What is a Stack?

Stack is an ordered list in which, insertion and deletion can be performed only at one end that is called the top. It is a recursive data structure having pointer to its top element. The stack is sometimes called as Last-In-First-Out (LIFO) list i.e. the element which is inserted first in the stack will be deleted last from the stack.

List the area of applications where stack data structure can be used?

- Expression evaluation
- Backtracking
- Memory Management
- Function calling and return

What are the operations that can be performed on a stack?

- Push Operations
- Pop Operations
- Peek Operations

Write the stack overflow condition.

Overflow occurs when **top = Maxsize -1**

What is the difference between PUSH and POP?

PUSH and POP operations specify how data is stored and retrieved in a stack.

PUSH: PUSH specifies that data is being "inserted" into the stack.

POP: POP specifies data retrieval. It means that data is being deleted from the stack.

Write the steps involved in the insertion and deletion of an element in the stack.

Push:

- Increment the variable top so that it can refer to the next memory allocation
- Copy the item to the at the array index value equal to the top
- Repeat step 1 and 2 until stack overflows

Pop:

- Store the topmost element into the an another variable
- Decrement the value of the top
- Return the topmost element

What is a postfix expression?

An expression in which operators follow the operands is known as postfix expression. The main benefit of this form is that there is no need to group sub-expressions in parentheses or to consider operator precedence.

The expression "a + b" will be represented as "ab+" in postfix notation.

Write the postfix form of the expression: (A + B) * (C - D)

AB+CD-*

Which notations are used in Evaluation of Arithmetic Expressions using prefix and postfix forms?

Polish and Reverse Polish notations.

What is an array?

Arrays are defined as the collection of similar types of data items stored at contiguous memory locations. It is the simplest data structure in which each data element can be randomly accessed by using its index number.

How to reference all the elements in a one-dimension array?

It can be done by using an indexed loop such that the counter runs from 0 to the array size minus one. In this manner, you can reference all the elements in sequence by using the loop counter as the array subscript.

What is a multidimensional array?

The multidimensional array can be defined as the array of arrays in which, the data is stored in tabular form consists of rows and columns. 2D arrays are created to implement a relational database lookalike data structure. It provides ease of holding the bulk of data at once which can be passed to any number of functions wherever required.

How are the elements of a 2D array are stored in the memory?

There are two techniques by using which, the elements of a 2D array can be stored in the memory.

- **Row-Major Order:** In row-major ordering, all the rows of the 2D array are stored into the memory contiguously. First, the 1st row of the array is stored into the memory completely, then the 2nd row of the array is stored into the memory completely and so on till the last row.
- **Column-Major Order:** In column-major ordering, all the columns of the 2D array are stored into the memory contiguously. first, the 1st column of the array is stored into the memory completely, then the 2nd row of the array is stored into the memory completely and so on till the last column of the array.

Calculate the address of a random element present in a 2D array, given base address as BA.

Row-Major Order: If array is declared as a[m][n] where m is the number of rows while n is the number of columns, then address of an element a[i][j] of the array stored in row major order is calculated as,

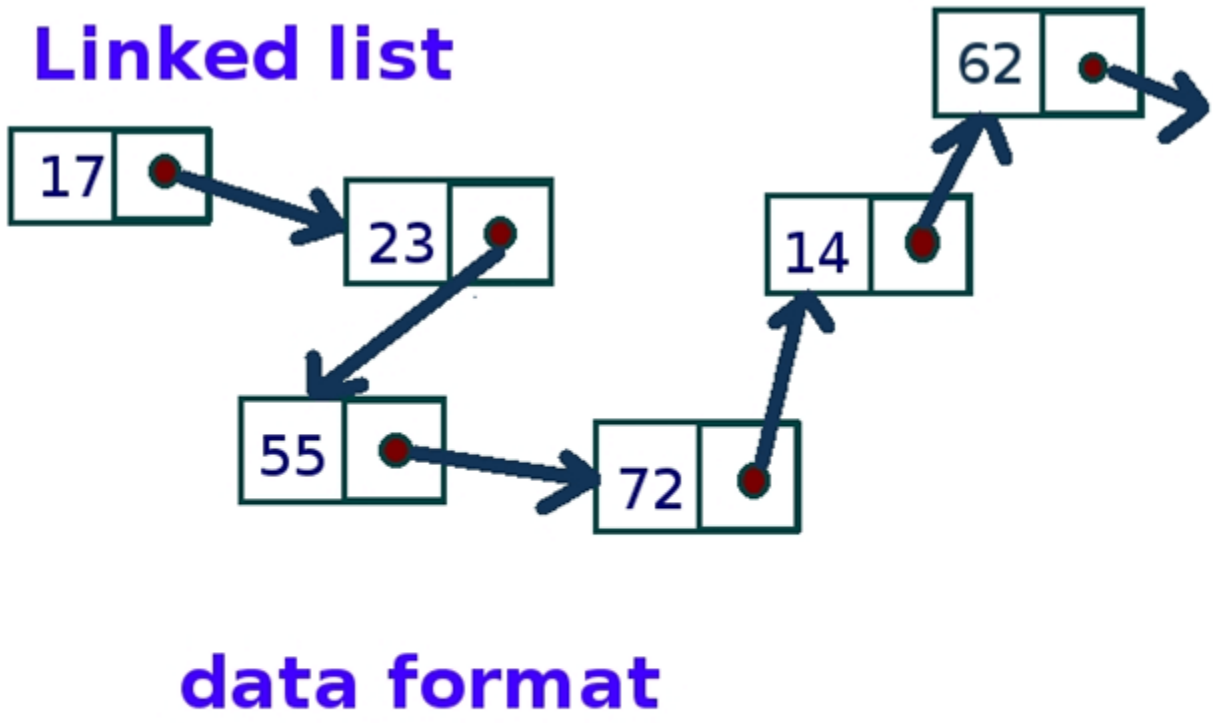
Address(a[i][j]) = B. A. + (i * n + j) * size

