of D and NEXT pointer of E
 - (b) NEXT pointer of E
 - (c) NEXT pointer of E and NEXT pointer of A
 - (d) NEXT pointer of E and START POINTER

4. Which node pointers should be updated if a new node B is to be inserted in the middle of A and C nodes of a doubly linked list?
 - (a) NEXT pointer of A, PREVIOUS pointer of B, NEXT pointer of C, and PREVIOUS pointer of C
 - (b) NEXT pointer of A, PREVIOUS pointer of B, NEXT pointer of B, and PREVIOUS pointer of C
 - (c) NEXT pointer of A, PREVIOUS pointer of A, NEXT pointer of B, and PREVIOUS pointer of C
 - (d) None of the above

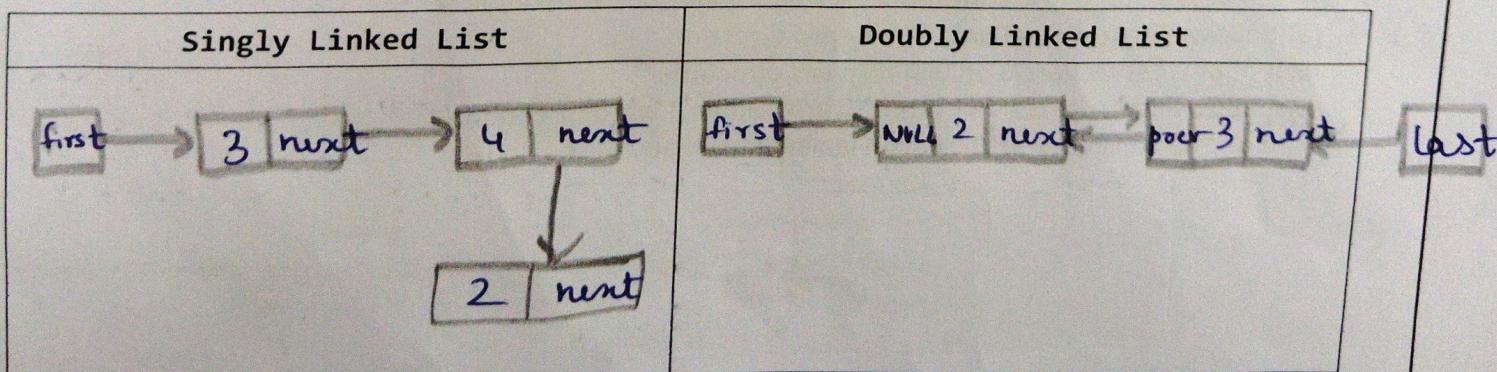
5. Which of the following statements is true about doubly linked list?
 - (a) It allows list traversal only in forward direction.
 - (b) It allows list traversal only in forward direction.
 - (c) It allows list traversal in both forward and backward direction.
 - (d) It allows complete list traversal starting from any of the nodes.

6. Which of the following statements is true about circular linked list?
- (a) It allows complete list traversal starting from any of the nodes.
 - (b) It allows complete list traversal only if we begin from the FIRST node.
 - (c) Like singly and doubly linked lists, the NEXT part of the last node of a circular linked list contains a NULL pointer indicating end of the list.
 - (d) None of the above
7. You are required to create a linked list for storing integer elements. Which of the following linked list implementations will require maximum amount of memory space?
- (a) Singly linked
 - (b) Doubly linked
 - (c) Circular
 - (d) All of the above will occupy same space in memory
8. Which of the following linked list types allows you to print the list elements in reverse order?
- (a) Doubly
 - (b) Singly
 - (c) Circular
 - (d) None of the above
9. Which of the following is the fastest and easiest sorting technique?
- (a) Bubble
 - (b) Quick
 - (c) Insertion
 - (d) Bucket
10. Which of the following searching techniques mandatorily requires the list to be already sorted?
- (a) Linear
 - (b) Binary
 - (c) Hash
 - (d) None of the above

ANSWERS

1.		2.		3.		4.		5.		6.		7.		8.		9.		10.	
----	--	----	--	----	--	----	--	----	--	----	--	----	--	----	--	----	--	-----	--

