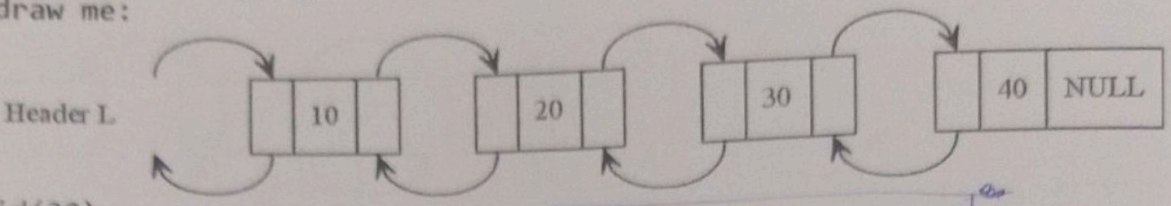
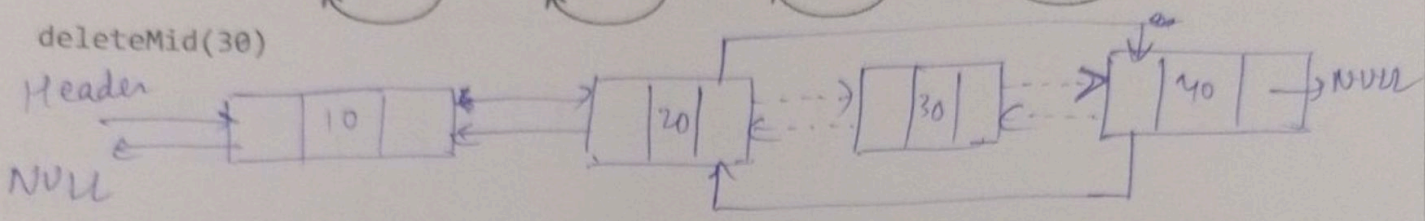
XXI. Redraw me:

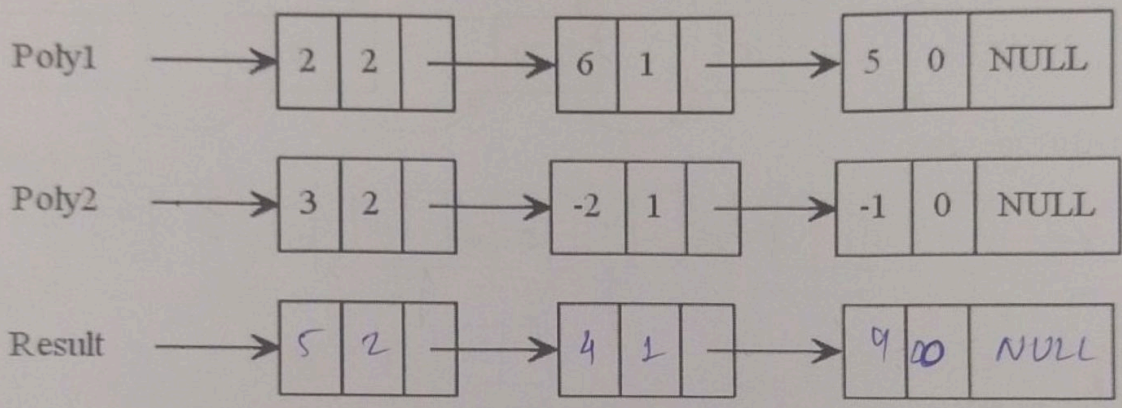


deleteMid(30)



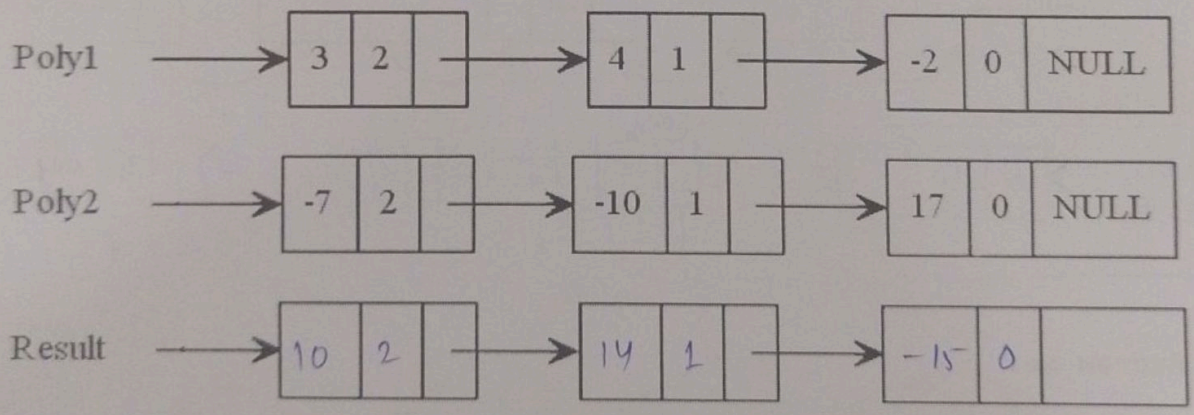
XXII. Complete me:

Adding Polynomials



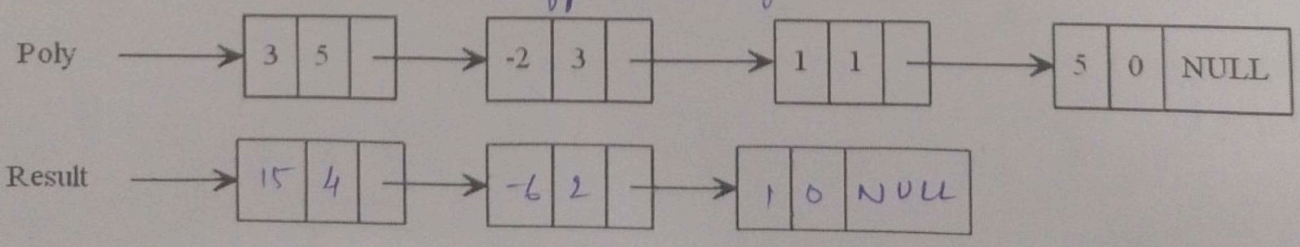
XXIII. Complete me:

Polynomial 1 - Polynomial 2

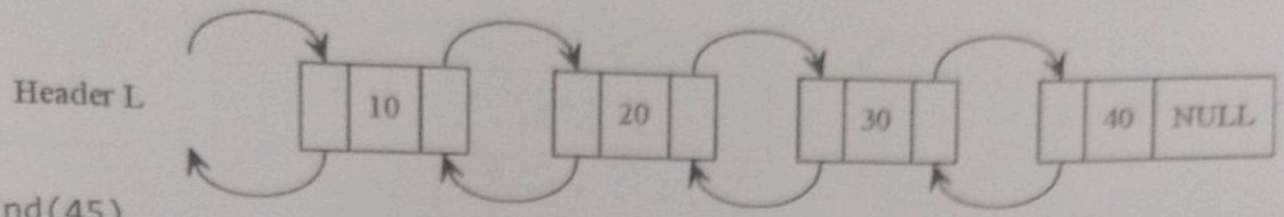


XIV. Complete me:

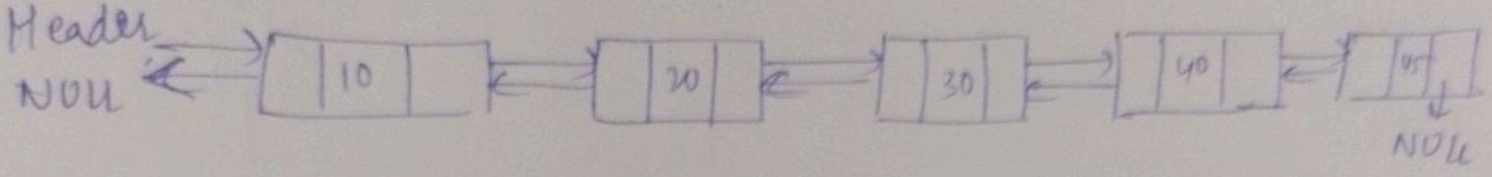
Differentiating



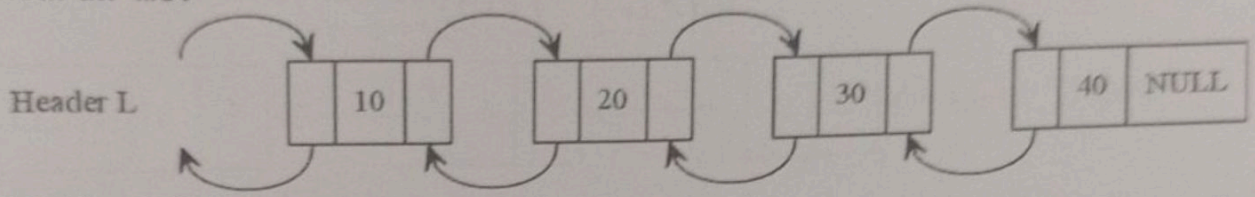
XVII. Redraw me:



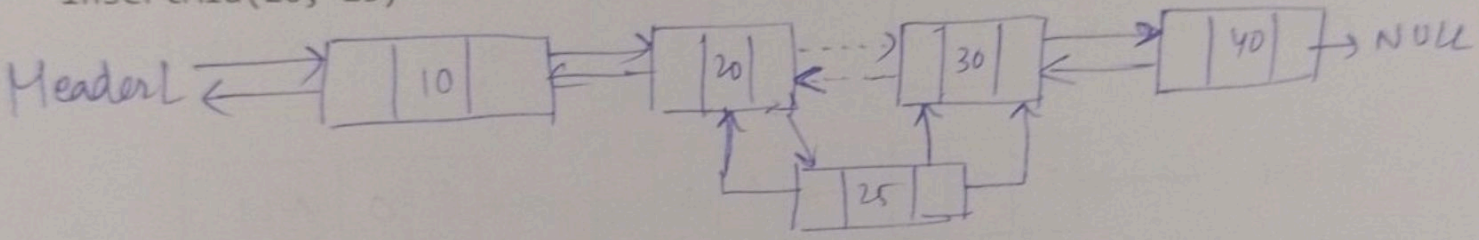
insertEnd(45)



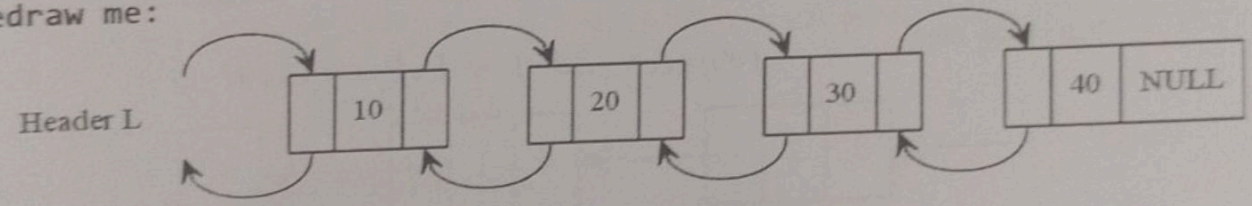
XVIII. Redraw me:



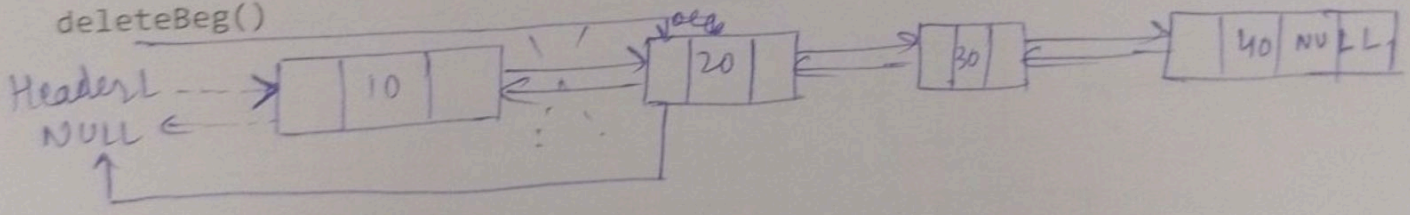
insertMid(20, 25)



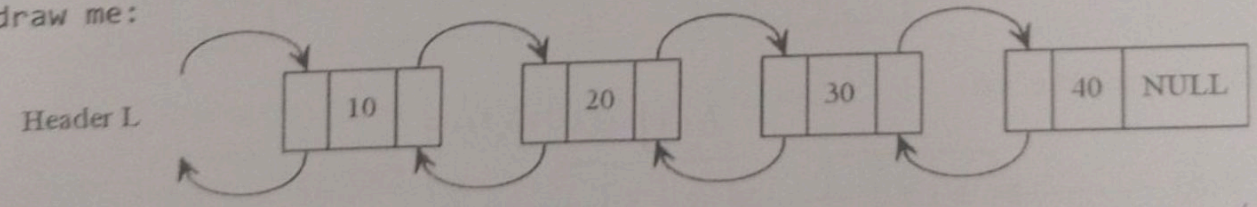
XIX. Redraw me:



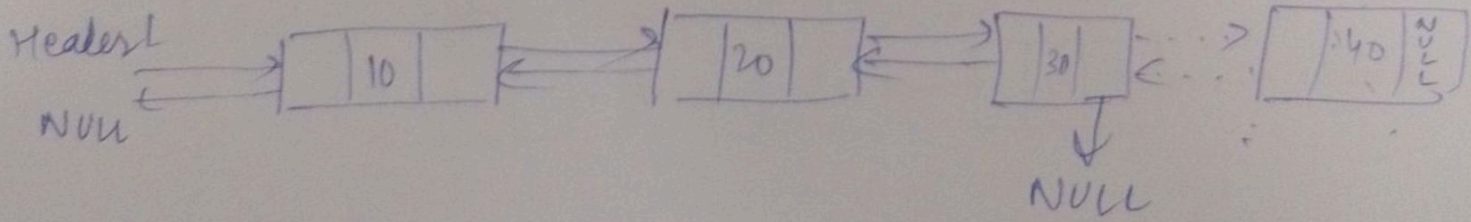
deleteBeg()



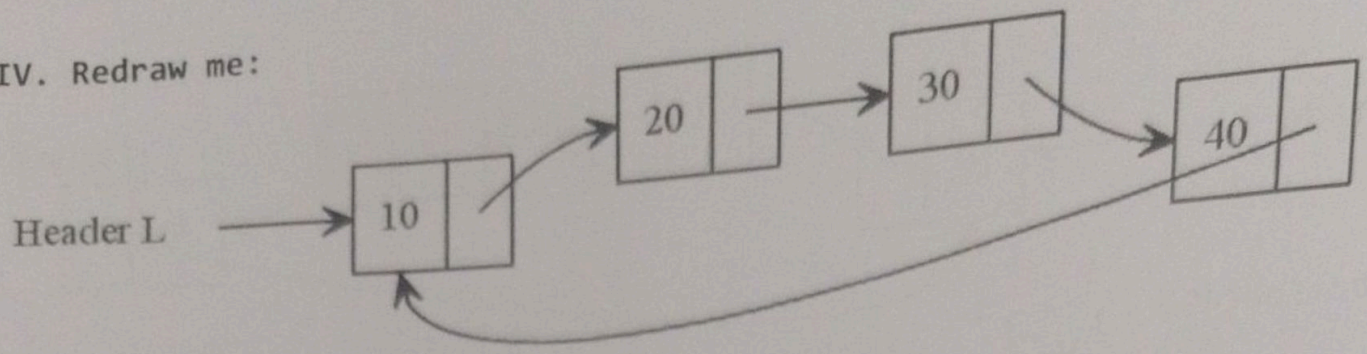
XX. Redraw me:



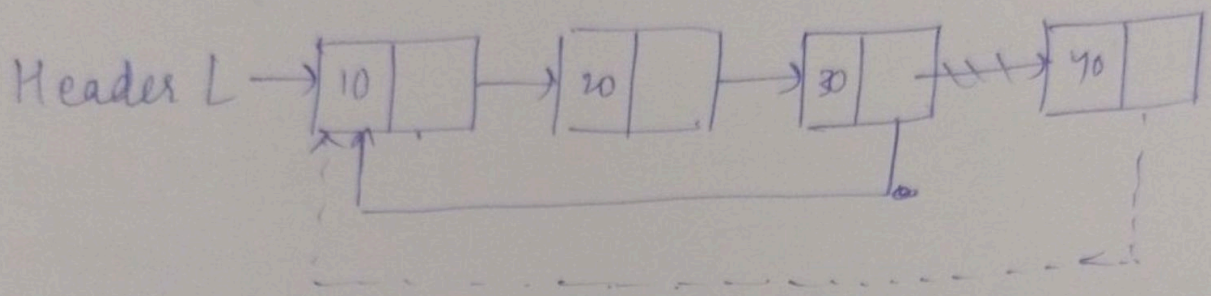
deleteEnd()



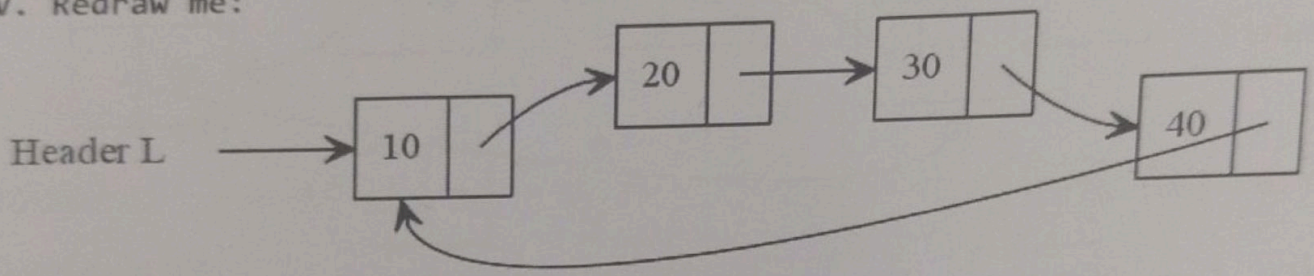
XIV. Redraw me:



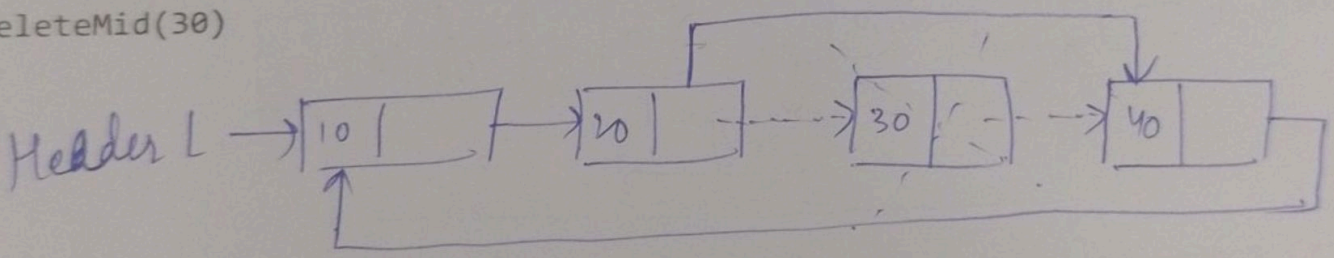
deleteEnd()



