1. What is a queue data structure?
   1. **A linear data structure**
   2. A non-linear data structure
   3. A hierarchical data structure
   4. None of the above
2. In a queue, which element is removed first?
   1. **Front**
   2. Rear
   3. Middle
   4. Random
3. Which operation adds an element to the queue?
   1. **Enqueue**
   2. Dequeue
   3. Peek
   4. Remove
4. Which operation removes an element from the queue?
   1. Enqueue
   2. **Dequeue**
   3. Peek
   4. Remove
5. Which of the following is not a characteristic of a queue?
   1. First-In-First-Out (FIFO) order
   2. **Last-In-First-Out (LIFO) order**
   3. Insertion at the rear
   4. Deletion at the front
6. Which data structure is typically used to implement a queue?
   1. Array
   2. **Linked list**
   3. Stack
   4. Tree
7. What is the time complexity to enqueue an element in a queue?
   1. **O(1)**
   2. O(n)
   3. O(log n)
   4. O(n^2)
8. What is the time complexity to dequeue an element from a queue?
   1. **O(1)**
   2. O(n)
   3. O(log n)
   4. O(n^2)
9. Which operation is used to retrieve the element at the front of the queue without removing it?
   1. Enqueue
   2. Dequeue
   3. **Peek**
   4. Remove
10. Which statement is true about an empty queue?
    1. **Front and rear are both NULL**
    2. Front and rear point to the same element
    3. Front points to the last element, and rear is NULL
    4. Front is NULL, and rear points to the last element
11. How can you check if a queue is empty?
    1. Check if front is NULL
    2. Check if rear is NULL
    3. **Check if both front and rear are NULL**
    4. Check if both front and rear are not NULL
12. How can you check if a queue is full?
    1. Check if front is NULL
    2. Check if rear is NULL
    3. Check if both front and rear are NULL
    4. **Check if rear=size-1**
13. Which operation is used to count the number of elements in a queue?
    1. Count
    2. Size
    3. Length
    4. **None of the above**
14. Which of the following is not an advantage of using a queue data structure?
    1. **Efficient insertion and deletion at both ends**
    2. Easy implementation using linked list or array
    3. Provides a natural way to model real-life scenarios
    4. Follows the FIFO order, which is useful in scheduling and resource allocation
15. Which of the following is not a standard queue operation?
    1. **Merge**
    2. Enqueue
    3. Dequeue
    4. Peek
16. How many queues are needed to implement a stack. Consider the situation where no other data structure like arrays, linked list is available to you.
    1. 1
    2. **2**
    3. 3
    4. 4
17. Which of the following is true about a circular queue?
    1. It uses a doubly linked list for implementation
    2. Elements can only be inserted at the rear end
    3. **The front and rear pointers move in a circular manner**
    4. Elements can be inserted and removed from both ends
18. Which operation is used to initialize an empty queue?
    1. init\_queue()
    2. create\_queue()
    3. new\_queue()
    4. **None of the above**
19. Which of the following data structures can be used to implement a priority queue?
    1. Array
    2. Linked list
    3. Binary heap
    4. **All of the above**
20. What is the time complexity to insert an element with a priority in a priority queue?
    1. O(1)
    2. O(n)
    3. **O(log n)**
    4. O(n^2)
21. What is the time complexity to remove the highest priority element from a priority queue?
    1. O(1)
    2. O(n)
    3. **O(log n)**
    4. O(n^2)
22. Which of the following is not a valid approach to implement a queue using an array?
    1. Fixed-size circular array
    2. Dynamic array with resizing
    3. Dynamic array with fixed-size limit
    4. **Linked list with nodes containing the array elements**
23. Which of the following is a disadvantage of using a linked list to implement a queue?
    1. Efficient random access to elements
    2. **Requires extra memory for pointers**
    3. Limited capacity based on the size of the array
    4. Limited flexibility for resizing
24. Which of the following is not a queue variant?
    1. Circular queue
    2. Priority queue
    3. Double-ended queue (Deque)
    4. **Stack queue**
25. In a circular queue, what happens when the rear pointer reaches the end of the array?
    1. Elements cannot be inserted anymore
    2. **The rear pointer wraps around to the beginning of the array**
    3. The rear pointer remains at the end, and elements are inserted at the beginning
    4. The program terminates with an error
26. Which operation is used to check the element at the front of a queue?
    1. rear()
    2. last()
    3. **peek()**
    4. top()
27. Which of the following is true about a double-ended queue (Deque)?
    1. Elements can only be inserted and removed from the front end
    2. Elements can only be inserted and removed from the rear end
    3. **Elements can be inserted and removed from both ends**
    4. Elements can be inserted and removed from any position
28. Which data structure is commonly used to implement a deque?
    1. Array
    2. **Linked list**
    3. Stack
    4. Queue
29. Which of the following is not a valid advantage of using a deque over a queue or stack?
    1. Efficient insertion and deletion at both ends
    2. Flexibility to insert and remove elements from any position
    3. **Ability to store elements of different data types**
    4. Provides support for double-ended operations
30. Guess the output:

void fun(Queue \*Q)

{

Stack S; // Say it creates an empty stack S

// Run while Q is not empty

while (!isEmpty(Q))

{

// deQueue an item from Q and push the dequeued item to S

push(&S, deQueue(Q));

}

// Run while Stack S is not empty

while (!isEmpty(&S))

{

// Pop an item from S and enqueue the popped item to Q

enQueue(Q, pop(&S));

}

}

What does the above function do in general?

a)Removes the last from Q

b)Keeps the Q same as it was before the call

c) Makes Q empty

**d) Reverse Queue**

1. Which one of the following is an application of Queue Data Structure?

a) When a resource is shared among multiple consumers.

b) When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes

c) Load Balancing

**d) All of the above**

1. Suppose a stack implementation supports an instruction REVERSE, which reverses the order of elements on the stack, in addition to the PUSH and POP instructions. Which one of the following statements is TRUE with respect to this modified stack?
2. A queue cannot be implemented using this stack.
3. A queue can be implemented where ENQUEUE takes a single instruction and DEQUEUE takes a sequence of two instructions.
4. **A queue can be implemented where ENQUEUE takes a sequence of three instructions and DEQUEUE takes a single instruction.**
5. A queue can be implemented where both ENQUEUE and DEQUEUE take a single instruction each.
6. A queue is implemented using an array such that ENQUEUE and DEQUEUE operations are performed efficiently. Which one of the following statements is CORRECT (n refers to the number of items in the queue)?
   1. **Both operations can be performed in O(1) time**
   2. At most one operation can be performed in O(1) time but the worst case time for the other operation will be Ω(n)
   3. The worst case time complexity for both operations will be Ω(n)
   4. Worst case time complexity for both operations will be Ω(log n)
7. Consider the following statements:
   1. First-in-first out types of computations are efficiently supported by STACKS.
   2. Implementing LISTS on linked lists is more efficient than implementing LISTS on an array for almost all the basic LIST operations.
   3. Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices.
   4. Last-in-first-out type of computations are efficiently supported by QUEUES.

Which of the following is correct?

* + 1. (ii) is true
    2. (i) and (ii) are true
    3. **(iii) is true**
    4. (ii) and (iv) are true