Column-Major Order: If array is declared as a[m][n] where m is the number of rows while n is the number of columns, then address of an element a[i][j] of the array stored in column major order is calculated as

Address(a[i][j]) = ((j*m)+i)*Size + BA.

Define Linked List Data structure.

Linked List is the collection of randomly stored data objects called nodes. In Linked List, each node is linked to its adjacent node through a pointer. A node contains two fields, i.e. Data Field and Link Field.



Are linked lists considered linear or non-linear data structures?

A linked list is considered both linear and non-linear data structure depending upon the situation.

- On the basis of data storage, it is considered as a non-linear data structure.

- On the basis of the access strategy, it is considered as a linear data-structure.

What are the advantages of Linked List over an array?

- The size of a linked list can be incremented at runtime which is impossible in the case of the array.
- The List is not required to be contiguously present in the main memory, if the contiguous space is not available, the nodes can be stored anywhere in the memory connected through the links.
- The List is dynamically stored in the main memory and grows as per the program demand while the array is statically stored in the main memory, size of which must be declared at compile time.
- The number of elements in the linked list are limited to the available memory space while the number of elements in the array is limited to the size of an array.

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

```
1. struct node
2. {
3.     int data;
4.     struct node *next;
5. };
6. struct node *head, *ptr;
7. ptr = (struct node *)malloc(sizeof(struct node));
```

If you are using C language to implement the heterogeneous linked list, what pointer type should be used?

The heterogeneous linked list contains different data types, so it is not possible to use ordinary pointers for this. For this purpose, you have to use a generic pointer type like void pointer because the void pointer is capable of storing a pointer to any type.

What is doubly linked list?

The doubly linked list is a complex type of linked list in which a node contains a pointer to the previous as well as the next node in the sequence. In a doubly linked list, a node consists of three parts:

- node data
- pointer to the next node in sequence (next pointer)
- pointer to the previous node (previous pointer).

Write the C program to insert a node in circular singly list at the beginning.

```
#include<stdio.h>
#include<stdlib.h>
void beg_insert(int);
struct node
{
    int data;
    struct node *next;
};
struct node *head;
void main ()
{
    int choice,item;
    do
    {
        printf("\nEnter the item which you want to insert?\n");
        scanf("%d",&item);
        beg_insert(item);
        printf("\nPress 0 to insert more ?\n");
        scanf("%d",&choice);
    }while(choice == 0);
}
void beg_insert(int item)
{

    struct node *ptr = (struct node *)malloc(sizeof(struct node));
    struct node *temp;
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
```

```
31.    else
32.    {
33.       ptr -> data = item;
34.       if(head == NULL)
35.       {
36.          head = ptr;
37.          ptr -> next = head;
38.       }
39.       else
40.       {
41.          temp = head;
42.          while(temp->next != head)
43.             temp = temp->next;
44.          ptr->next = head;
45.          temp -> next = ptr;
46.          head = ptr;
47.       }
48.    printf("\nNode Inserted\n");
49.    }
50.
51. }
52.
```

Define the queue data structure.

A queue can be defined as an ordered list which enables insert operations to be performed at one end called REAR and delete operations to be performed at another end called FRONT.

List some applications of queue data structure.

