Q1) Data Structures in C are used to store data in an organized and efficient manner.

Q2) Applications: array, queue, linked lists, stack, graph, tree, hash map and heap

Q3) Operations like insertion and deletion in arrays consume a lot of time.

The performance of these operations in Linked lists is fast.

Arrays are of fixed size. In contrast, Linked lists are dynamic and flexible and can expand and contract its size.

Elements are stored consecutively in arrays whereas it is stored randomly in Linked lists.

Q4) struct node

```
{
        int data;
        struct node *next;
};
```

Q5) Doubly linked lists are more efficient access to elements than Singly linked lists.

Q6) A stack is a linear data structure in which elements can be inserted and deleted only from one side of the list, called the top. A stack follows the LIFO Principle. Stack can contain elements of different data type.

An array is a collection of items stored at contiguous memory locations. Array contains elements of same data type.

Q7) 2 queues. one is used for storing data... another is used for priorities.

Q8) PreOrder, InOrder and PostOrder traversals

Q9) Binary Search Tree is a type of binary tree which keeps the keys in a sorted order for fast lookup. This makes searching process easy.

Q10) Graphs are used in Computer Science, Google maps, face book, World Wide Web, and operating system.

Q11) Yes, Binary search is possible on the linked list if the list is ordered and you know the count of elements in list.

Q12) In computer science, a memory leak is a type of resource leak that occurs when a computer program incorrectly manages memory allocations in a way that memory which is no longer needed

is not released. A memory leak may also happen when an object is stored in memory but cannot be accessed by the running code.

Q13) I will find the InOrder traversal of the binary tree and if the resultant is an ascending ordered array then we can say that the binary tree is a BST.

Q14) Stack. Because of its LIFO (Last In First Out) property it remembers its 'caller' so knows whom to return when the function has to return.

Recursion makes use of system stack for storing the return addresses of the function calls.

Q15) Expression Handling (INFIX, POSTFIX, PREFIX) Backtracking Procedure To check balanced parenthesis