II. Draw my node:



On 2
Online - in

III. Declare me:

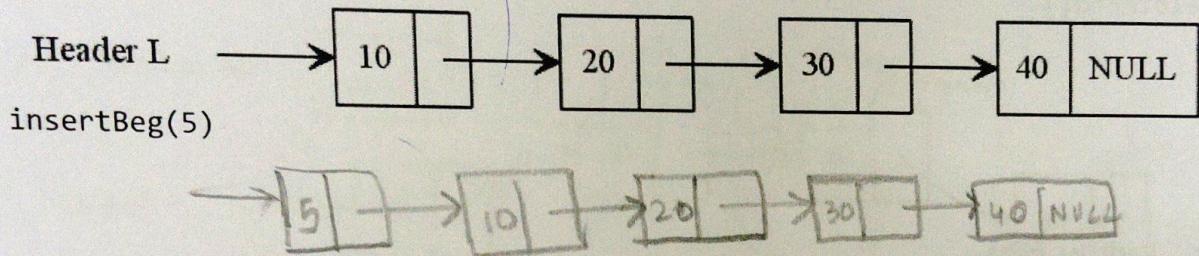
Singly Linked List

```
Struct Node<-->
    int data;
    struct Node* next;
3;
```

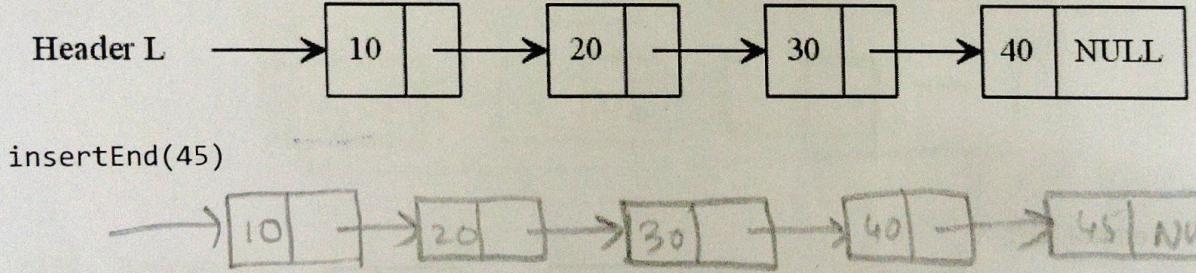
Doubly Linked List

```
Struct Node<-->
    int data;
    struct Node* next;
    struct Node* prev;
3;
```

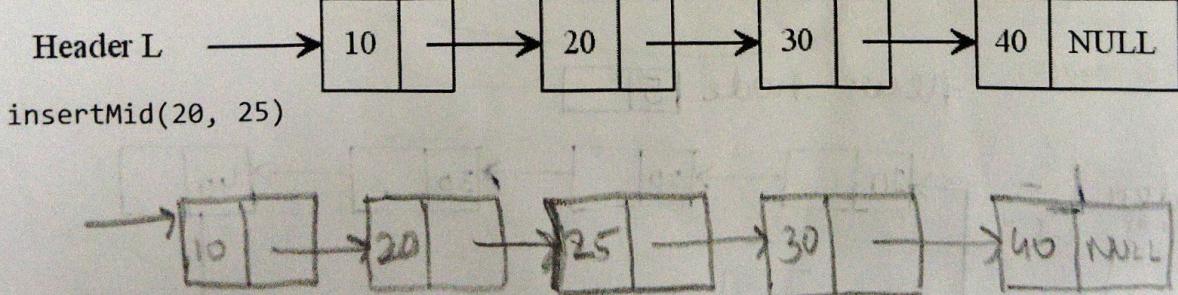
IV. Redraw me:



V. Redraw me:

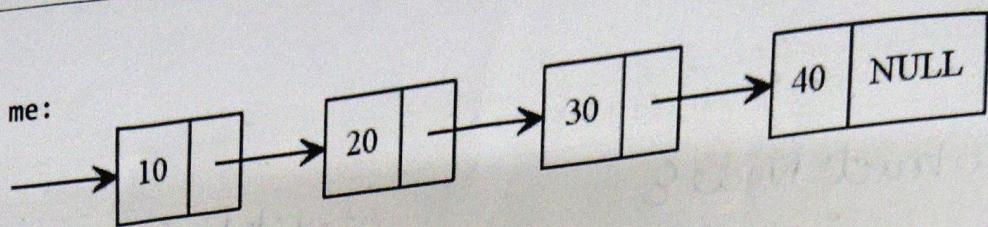


VI. Redraw me:

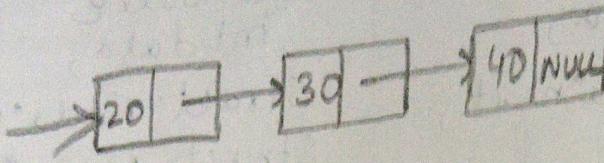


VII. Redraw me:

Header L

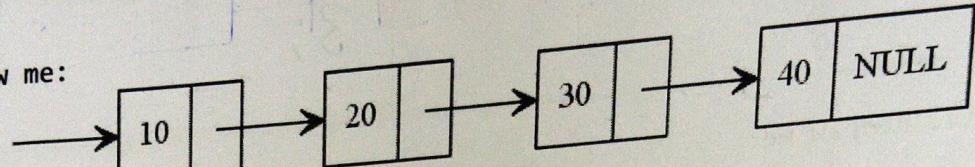


deleteBeg()