The Applications of the queue is given as follows:

- Queues are widely used as waiting lists for a single shared resource like a printer, disk, CPU.
- Queues are used in the asynchronous transfer of data (where data is not being transferred at the same rate between two processes) for eg. pipes, file IO, sockets.
- Queues are used as buffers in most of the applications like MP3 media player, CD player, etc.

- Queues are used to maintain the playlist in media players to add and remove the songs from the play-list.
- Queues are used in operating systems for handling interrupts.

What are the drawbacks of array implementation of Queue?

- **Memory Wastage:** The space of the array, which is used to store queue elements, can never be reused to store the elements of that queue because the elements can only be inserted at front end and the value of front might be so high so that, all the space before that, can never be filled.
- **Array Size:** There might be situations in which, we may need to extend the queue to insert more elements if we use an array to implement queue, It will almost be impossible to extend the array size, therefore deciding the correct array size is always a problem in array implementation of queue.

What are the scenarios in which an element can be inserted into the circular queue?

- If (rear + 1)%maxsize = front, the queue is full. In that case, overflow occurs and therefore, insertion can not be performed in the queue.
- If rear != max - 1, the rear will be incremented to the mod(maxsize) and the new value will be inserted at the rear end of the queue.
- If front != 0 and rear = max - 1, it means that queue is not full therefore, set the value of rear to 0 and insert the new element there.

What is a dequeue?

Dequeue (also known as double-ended queue) can be defined as an ordered set of elements in which the insertion and deletion can be performed at both the ends, i.e. front and rear.

What is the minimum number of queues that can be used to implement a priority queue?

Two queues are needed. One queue is used to store the data elements, and another is used for storing priorities.

Define the tree data structure.

The Tree is a recursive data structure containing the set of one or more data nodes where one node is designated as the root of the tree while the remaining nodes are called as the children of the root. The nodes other than the root node are partitioned into the nonempty sets where each one of them is to be called sub-tree.

List the types of tree.

There are six types of tree given as follows.

- General Tree
- Forests
- Binary Tree
- Binary Search Tree
- Expression Tree
- Tournament Tree

What are Binary trees?

A binary Tree is a special type of generic tree in which, each node can have at most two children. Binary tree is generally partitioned into three disjoint subsets, i.e. the root of the node, left sub-tree and Right binary sub-tree.

Write the C code to perform in-order traversal on a binary tree.

```
1.  void in-order(struct treenode *tree)
2.    {
3.       if(tree != NULL)
4.       {
5.          in-order(tree→ left);
6.          printf("%d",tree→ root);
7.          in-order(tree→ right);
8.       }
9.    }
```

What is the maximum number of nodes in a binary tree of height k?

2^{k+1}-1 where k >= 1

Which data structure suits the most in the tree construction?

Queue data structure

Which data structure suits the most in the tree construction?

Queue data structure

Write the recursive C function to count the number of nodes present in a binary tree.

```
1.  int count (struct node* t)
2.  {
3.     if(t)
4.     {
5.        int l, r;
6.        l = count(t->left);
7.        r=count(t->right);
8.        return (1+l+r);
9.     }
10.    else
11.    {
12.       return 0;
13.    }
14. }
```

Write a recursive C function to calculate the height of a binary tree.

```
1.  int countHeight(struct node* t)
2.  {
3.     int l,r;
```

```
4.      if(!t)
5.          return 0;
6.      if((!(t->left)) && (!(t->right)))
7.          return 0;
8.      l=countHeight(t->left);
9.      r=countHeight(t->right);
10.     return (1+((l>r)?l:r));
11. }
```

How can AVL Tree be useful in all the operations as compared to Binary search tree?

AVL tree controls the height of the binary search tree by not letting it be skewed. The time taken for all operations in a binary search tree of height h is O(h). However, it can be extended to O(n) if the BST becomes skewed (i.e. worst case). By limiting this height to log n, AVL tree imposes an upper bound on each operation to be O(log n) where n is the number of nodes.

State the properties of B Tree.

A B tree of order m contains all the properties of an M way tree. In addition, it contains the following properties.

- Every node in a B-Tree contains at most m children.
- Every node in a B-Tree except the root node and the leaf node contain at least m/2 children.
- The root nodes must have at least 2 nodes.
- All leaf nodes must be at the same level.

What are the differences between B tree and B+ tree?

