

Short answer questions on Stacks & Queues

1. Define Stack.

A stack is an ordered list in which all insertions and deletions are made at one end, called the top. It is an abstract data type and based on the principle of LIFO (Last In First Out).

2. What are the operations of the stack?

- a. Push(Item) – pushes an item on the top of the stack
- c. Pop(Item) – removes the top most element from the stack
- d. Top(Stack) – returns the first element from the stack
- e. IsFull(Stack) – returns true if the stack is full
- f. IsEmpty(Stack) – returns true if the stack is empty

3. How do you test for an empty stack?

To check if the stack is empty,

- when stack is implemented by using array, we need to check whether $\text{top} == -1$.
- when stack is implemented by using a linked list, we need to check whether $\text{top} == \text{NULL}$.

4. How do you test for stack is full/ stack overflow condition?

We need to check if the stack is full only when it is implemented by using array, we need to check whether $\text{top} == \text{MAXSIZE}-1$.

For example, if the max size of stack is 10, then we need to check if $\text{top} == 9$.

5. What are the postfix and prefix forms of the expression $A+B*(C-D)/(P-R)$?

- Postfix form: $\text{ABCD-}*PR-/+$
- Prefix form: $+A/*B-CD-PR$

6. Evaluate the postfix expression $\text{ABC}*+D-$ when $A=2, B=3, C=4$ and $D=3$.

The result is 11. $(2+(3*4)-3)$.

7. What are the applications of stack?

The following are the applications of stacks

- Evaluating arithmetic expressions
- Balancing the parenthesis
- Towers of Hanoi
- Function calls
- Depth First Search traversal on a graph

8. Stack overflow condition is not checked in linked stack: give reason.

Stack overflow condition arises whenever memory required to store stack elements is allocated statically (using an array) and the elements placed in the stack are of maximum capacity of the stack. **and we want to push an element into the stack.** This will occur because we have to define the capacity of the array. Linked stacks are dynamic data structures, whose memory is allocated dynamically, that can be stored

anywhere, non-contiguously in the memory. We need not define capacity of linked stack. It can grow as long as there is a vacant slot in memory. Hence, stack overflow condition doesn't arise for linked stacks.

9. Distinguish between stack and queue.

Stack	Queue
Based on Last in First Out (LIFO) principle	Based on First in First Out (FIFO) principle
Insertion and deletion are made at one end.	Insertion is made at one end and deletion at other end.

10. What are the methods to implement stack in C?

The methods to implement stacks are:

- Array based
- Linked list based

11. Define queue.

It is a linear data structure that maintains a list of elements such that insertion happens at rear end and deletion happens at front end. Queue is based on FIFO (First In First Out) principle

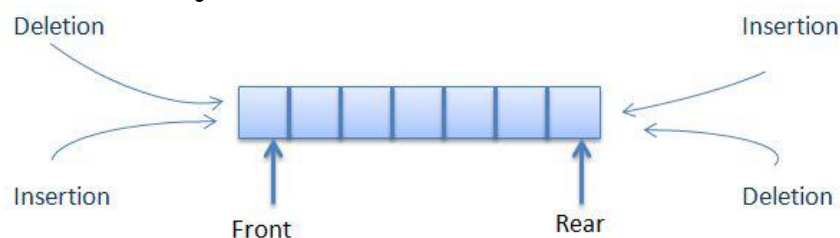
12. List the types of queues.

The following are the types of queues:

- Double ended queue
- Circular queue
- Priority queue

13. Define double ended queue

It is a special type of queue that allows insertion and deletion of elements at both Ends. It is also termed as DEQUE.



14. What are the methods to represent/implement queue in C?

The methods to implement queues are:

- Array based
- Linked list based

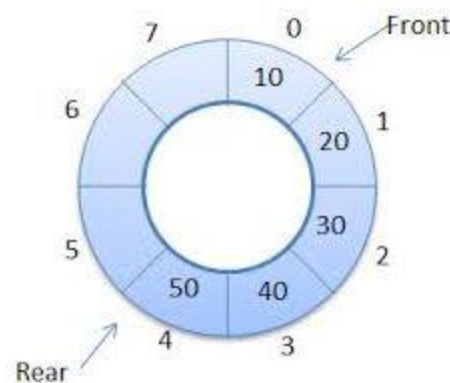
15. List the applications of queue.

1. Serving requests on a single shared resource, like a printer, CPU task scheduling etc.

2. In real life scenario, Call Center phone systems uses Queues to hold people calling them in an order, until a service representative is free.
3. Handling of interrupts in real-time systems. The interrupts are handled in the same order as they arrive i.e First come first served.
4. Maintaining the playlist in media players.

16. Define circular queue

A Circular queue is a queue whose start and end locations are logically connected with each other. That means the start location comes after the end location.



17. How do you test for an empty queue?

- For a queue implemented using array – check if front and rear are equal to initial value or front points to next of rear ($\text{front} == -1 \ \&\& \ \text{rear} == -1 \ || \ \text{front} > \text{rear}$)
- For a queue implemented using linked list – check if front and rear are equal to NULL. ($\text{front} == \text{NULL} \ \&\& \ \text{rear} == \text{NULL}$)

18. How do you test for queue is full?

- For linear queue implemented using array – ($\text{REAR} == \text{MAXSIZE}-1$)
- For Circular queue implemented using array - ($((\text{REAR}+1) \% \text{MAXSIZE} == \text{FRONT})$)

19. Define Priority queue.

A priority queue is an abstract data type that behaves similarly to the normal queue except that each element has some priority, i.e., the element with the highest priority would come first in a priority queue. The priority of the elements in a priority queue will determine the order in which elements are removed from the priority queue.

20. Write all operations performed on a Queue.

- Enqueue: Add an element to the end of the queue.
- Dequeue: Remove an element from the front of the queue.
- IsEmpty: Check if the queue is empty.
- IsFull: Check if the queue is full.
- Peek: Get the value of the front of the queue without removing it.

21. Compare Array based queue with linked queue.

An **array based Queue** is a queue of fixed size implemented using array. Hence there is a restriction on maximum number of elements that can be placed in the queue. Due to this, an array based queue requires to check for queue overflow condition. While a dynamic queue has no size restriction. It can grow dynamically.

22. List out the applications of heap tree/ priority queue.

- Heap Implemented priority queues are used in Graph algorithms like Prim's Algorithm and Dijkstra's algorithm.
- *Order statistics*: The Heap data structure can be used to efficiently find the kth smallest (or largest) element in an array.

23. The postfix equivalent of the prefix * + a b - c d is
ab+cd-*