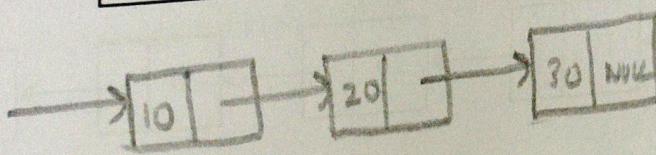


VIII. Redraw me:

Header L

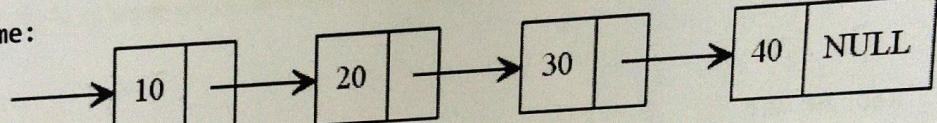


deleteEnd()

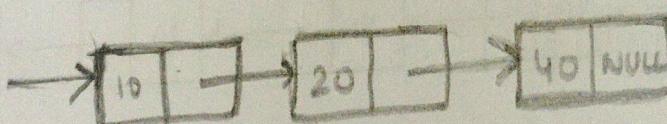


IX. Redraw me:

Header L

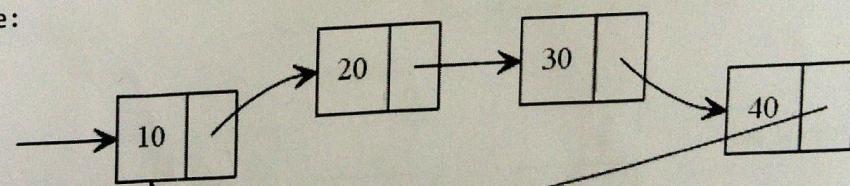


deleteMid(30)



X. Redraw me:

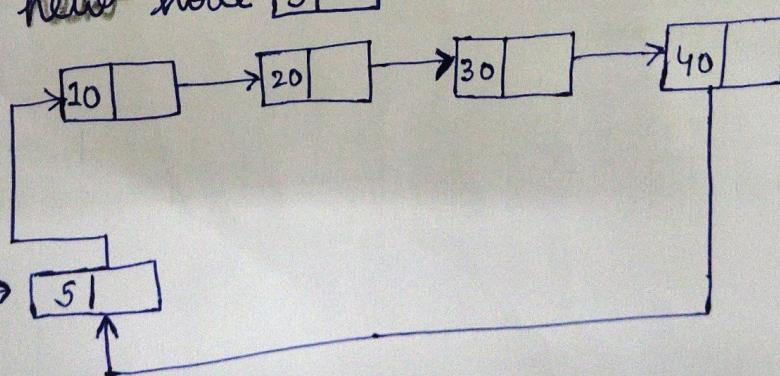
Header L



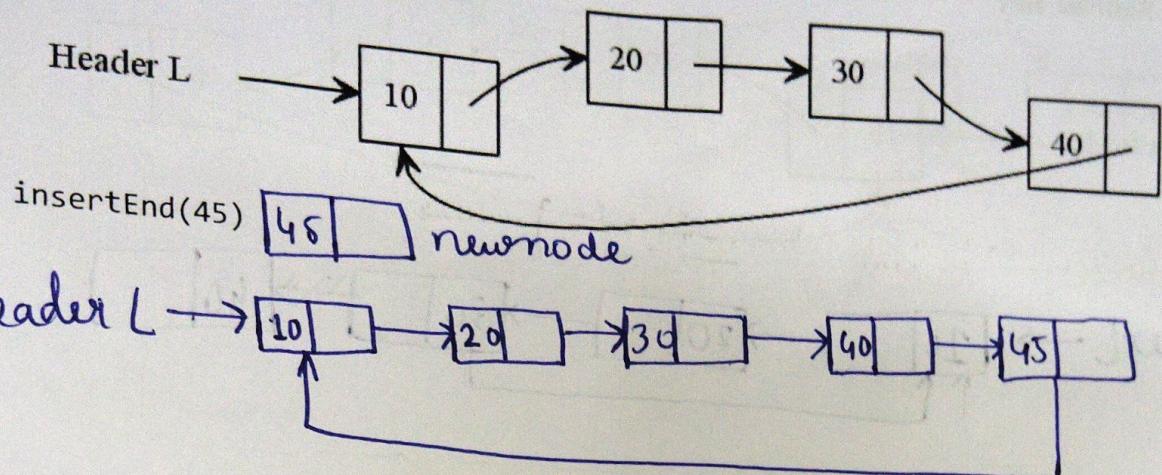
insertBeg(5)