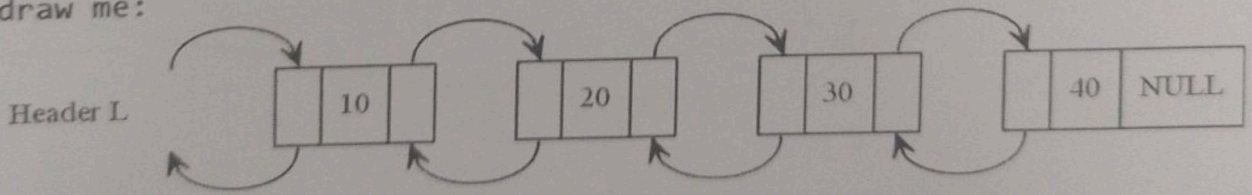
XV. Redraw me:



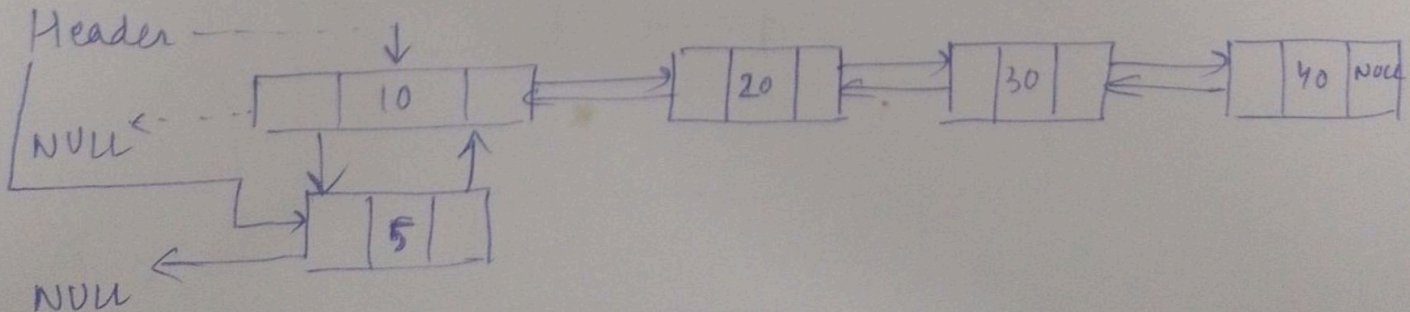
deleteMid(30)



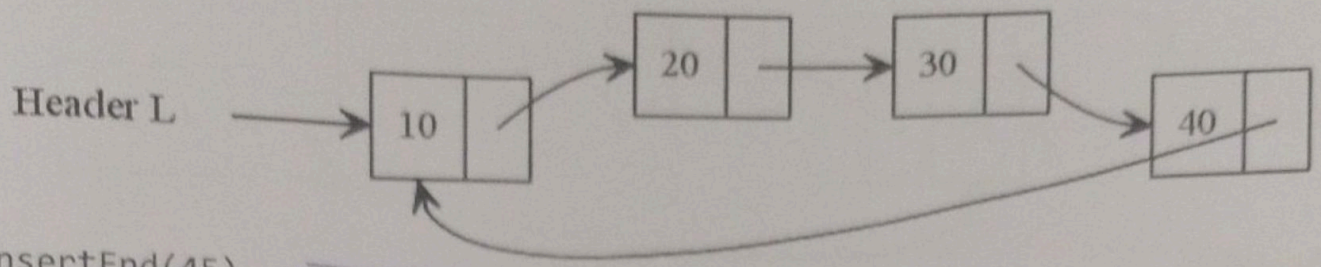
XVI. Redraw me:



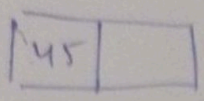
insertBeg(5)

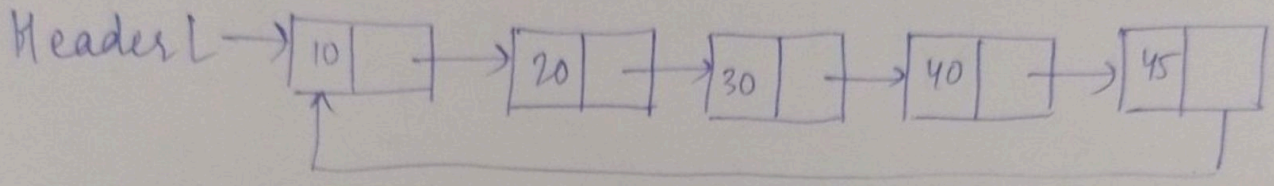


XI. Redraw me:

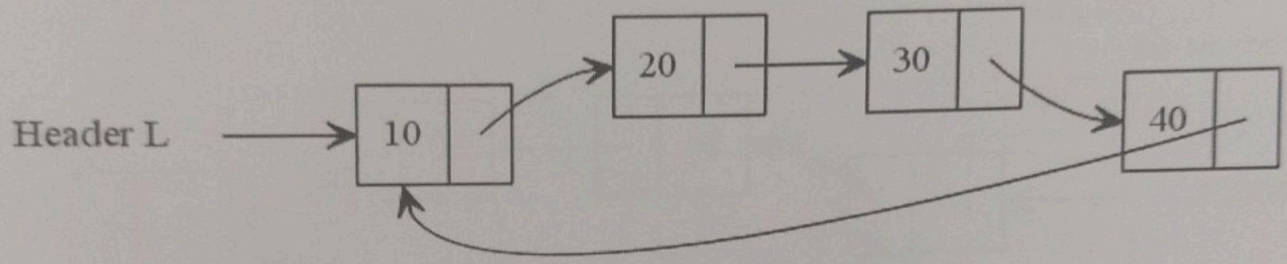


insertEnd(45)

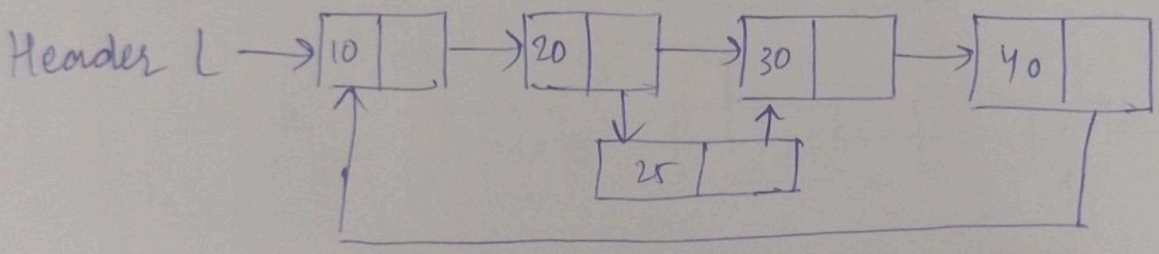




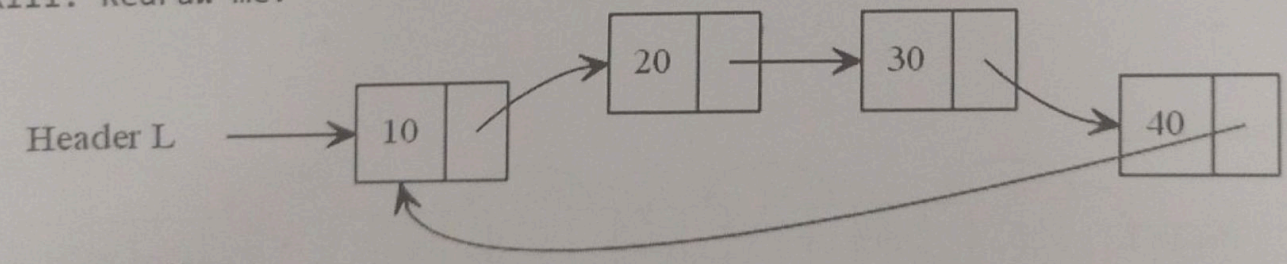
XII. Redraw me:



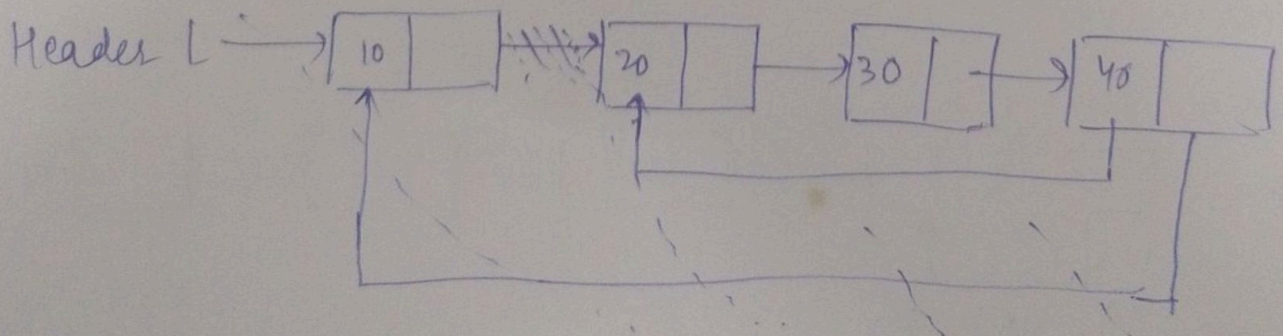
insertMid(20, 25)



