Q1. What do you mean by a Data Structure?

Data Structure is a way of collecting and organising data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage. For example, we have some data which has, player's **name** "Virat" and **age** 26. Here "Virat" is of **String** data type and 26 is of **integer** data type.

2) What are some of the applications of DS?

Following are common DS applications:

1. Arrays : Implementation of other data structures, Execution of matrices and vectors, Dynamic memory allocation, Pointer container, Control tables.
2. Stack: Evaluation of expressions, Backtracking, Runtime memory management, Arrangement of books in a library.
3. Queue: Here, the data sent need not be received at the same rate at which it was sent. A certain system resource is to be shared between different processes.
4. Linked-List: Representation of sparse matrices, Non-contiguous data storage, Implementation of non-binary tree or other data structures, Dynamic memory management, Equalizing parenthesis, Symbol tables.
5. Set: Mapping of data, Common data storage.

Q3.What is advantage of linked list over arrays?

The size of the arrays is fixed: So we must know the upper limit on the number of elements in advance. Also, generally, the allocated memory is equal to the upper limit irrespective of the usage, and in practical uses, upper limit is rarely reached.

And inserting a new element in an array of elements is expensive, because room has to be created for the new elements and to create room existing elements have to shifted.

Q4.Write the syntax in C to create a node in the singly linked list.

```
public class SinglyLinkedList {

    //Represent a node of the singly linked list
    class Node{
```

```
        int data;
        Node next;

        public Node(int data) {
            this.data = data;
            this.next = null;
        }
    }

    //Represent the head and tail of the singly linked list
    public Node head = null;
    public Node tail = null;

    //addNode() will add a new node to the list
    public void addNode(int data) {
        //Create a new node
        Node newNode = new Node(data);

        //Checks if the list is empty
        if(head == null) {
            //If list is empty, both head and tail will point to new node
            head = newNode;
            tail = newNode;
        }
        else {
            //
newNode will be added after tail such that tail's next will point to newNo
de
            tail.next = newNode;
            //newNode will become new tail of the list
            tail = newNode;
        }
    }
```

```
    //display() will display all the nodes present in the list
    public void display() {
        //Node current will point to head
        Node current = head;

        if(head == null) {
            System.out.println("List is empty");
            return;
        }
        System.out.println("Nodes of singly linked list: ");
        while(current != null) {
            //Prints each node by incrementing pointer
            System.out.print(current.data + " ");
            current = current.next;
        }
        System.out.println();
    }

    public static void main(String[] args) {

        SinglyLinkedList sList = new SinglyLinkedList();

        //Add nodes to the list
        sList.addNode(1);
        sList.addNode(2);
        sList.addNode(3);
        sList.addNode(4);

        //Displays the nodes present in the list
        sList.display();
    }
}
```

Q5.What is the use of a doubly-linked list when compared to that of a singly linked list?

Doubly linked list allows element two way traversal. On other hand doubly linked list can be used to implement stacks as well as heaps and binary trees. Singly linked list is preferred when we need to save memory and searching is not required as pointer of single index is stored

Q6.What is the difference between an Array and Stack?

In an array, you have a list of elements and you can access any of them at any time. But in a stack, there's no random-access operation; there are only Push, Peek and Pop, all of which deal exclusively with the element on the top of the stack.
A stack is data-structure which has a last in first out policy.

Q7.What are the minimum number of Queues needed to implement the priority queue?

2 queues. one is used for storing data... another is used for priorities. Priority queues r applied using 2-D array where it has two rows one for element and second for priority ,so minimum numbers of queues are needed to implement are two.

Q8.What are the different types of traversal techniques in a tree?

The four as inorder, postorder, preorder and level order.

Q9.Why it is said that searching a node in a binary search tree is efficient than that of a simple binary tree?

It supports three main operations: searching of elements, insertion of elements, and deletion of elements. Binary Search Tree allows for fast retrieval of elements stored in the tree as each node key is thoroughly compared with the root node, which discards half of the tree.

Q10) What are the applications of Graph DS?

A graph is a non-linear data structure, which consists of vertices(or nodes) connected by edges(or arcs) where edges may be directed or undirected.
In Computer science graphs are used to represent the flow of computation.

Q11.Can we apply Binary search algorithm to a sorted Linked list?

Yes, Binary search is possible on the linked list if the list is ordered and you know the count of elements in list. But While sorting the list, you can access a single element at a time through a pointer to that node i.e. either a previous node or next node.

Q12) When can you tell that a Memory Leak will occur?

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)How will you check if a given Binary Tree is a Binary Search Tree or not?

There are two ways to check a given binary tree is a binary search tree.
If in-order traversal of the binary tree is sorted then is a bst.
Or we can travel each node in tree and check if left child is smaller than the parent and right child is larger than the parent.

Q14) Which data structure is ideal to perform recursion operation and why?

Stack has the LIFO (Last In First Out) property; it remembers it's ‘caller’. Therefore, it knows to whom it should return when the function has to return. On the other hand, recursion makes use of the system stack for storing the return addresses of the function calls.

Every recursive function has its equivalent iterative (non-recursive) function. Even when such equivalent iterative procedures are written explicit, stack is to be used.

Q15) What are some of the most important applications of a Stack?

Some important applications linked list are:
Expression Handling, Backtracking Procedure, return during function call.

Q16) Convert the below given expression to its equivalent Prefix And Postfix notations.

*+AB-CD (Infix : (A+B) * (C-D)) => AB+CD-* (Infix : (A+B * (C-D))