new node 5

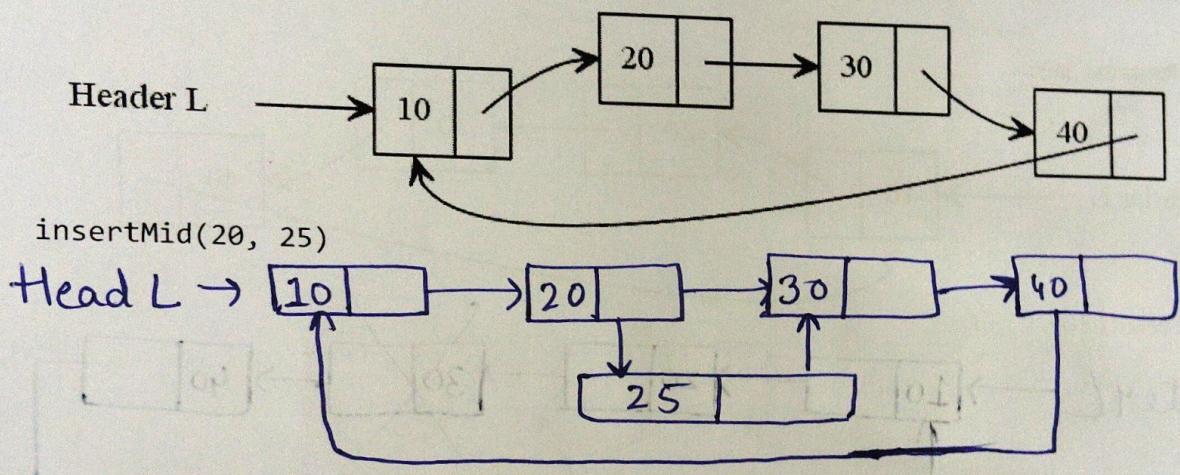
Header L -



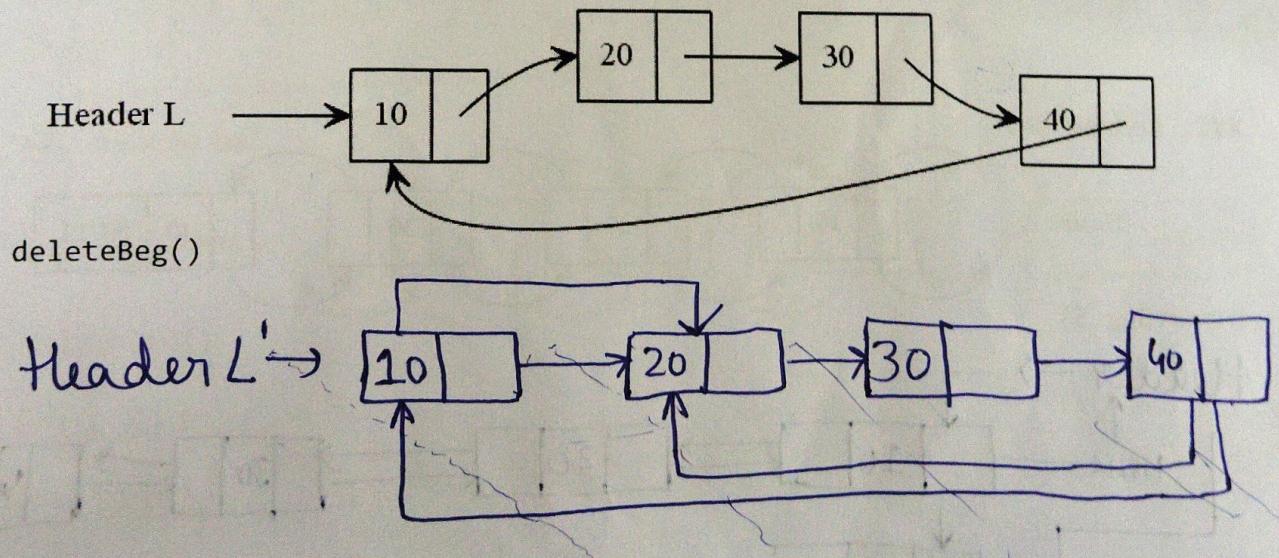
XI. Redraw me:



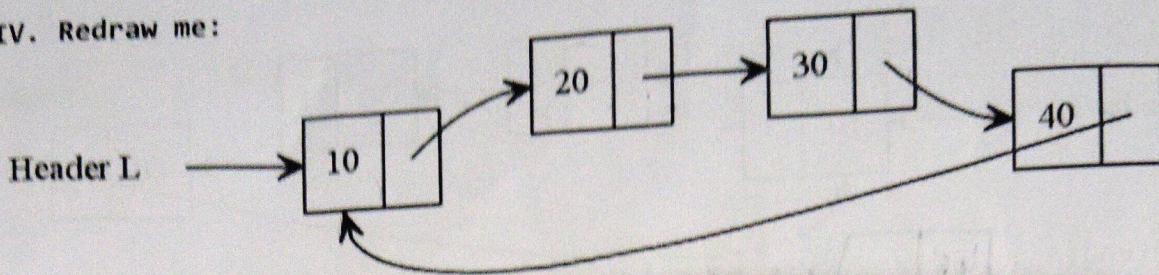
XII. Redraw me:



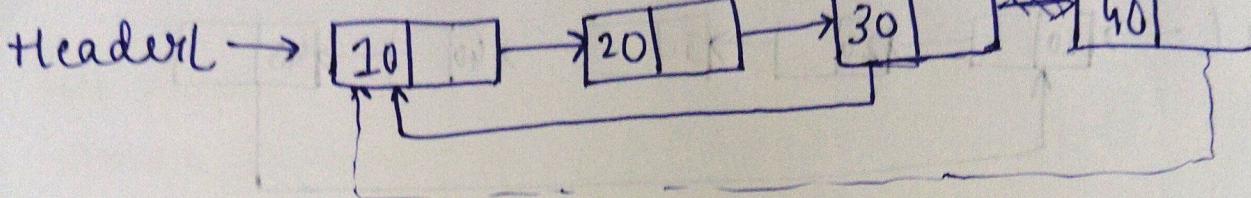
XIII. Redraw me:



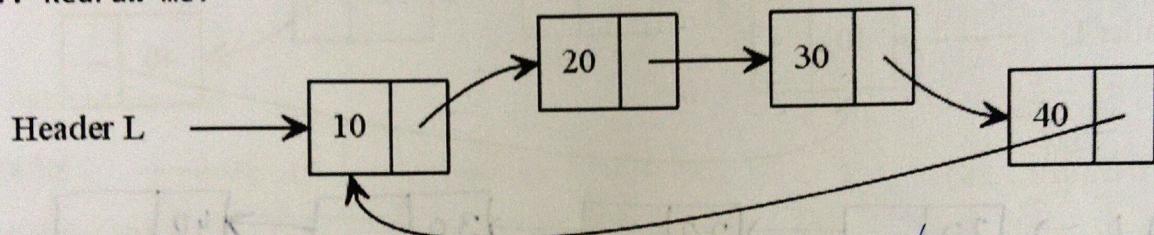
XIV. Redraw me:



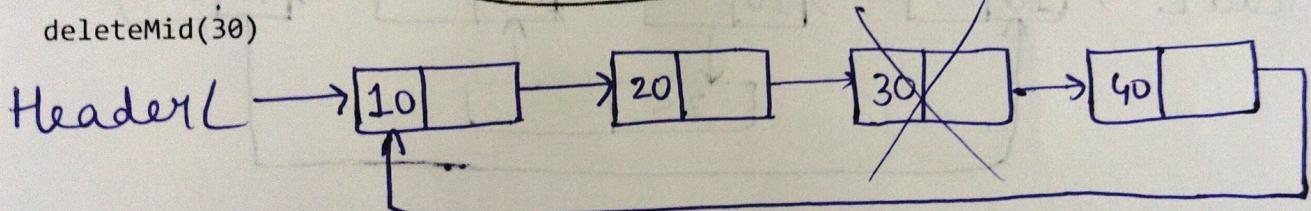
deleteEnd()



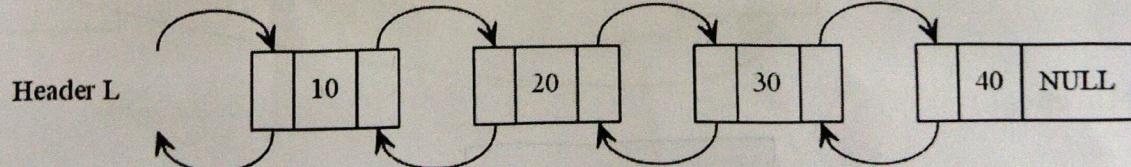
XV. Redraw me:



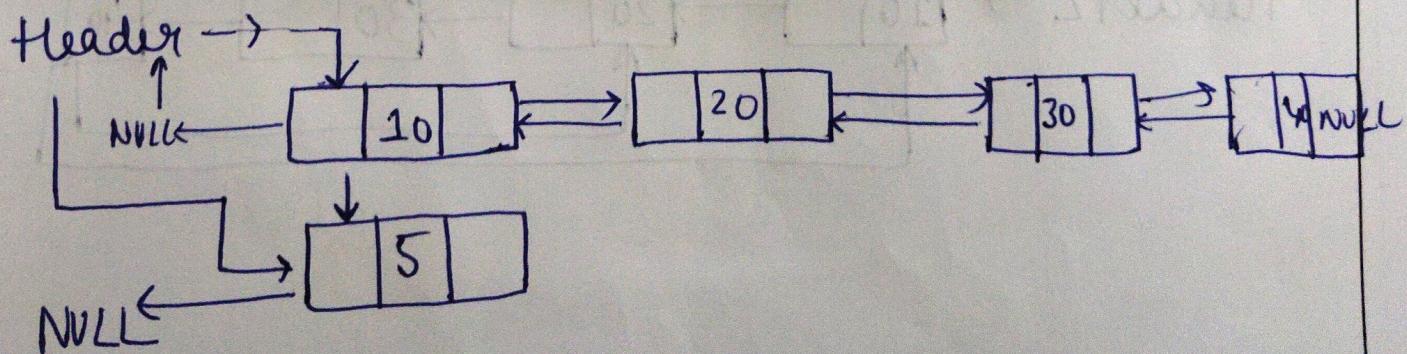
deleteMid(30)



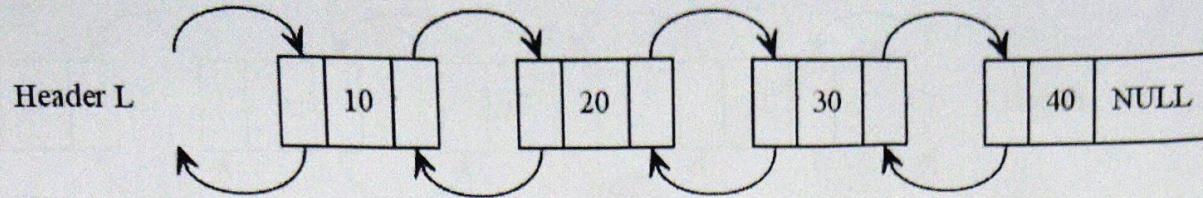
XVI. Redraw me:



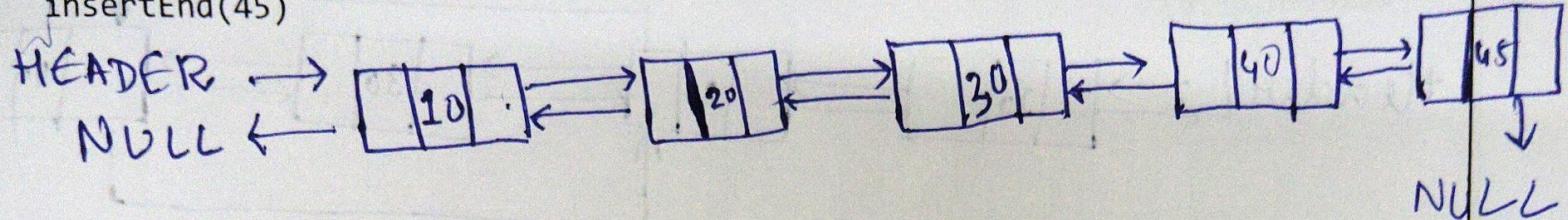
insertBeg(5)



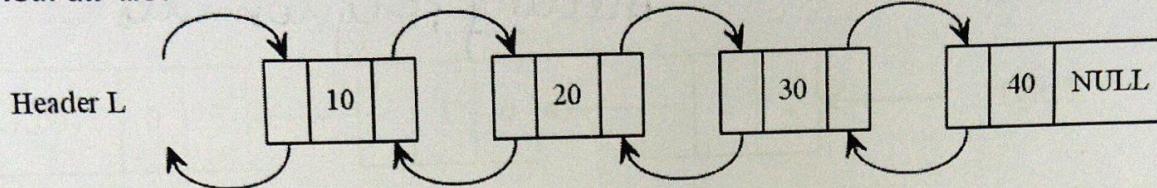
XVII. Redraw me:



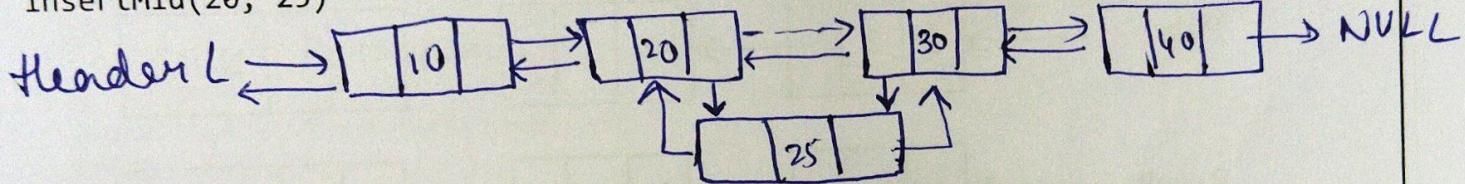
insertEnd(45)



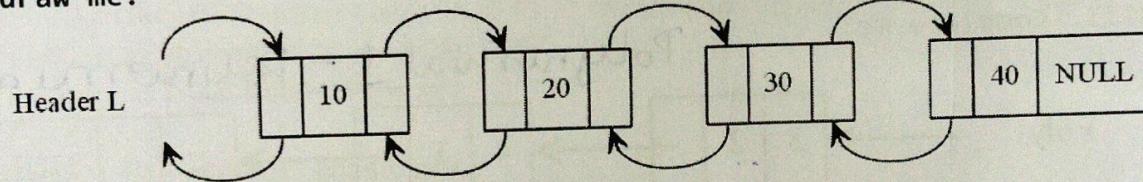
XVIII. Redraw me:



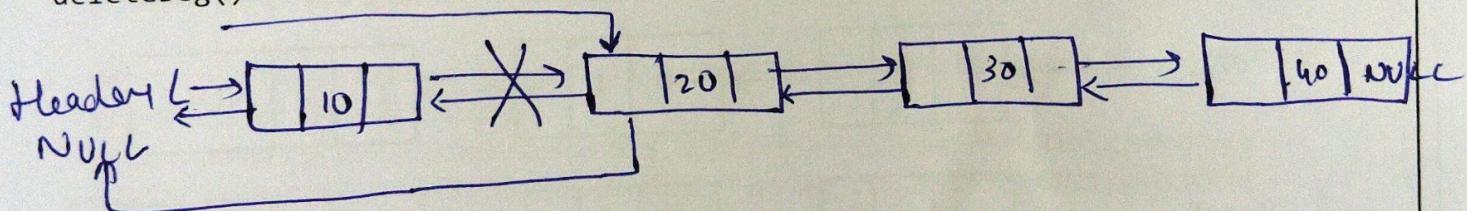
insertMid(20, 25)



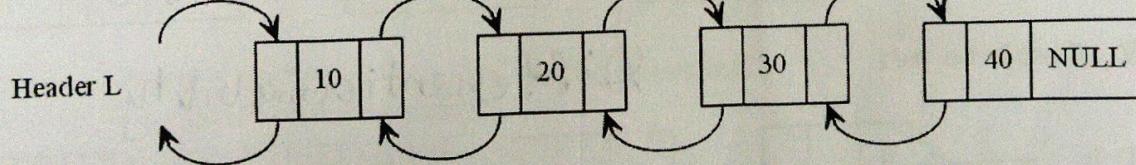
XIX. Redraw me:



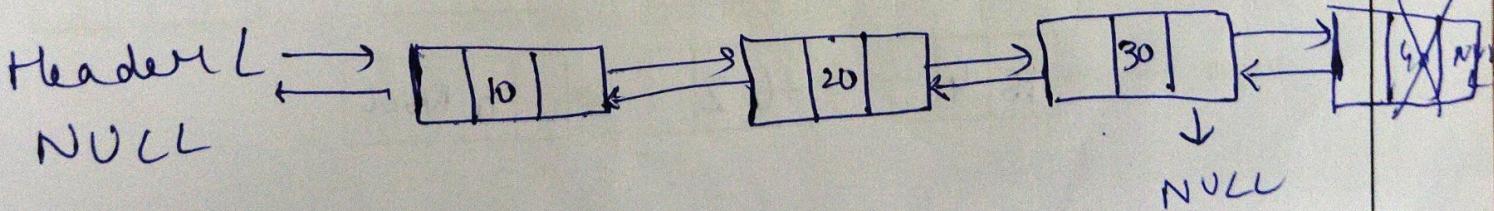
deleteBeg()