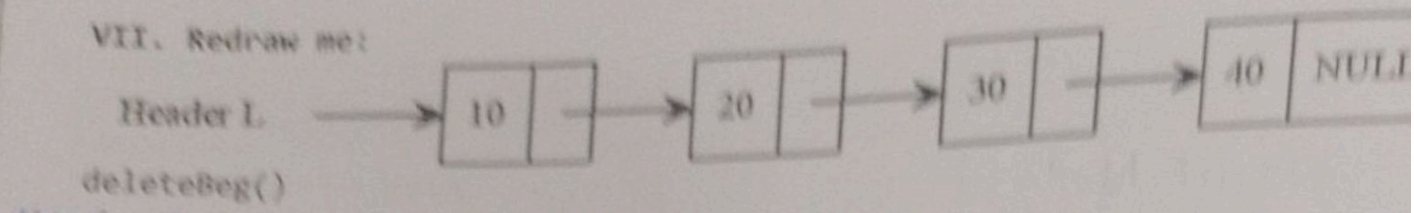
XIII. Redraw me:



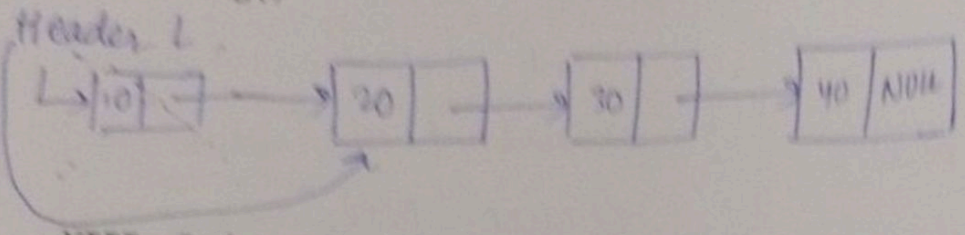
deleteBeg()



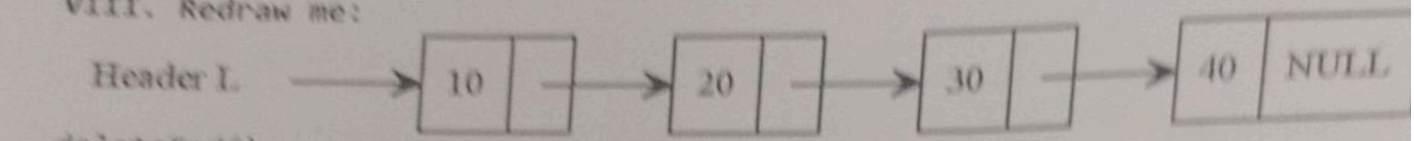
VII. Redraw me:



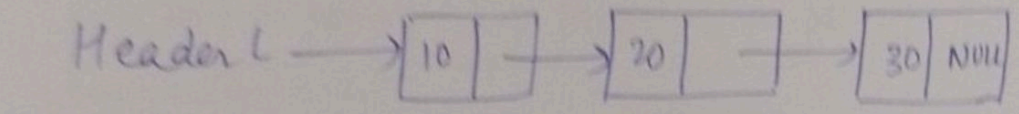
deleteBeg()



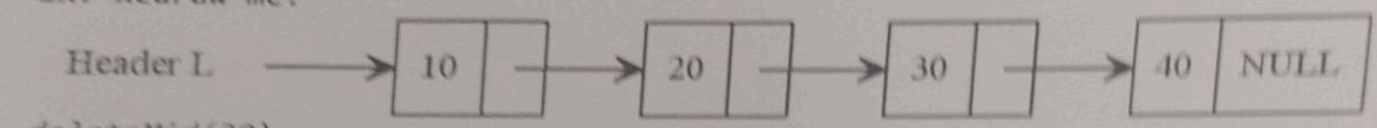
VIII. Redraw me:



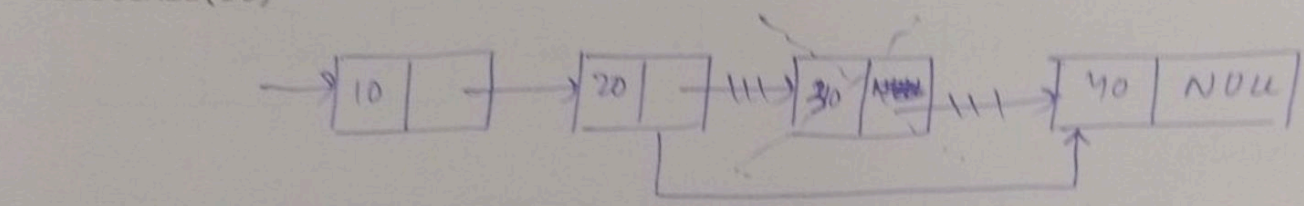
deleteEnd()



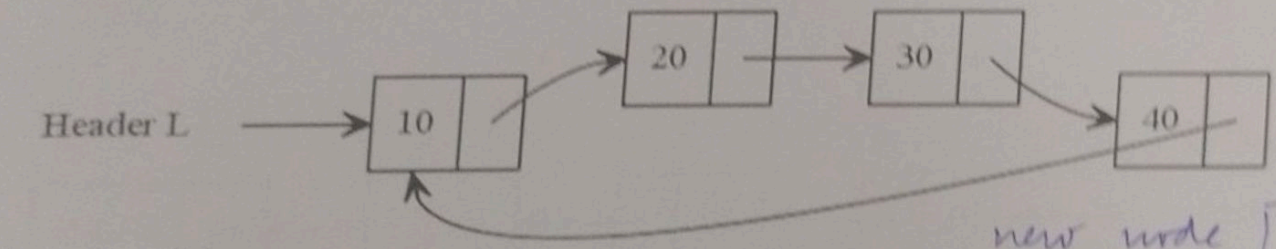
IX. Redraw me:



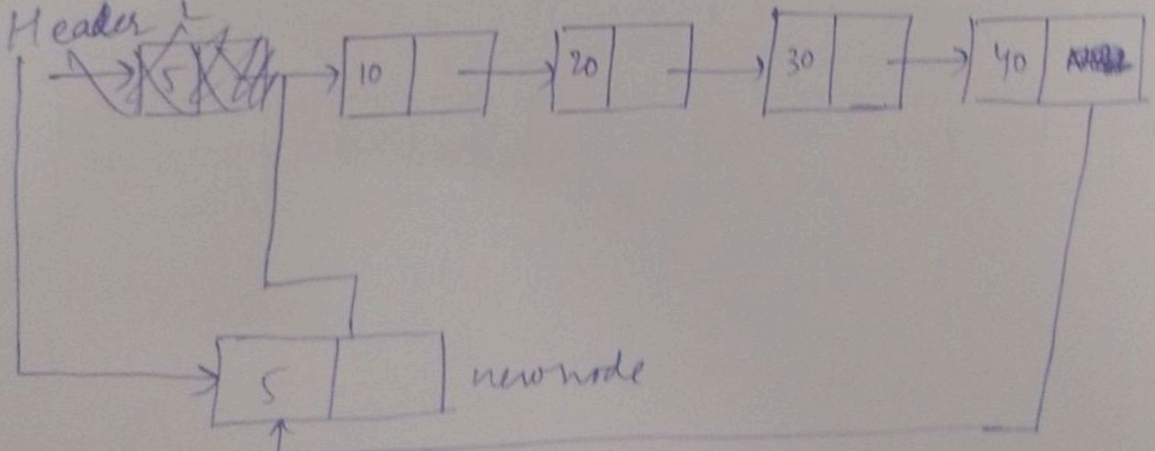
deleteMid(30)



X. Redraw me:



insertBeg(5)

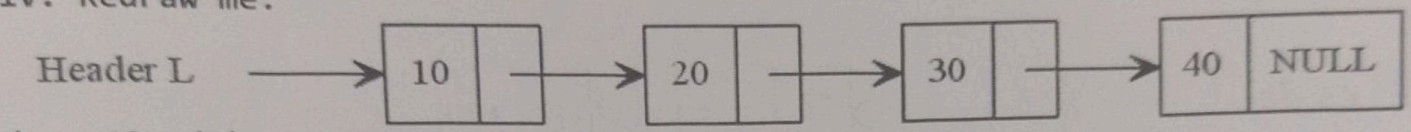


III. Declare me:

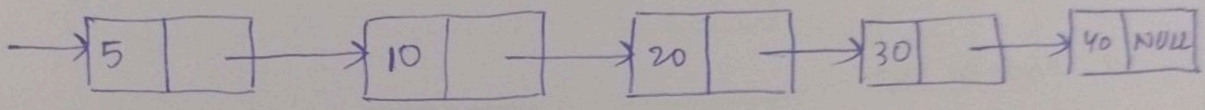
Singly Linked List	Doubly Linked List

struct Node {
int data;
struct Node* next;
};
	struct Node {
	int data;
	struct Node* prev;
	struct Node* next;
	};

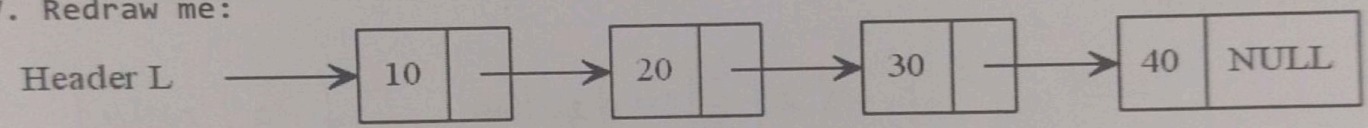
IV. Redraw me:



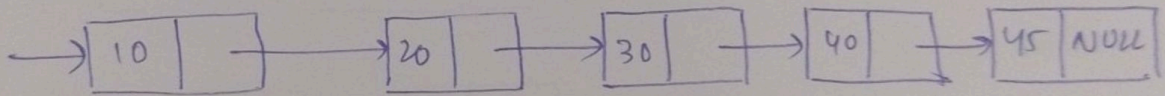
insertBeg(5)



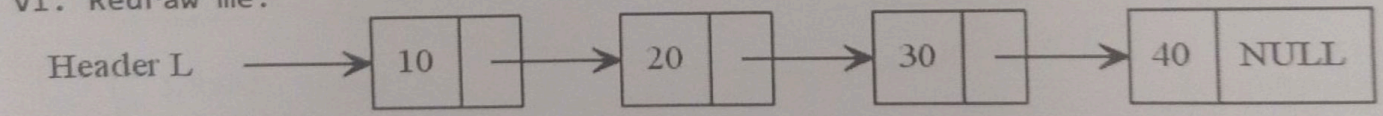
V. Redraw me:



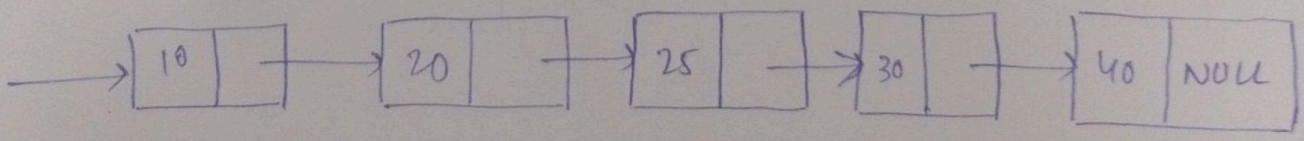
insertEnd(45)



VI. Redraw me:



insertMid(20, 25)



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

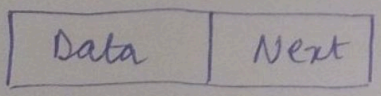
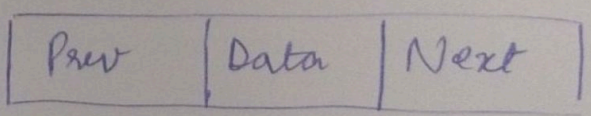
9. Which of the following is the fastest and easiest sorting technique?
   (a) Bubble
   (b) Quick
   (c) Insertion
   (d) Bucket

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.	c	2.	d	3.	a	4.	b	5.	c	6.	a	7.	b	8.	a	9.	b	10.	b

II. Draw my node:

Singly Linked List	Doubly Linked List
	

DATA STRUCTURES AND ALGORITHMS
Unit-II-Assignment

Reg. No. : RA2311026010654 Name : Surabhi Krishna

Year : 2 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.

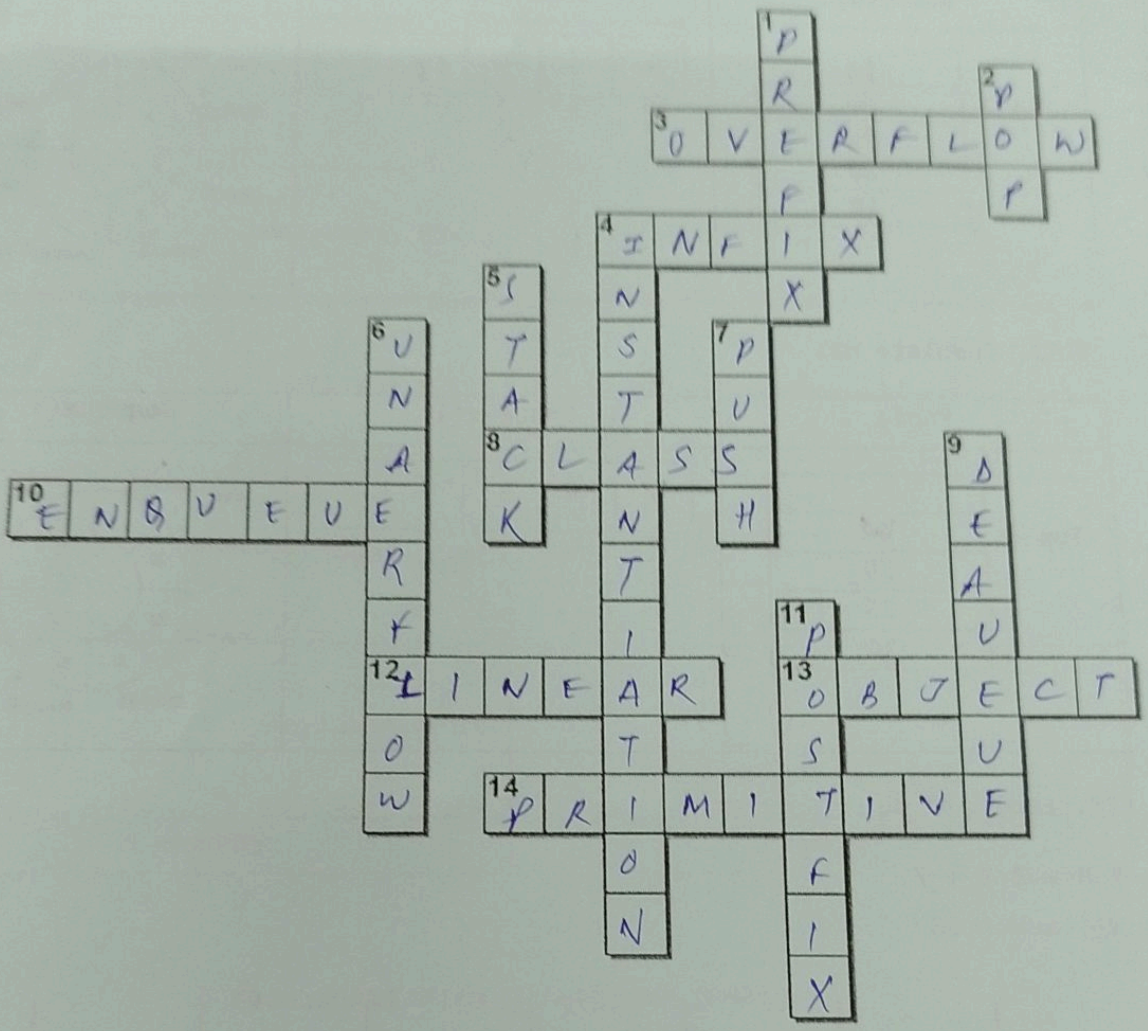
2. Which node pointers should be updated if a new node B is to be inserted in the middle of A and C nodes of a singly linked list?
   (a) NEXT pointer of A and NEXT pointer of C
   (b)NEXT pointer of B and NEXT pointer of C
   (c)NEXT pointer of B
   (d)NEXT pointer of A and NEXT pointer of B

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.

X. Fill me:



Across:

3 Attempt to insert an element when the stack is full is said to be _____.

4 In _____ notation, the arithmetic operator appears between the two operands to which it is being applied.

8 _____ is a blueprint or template for the object.

10 The process of inserting a new element on to the rear of the queue is called _____ operation.

12 In _____ data structures, all the data elements are arranged in a sequential fashion.

13 _____ is simply a collection of data (variables) and methods (functions) that act on those data.

14 _____ data structures include all the fundamental data structures that can be directly manipulated by machine level instructions.

Down:

1 In _____ notation, the arithmetic operator is placed before the two operands to which it applies.

2 The process of deleting an element from the top of stack is called _____ operation.

4 An object is also called an instance of a class and the process of creating this object is called _____.

5 A _____ is a list with the restriction that insertions and deletions can be performed in only one position, namely, the end of the list, called the top. It follows Last-In-First-Out (LIFO) principle.

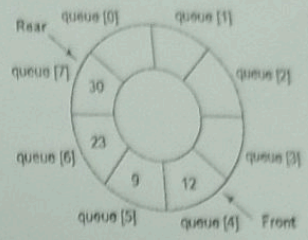
6 Attempt to delete an element when the stack is empty is said to be _____.

7 The process of inserting a new element to the top of the stack is called _____ operation.

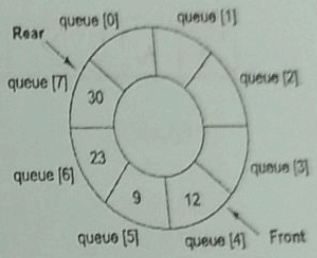
9 The process of deleting an element from the front of queue is called _____ operation.