SN	B Tree	B+ Tree
1	Search keys cannot repeatedly be stored.	Redundant search keys can be pre
2	Data can be stored in leaf nodes as well as internal nodes	Data can only be stored on the lea

3	Searching for some data is a slower process since data can be found on internal nodes as well as on the leaf nodes.	Searching is comparatively faster be found on the leaf nodes.
4	Deletion of internal nodes is so complicated and time-consuming.	Deletion will never be a complexe element will always be deleted fro
5	Leaf nodes cannot be linked together.	Leaf nodes are linked together to operations more efficient.

List some applications of Tree-data structure?

Applications of Tree- data structure:

- The manipulation of Arithmetic expression,
- Symbol Table construction,
- Syntax analysis
- Hierarchal data model

Define the graph data structure?

A graph G can be defined as an ordered set G(V, E) where V(G) represents the set of vertices and E(G) represents the set of edges which are used to connect these vertices. A graph can be seen as a cyclic tree, where the vertices (Nodes) maintain any complex relationship among them instead of having parent-child relations.

Differentiate among cycle, path, and circuit?

- **Path:** A Path is the sequence of adjacent vertices connected by the edges with no restrictions.
- **Cycle:** A Cycle can be defined as the closed path where the initial vertex is identical to the end vertex. Any vertex in the path can not be visited twice
- **Circuit:** A Circuit can be defined as the closed path where the intial vertex is identical to the end vertex. Any vertex may be repeated.

Mention the data structures which are used in graph implementation.

For the graph implementation, following data structures are used.

- In sequential representation, Adjacency matrix is used.
- In Linked representation, Adjacency list is used.

Which data structures are used in BFS and DFS algorithm?

- In BFS algorithm, Queue data structure is used.
- In DFS algorithm, Stack data structure is used.

What are the applications of Graph data structure?

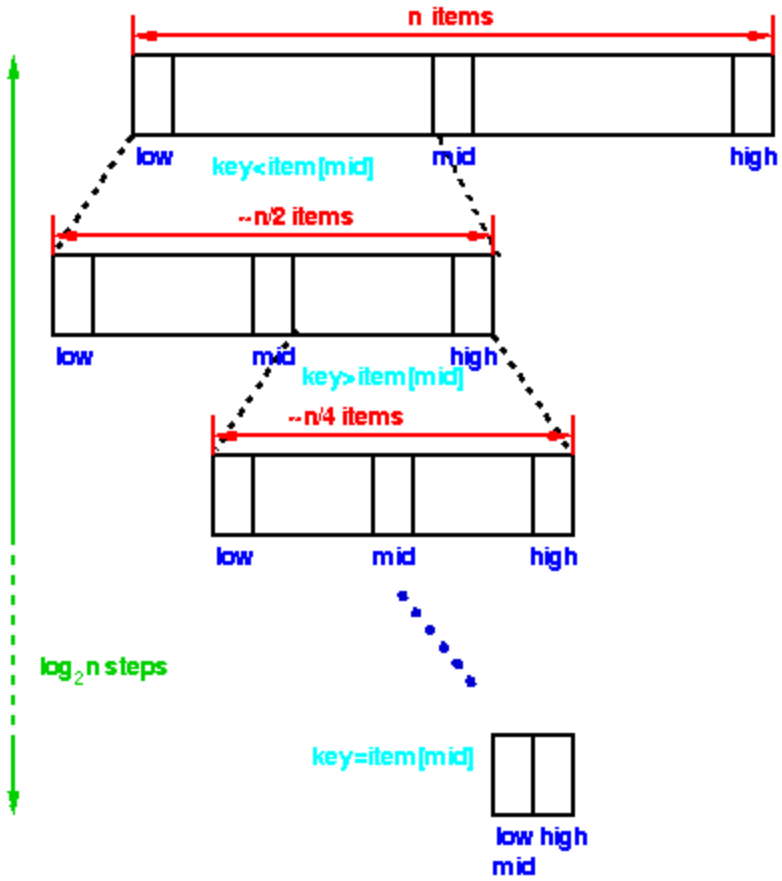
The graph has the following applications:

- Graphs are used in circuit networks where points of connection are drawn as vertices and component wires become the edges of the graph.
- Graphs are used in transport networks where stations are drawn as vertices and routes become the edges of the graph.
- Graphs are used in maps that draw cities/states/regions as vertices and adjacency relations as edges.
- Graphs are used in program flow analysis where procedures or modules are treated as vertices and calls to these procedures are drawn as edges of the graph.

In what scenario, Binary Search can be used?

Binary Search algorithm is used to search an already sorted list. The algorithm follows divide and conqer approach

Example:


n items
low
key<item[mid]
mid
high
~n/2 items
low
mid
high
key>item[mid]
~n/4 items
low
mid
high
log2n steps
key=item[mid]
low high
mid