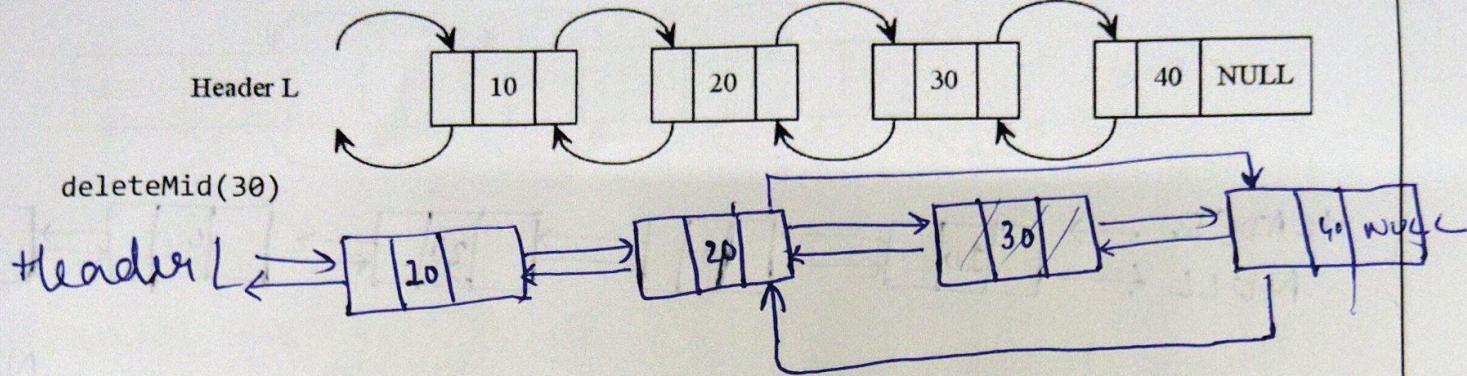
XX. Redraw me:



deleteEnd()

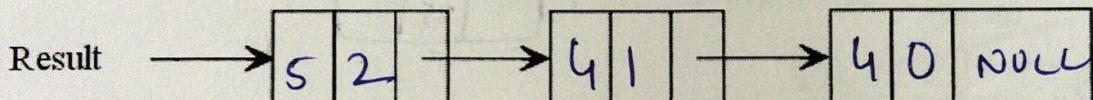
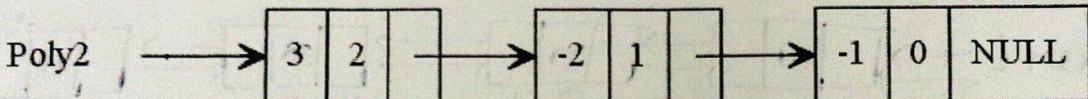
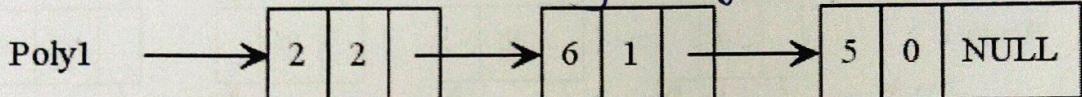


XXI. Redraw me:



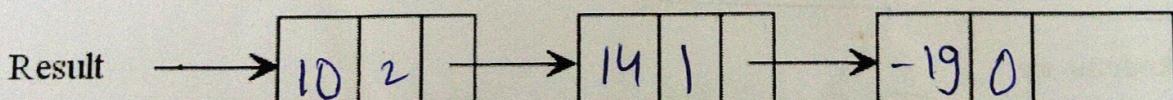
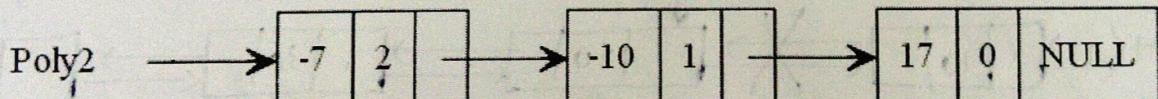
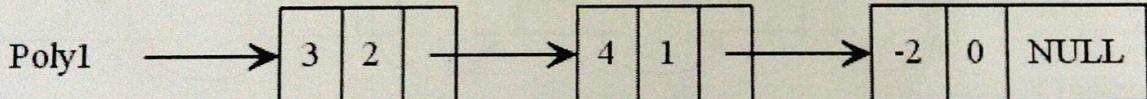
XXII. Complete me:

Adding Polynomials



XXIII. Complete me:

Polynomial 1 - Polynomial 2



XIV. Complete me:

Differentiability