11 In _____ notation, the arithmetic operator appears directly after the two operands to which it applies.

VII. Complete me:

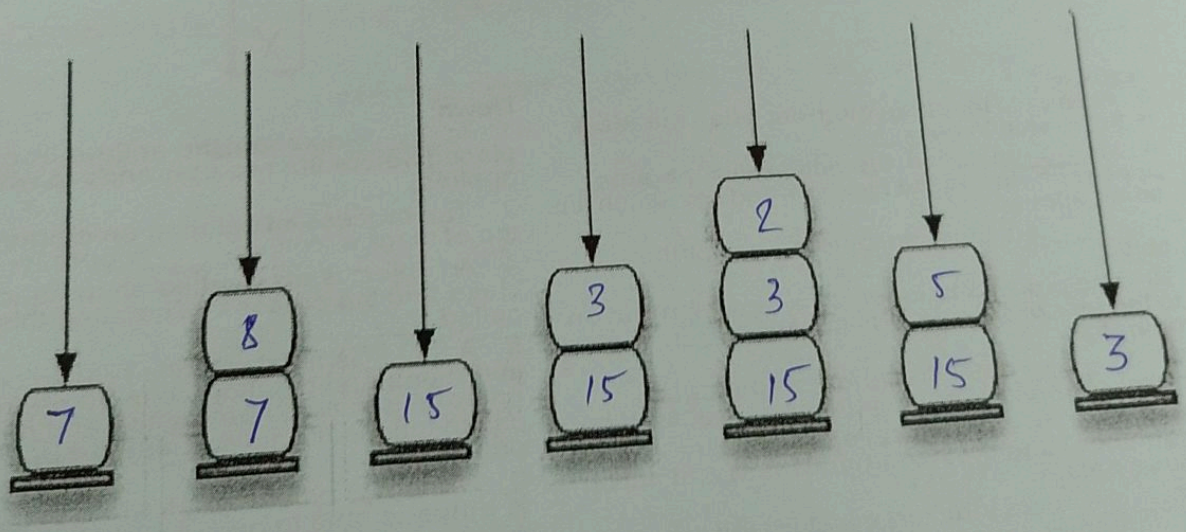
Push(50)	Enqueue(50)	Enqueue(50)
Top → 50, 40, 30, 20, 10	10, 20, 30, 50 (0 ↑F, 2 ↑R)	

VIII. Complete me:

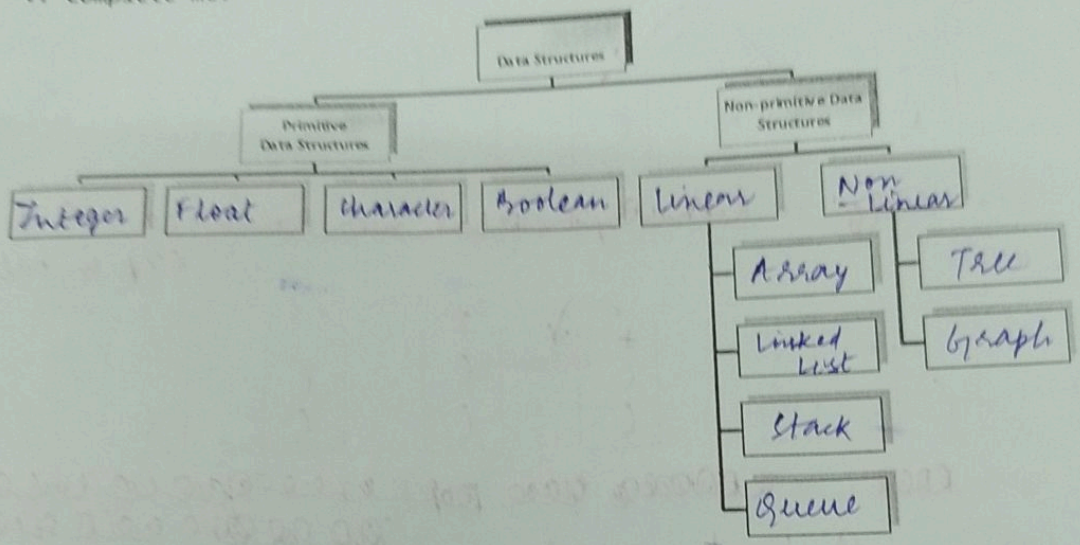
Pop()	Dequeue()	Dequeue()
Top → ~~40~~, 30, 20, 10	~~10~~, 20, 30 (0 ↑F, 2 ↑R)	

IX. Evaluate me:

7 8 + 3 2 + /



V. Complete me:



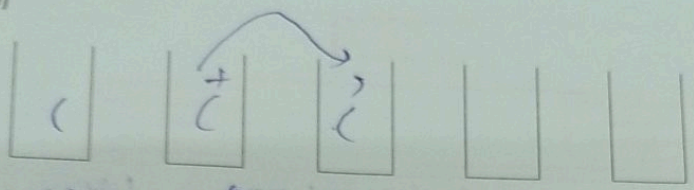
Classification of Data Structures

VI. Match me:

Column A	Column B
Top → 50, 40, 30, 20, 10	Queue Overflow
Top →	Queue Underflow
10 20 30 40 50 (0 1 2 3 4; F, R)	Stack Overflow
None (0 1 2 3 4; F R)	Stack Underflow

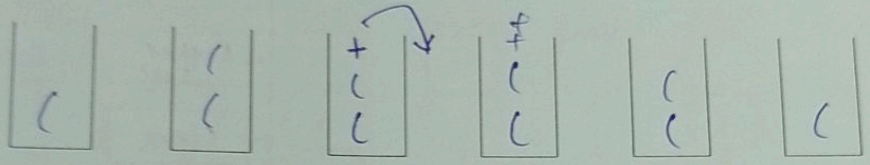
IV. Balance me:

(a+b)



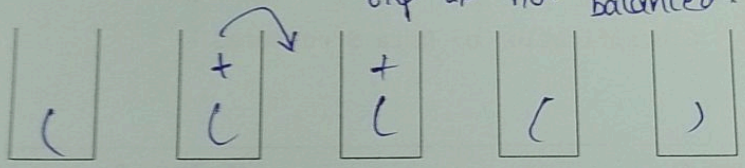
Stack is ~~empty~~ → ~~Each parenthesis are balanced~~ a b + exp is balanced

((a+b)



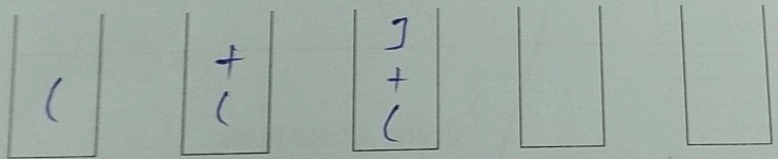
Stack is ~~Push~~ → ~~Here opening and closing~~ ~~are~~ ~~not~~ ~~balanced~~

(a+b)) a b + Exp is not balanced.



Stack is ~~Push~~ → ~~Here opening and closing~~ ~~are~~ ~~not~~ ~~balanced~~

(a+b] a b +



Not corresponding to the ~~closing parenthesis~~ because

+ a b (Unbalanced) ~~there is~~ corresponding opening ~~of~~ ~~the~~ last bracket

19. If a delete operation is performed on an empty queue, then which of the following situations will occur?
(a) Overflow
(b) Underflow
(c) Array out of bound
(d) None of the above

20. Which of the following is not a queue application?
(a) Recursion control
(b) CPU scheduling
(c) Message queuing
(d) All of the above are queue applications

ANSWERS

1.	d	2.	c	3.	b	4.	a	5.	c	6.	d	7.	a	8.	d	9.	b	10.	c
11.	b	12.	a	13.	b	14.	d	15.	a	16.	b	17.	a	18.	b	19.	b	20.	a

II. Place me in the basket:

Stack	Queue	Double Ended Queue	Notations
Push	Enqueue	Enqueue Front	Infix
Pop	Dequeue	Enqueue Rear	Prefix
Peek		Dequeue Front	Postfix
		Dequeue Rear	

Following words are to be placed in the relevant basket:

Push
Pop
Peek
Enqueue
Dequeue
DequeueFront
Infix
DequeueRear
EnqueueRear
EnqueueFront
Prefix
Postfix

III. Match me:

Column A	Column B
Infix	+/ABC
Prefix	AB/C+
Postfix	A/B+C

13. CPU scheduler can be implemented by which of the following datastructures?
(a)Stack
(b)Queue
(c)Graph
(d)Tree

14. Which of the following is a type of a queue?
(a)Circular queue
(b)Priority queue
(c)Double-ended queue
(d)All of the above

15. If 1, 2, 3, 4 are the queue contents with element 1 at the front and 4 at the rear, then what will be the queue contents after following operations:
Insert (5)
Delete ( )
Delete ( )
Delete ( )
Insert (6)
Insert (−1)
Delete ( )
(a)5, 6, −1
(b)4, 5, 6, −1
(c)1, 2, 6
(d)1, 2, 6, −1

16. Which of the following is best suitable for implementing a print scheduler?
(a)Stack
(b)Queue
(c)Array
(d)None of the above

17. If 'front' points at the front end of the queue, 'rear' points at the rear end of the queue and 'queue []' is the array containing queue elements, then which of the following statements correctly reflects the insert operation for inserting 'item' into the queue?
(a)rear = rear + 1; queue [rear] = item;
(b)front = front + 1; queue [front] = item;
(c)queue [rear++] = item;
(d)Both (a) and (c) are correct

18. If 'front' points at the front end of the queue, 'rear' points at the rear end of the queue and 'queue []' is the array containing queue elements, then which of the following statements correctly reflects the delete operation for deleting an element from the queue?
(a)item = queue [rear]; rear = rear + 1;
(b)item = queue [front]; front = front + 1;
(c)item = queue [++front];
(d)Both (b) and (c) are correct

6. Which of the following is best suitable for storing a simple collection of employee records?
   (a)Stack
   (b) Queue
   (c)Array
   (d)None of the above

7. If 'top' points at the top of the stack and 'stack []' is the array containing stack elements, then which of the following statements correctly reflect the Push Operation for inserting 'item' into the stack?
   (a)top = top + 1; stack [top] = item;
   (b)stack [top] = item; top = top + 1;
   (c)stack [top++] = item;
   (d)Both (a) and (c) are correct

8. If 'top' points at the top of the stack and 'stack []' is the array containing stack elements, then which of the following statements correctly reflect the pop operation?
   (a)top = top – 1; item = stack [top];
   (b)item = stack [top]; top = top – 1;
   (c)item = stack [--top];
   (d)Both (b) and (c) are correct

9. If a pop operation is performed on an empty stack, then which of the following situations will occur?
   (a) Overflow
   (b)Underflow
   (c)Array out of bound
   (d)None of the above

10. Which of the following is not a stack application?
   (a)Recursion control
   (b) Expression evaluation
   (c)Message queuing
   (d)All of the above are stack applications

11. Which of the following statements is not true for queues?
   (a)It is a linear data structure.
   (b)It allows insertion/deletion of elements only at one end.
   (c)It has two ends front and rear.
   (d)It is based on First-In-First-Out principle.

