Q1-MCQ(35)

1.In a linked list implementation of a stack, what does the top pointer store?

2. In a queue implemented using a linked list, after deleting the only element in the queue, what happens to front and rear?

3. What happens when a new element is added to a full circular queue?

4. What will be the output of the following circular queue code?

5. In a deque, which of the following is true?

6. What will be the value of front and rear after inserting two elements into an initially empty queue implemented using an array?

7. Which of the following rotations is used in an AVL tree when a node is inserted into the left subtree of the right child?

8. A tree is said to be a/an ____ if all levels are filled except possibly the last level.

9. What will be the output of the post-order traversal after inserting 40, 20, 60, 10 into a BST?

10. What is the balance factor of the root node after inserting 30, 20, 10 in an empty AVL tree?

11. Consider the array representation of a binary tree: A[0]=10, A[1]=20, A[2]=30, A[3]=40, A[4]=50. What is the left child of A[1]?

12. Based on the tree above, which of the following statements accurately describes the degree, height, and depth of node (0070)?

13. If a stack is implemented using an array of size 5, which of the following statements is correct regarding the properties of the stack?

14. According to the following Binary Search Tree. What is the sequence of elements displayed after performing an inorder traversal of the given tree?

15. What is the correct statement regarding the deletion of the node with the value 30 from the given tree?

16. What is the correct statement regarding the insertion of a new node with the value 15 into the last tree?

17. What is the primary principle of stack data structure?

18. Which of the following is a disadvantage of using a doubly linked list compared to a singly linked list?

19. Which of the following correctly describes the imbalance in the given tree?

20. What will be the result of pushing three elements into an empty stack and then popping two of them?

21. What will happen if you attempt to pop an element from an empty stack?

22. ____ is a random-access data structure.

23. In a binary search tree, the left child of a node contains a value __ than the parent node, and the right child contains a value __ than the parent node.

24. What procedure is executed to re-establish balance in an AVL tree after an insertion process?

25. What is the allowed range of the balance factor for any node in an AVL tree?

True / False Questions:

26. Deletion in BST is always simpler than insertion.

27. The depth of a tree is the length of the shortest path from root to any leaf.

28. In a stack implemented with linked list, pop() removes the last node of the list.

29. Doubly linked lists support both forward and backward traversal.

30. Deleting a node in a singly linked list usually requires updating one pointer (the previous node's next).

31. In a queue represented as an array, the queue is considered empty if both rear and front are -1.

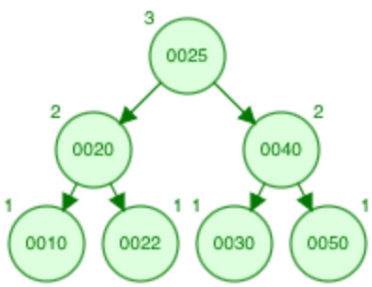
32. In a queue represented as an array, the queue contains only one element if the rear and front have different values.

33. In a queue represented as a linked list, when adding a new node (except the first element), both front and rear pointers must be updated.

34. In a doubly linked list, when adding a new node to the beginning of the list, the previous pointer of the new node is set to NULL.

35. The primary advantage of a queue over other data structures is easy to navigate and search.

Q.2
Construct an AVL tree containing the following elements:
40, 20, 10, 25, 30, 22, 50



Q.3
Write a function in Dobly Linked List Called **add()** to add a node After specified number of nodes

```
void linklist:: d_addafter (int loc, int num)
{
      dnode *q;
      q = p;
      // skip to desired portion
      for (int i = 1; i <= loc; i++)
      {
            q = q -> next;
      // if end of linked list is encountered
            if (q == NULL)
            {
            cout << "There are less than" <<
loc << "elements" << endl;
            return;
            }
}
      // insert new node
      q = q -> prev;
      dnode *temp = new dnode;
      temp -> data = num;
      temp -> prev = q;
      temp -> next = q -> next;
      temp -> next -> prev = temp;
      q -> next = temp;
}
```

Q.4
Write a function in Queue implemented as linked list called **extract()** to remove an element from Queue

```
int queue:: delq()
{
      if (front == NULL)
      {
      cout << "Queue is empty" << endl;
      return NULL;
      }
      node *temp;
      int item;
      item = front -> data;
      temp = front;
      front = front -> link;
      delete temp;
      return item;
}
```