12. Which of the following statements is not an example of a queue?
   (a)Collection of tiles one over another.
   (b) A queue of print jobs.
   (c) A line up of people waiting for the bus at the bus stop.
   (d)All of the above are queue examples.

DATA STRUCTURES AND ALGORITHMS
Unit-III-Assignment

Reg. No. : RA2311026010654 Name : Surabhi Krishna

Year : II ND Branch: CSE AI-ML Section: AA-2

I. Choose the best answer:

1. Which of the following is not true for stacks?
   (a)It is a linear data structure.
   (b) It allows insertion/deletion of elements only at one end
   (c) It is widely used by systems processes, such as compilation and program control
   (d)It is based on First-In-First-Out principle

2. Which of the following is not an example of a stack?
   (a)Collection of tiles one over another
   (b) A set of bangles worn by a lady on her arm
   (c) A line up of people waiting for the bus at the bus stop
   (d)A pileup of boxes in a warehouse one over another

3. Tower of Hanoi can be regarded as a problem of which of the following data structures?
   (a)Stack
   (b) Queue
   (c)Graph
   (d)Tree

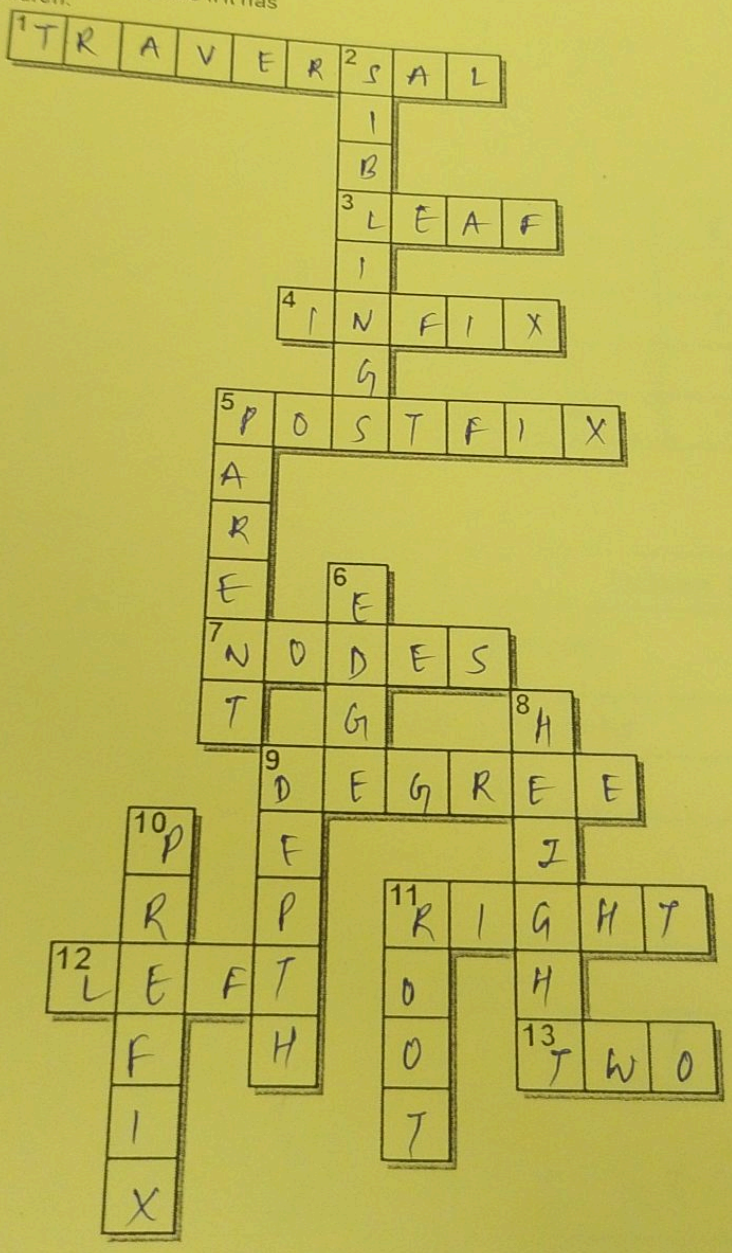
4. Recursive function calls are executed using which of the following data structures?
   (a)Stack
   (b) Queue
   (c)Graph
   (d)Tree

5. If 2, 1, 5, 8 are the stack contents with element 2 being at the top of the stack, then what will be the stack contents after following operations:
   Push (11)
   Pop ( )
   Pop ( )
   Pop ( )
   Push(7)
   (a)11, 2, 1
   (b)8, 11, 7
   (c)7, 5, 8
   (d)5, 8, 7

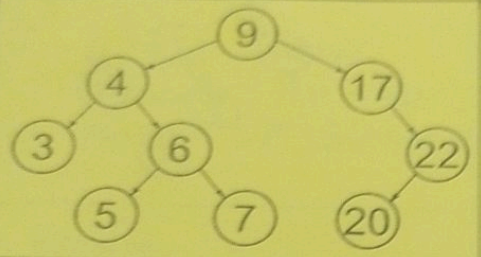
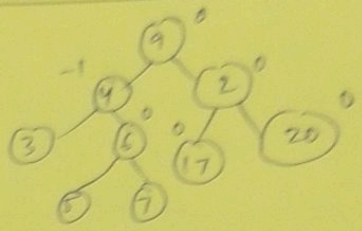
III. Fill me:

Across:
1 The process of visiting all nodes of a tree is called tree _____.
3 Nodes with no children are known as _____.
4 The inorder traversal of the binary tree for an arithmetic expression gives the expression in an _____ form.
5 The postorder traversal of the binary tree for the given expression gives in _____ form.
7 A tree is a collection of _____.
9 The number of subtrees of a node is called its _____.
11 In binary search tree, to perform a findMax, start at the root and go right as long as there is a _____ child.
12 In binary search tree, to perform findMin, start at the root and go left as long as there is a _____ child.
13 A tree is said to be a binary tree if it has atmost _____ children.

Down:
2 Nodes with the same parent are called _____.
5 In a tree, every node except the root has one _____.
6 An _____ refers to the link from parent to child.
8 The _____ of ni is the length of the longest path from ni to a leaf.
9 For any node ni, the _____ of ni is the length of the unique path from the root to ni.
10 The preorder traversal of the binary tree for the given expression gives in _____ form.
11 The _____ of a tree is the node with no parents.



XVI. Redraw me:

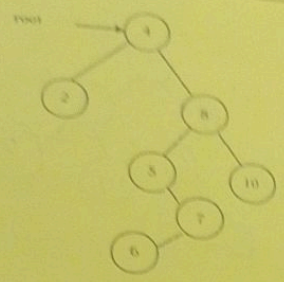
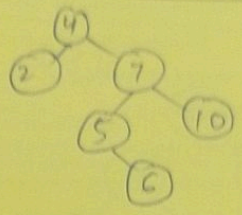
Binary Tree	Find Balance Factor
	

XVII. Search me:

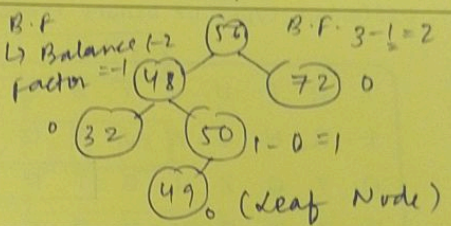
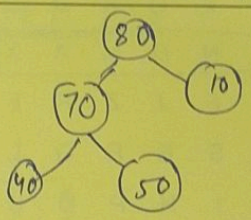
1. Tree traversals
2. Binary search tree operations
3. Different types of binary trees
4. Representation of binary tree

F	D	P	U	J	Y	S	G	E	R	Z	O	K	J	N	X	I	J
L	R	N	U	W	J	R	Q	E	N	U	E	A	F	B	D	P	X
Q	Y	C	Z	I	B	T	R	E	S	N	I	Z	L	M	G	N	Q
O	Z	E	O	K	D	Q	Q	D	Y	H	V	L	W	P	O	X	E
F	I	N	D	M	A	X	Z	S	P	T	U	A	O	J	R	U	X
V	A	F	J	Z	P	E	L	W	Y	F	A	C	B	E	A	R	B
J	K	S	E	E	O	L	O	U	J	E	P	Z	D	R	C	S	U
E	H	I	H	P	D	U	E	W	Y	Q	M	R	K	E	L	G	E
G	M	Y	D	U	N	B	G	T	J	O	O	C	D	D	S	G	J
F	R	S	T	R	I	C	T	J	E	E	S	N	M	R	E	A	S
Z	D	S	Y	E	F	S	Y	R	R	U	T	I	O	O	N	S	D
M	E	B	C	D	O	W	Z	P	L	J	V	M	R	T	M	F	E
D	L	F	U	R	C	H	N	Z	N	W	N	D	E	S	M	X	K
N	E	W	B	O	E	Q	L	U	T	Z	T	N	U	O	Z	Y	N
H	T	A	O	N	P	X	R	M	T	B	K	I	M	P	B	B	I
U	E	I	B	I	X	R	G	H	I	D	O	F	D	X	N	S	L
B	X	P	N	I	Y	C	O	E	L	I	N	E	A	R	C	X	B
Q	R	C	Z	K	X	P	M	Q	H	M	E	P	V	H	J	K	A

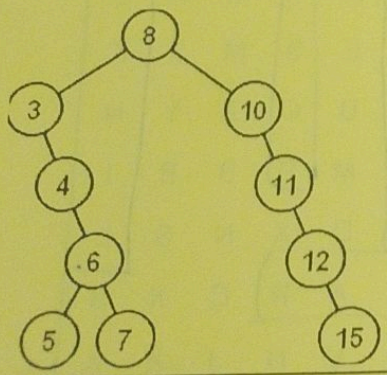
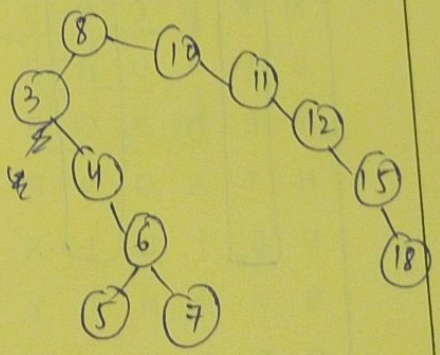
XIII. Redraw me:

Binary Search Tree	delete(8)
	

XIV. Draw me:

Concept	Description	Example
AVL Tree	Binary search tree but with additional feature where tree should be balanced.	
Max Haep	Here, the root has the maximum value and node is greater than child node.	

XV. Redraw me:

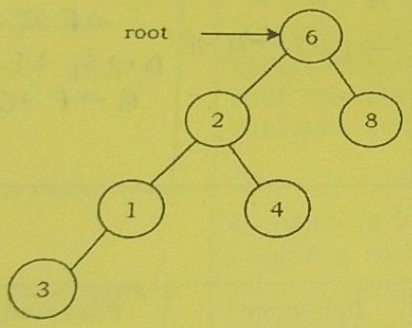
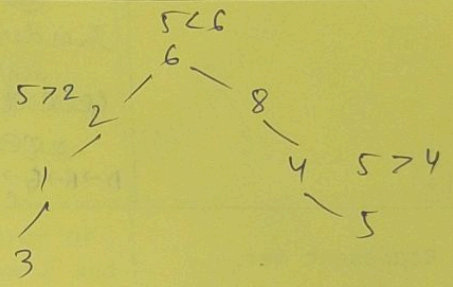
BST Tree	insert(19)
	

X. Fill my routine:

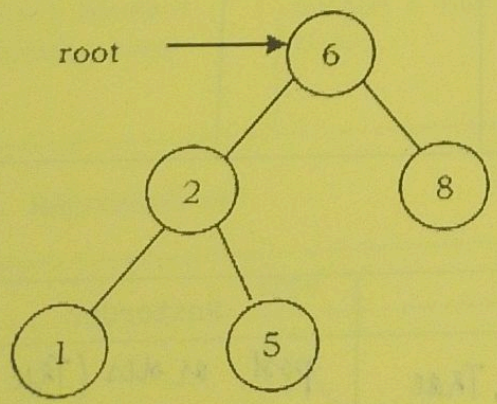
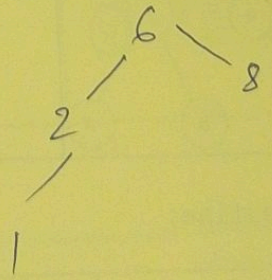
findMin (Binary Search Tree)	findMax (Binary Search Tree)

findmin (Node * Tree)
{
if (Tree != NULL)
inorder (Tree → left);
printf ("%d", Tree → data);
}
	findmax (Node* Tree)
	{
	if (Tree != NULL){
	inorder (Tree → right);
	printf ("%d", Tree → data);
	}}

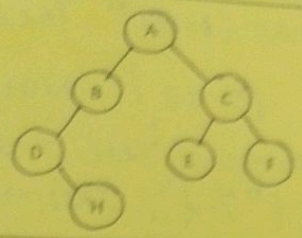
XI. Redraw me:

Binary Search Tree	insert(5)
	

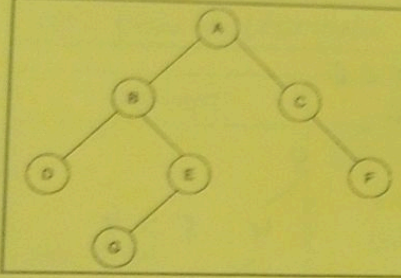
XII. Redraw me:

Binary Search Tree	delete(5)
	

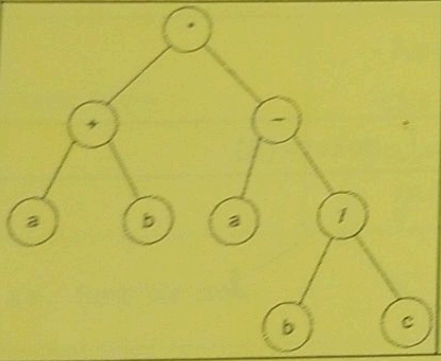
VI. Represent me:

Tree	Linked (SINGLY)
	A → B → D → H → C → E → F → NULL

VII. Represent me:

Tree	Inorder	Preorder	Postorder
	Inorder sequence D → B → G → E → A → C → F	A → B → D → G → E → C → F	D → G → E → B → F → C → A

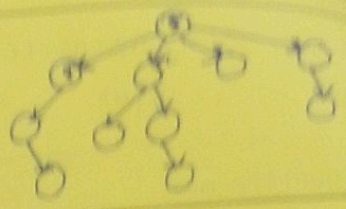
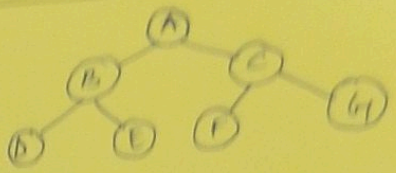
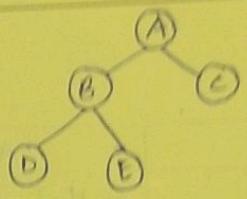
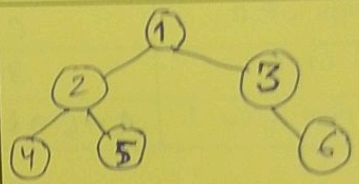
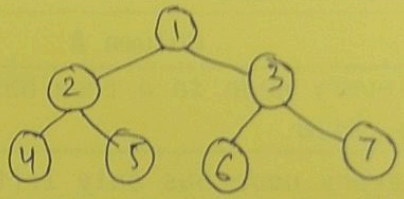
VIII. Represent me:

Tree	Inorder	Preorder	Postorder
	a + b * a - (b/c)	* + ab - a/bc	a * b + abc / -

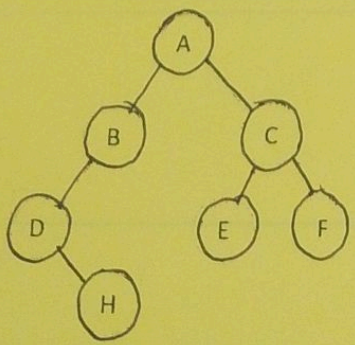
IX. Fill my routine:

Inorder	Preorder	Postorder
void inorder (Node * Tree) { if (Tree != NULL) { inorder (Tree → left); printf ("%d", Tree → data); inorder (Tree → right); } }	printf ("%d", Tree → data) preorder (Tree → left); preorder (Tree → right);	postorder (Tree → left); postorder (Tree → right); printf ("%d", Tree → data);

IV. Draw me:

Concept	Description	Example
General tree	General tree has any number of children.	
Binary tree	A Binary tree has not more than two children.	
Strict binary tree	A binary tree is called strict binary tree if each node has exactly two children or no children.	
Complete binary tree	A complete binary tree of height h has between 2^h and $2^{h+1} - 1$ nodes. In the bottom level the elements should be filled from left to right.	
Full binary tree (or) Perfect binary tree	A full binary tree of height h has $2^{h+1} - 1$ nodes.	

V. Represent me:

Tree	Linear
	A B D H C E F Preorder H D B A E C F Inorder H D B E F C A Post order

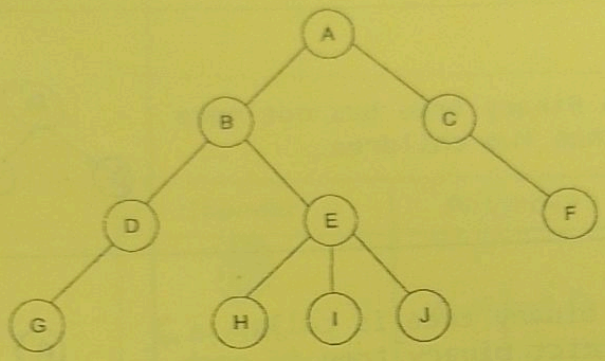
21CSC201J - DATA STRUCTURES AND ALGORITHMS

Unit-IV-Assignment

Reg. No. : RA2311026010654 Name : SURABHI KRISHNA

Year : II^ND Branch: CSE AI-ML Section: AA-2

I. Identify me:



Root	A	Parent of D	B
Leaf Nodes	G, H, I, J, F	Depth of J	3
Siblings of B	C	Height of B	2
Degree of E	3	Depth of tree	3
Path from A to J	A → C → E → J	Height of tree	3

II. Match me:

Column A	Column B
If every node in a tree has only one child.	Left skew tree
If every node has only left child.	Right skew tree
If every node has only right child.	Skew tree

III. Declare me:

Binary Tree Node
struct Node { int data; struct Node * left; struct Node * right; };