

## Stack and Queue

1. Stack follows the principle of **LIFO**.
2. The process of inserting an element into the stack is called as **Push**.
3. The process of deleting an element from the stack is called as **Pop**.
4. The process of accessing data stored in a serial access memory is similar to manipulating data on a **Stack**.
5. The data structure required to evaluate a postfix expression is **Stack**.
6. Expression in which Operator is written after Operand is called as **Postfix Expression**.
7. Expression in which Operator is written before the Operand is called as **Prefix Expression**.
8. Expression in which Operator is written between two Operand is called as **Infix Expression**.
9. What is the other name for a postfix expression? **Reverse polish Notation**.
10. Which of the following is an example for a postfix expression?  
**abc\*+de-+**.
11. The data structure required to check whether an expression contains balanced parenthesis is **Stack**.
12. Which data structure is needed to convert infix notation to postfix notation? **Stack**.
13. Which of the following statement(s) about stack data structure is/are NOT correct? **Stack is the FIFO data structure**.
14. In the stack, If user try to remove element from the empty stack then it called as **Underflow of Stack**.
15. User push 1 element in the stack having already five elements and having stack size as 5 then stack becomes **Overflow of Stack**.
16. Queue follows the principle of **FIFO**.
17. The process of inserting an element into the queue is called as **Enqueue**.
18. The process of deleting an element from the queue is called as **Dequeue**.

## Stack and Queue

19. Circular Queue is also known as **Ring Buffer**.
20. If the last position is full and the first position is empty then, an element can be inserted in the first position. This action is possible in a **Circular Queue**.
21. In a queue data structure all insertion and deletion are made respectively at: **rear and front**.
22. In a stack data structure all insertion and deletion are made respectively at: **top**.
23. Enqueue and Dequeue functions are found in **Queue**.
24. push and pop functions are found in **Stack**.
25. Which of the following data structure is an example of linear data structure? **Queue**.
26. Which of the following data structure is an example of non linear data structure? **Tree**.
27. Identify the data structure which allows deletion at both ends of the list but insertion at only one end. **Input restricted Queue**.
28. Identify the data structure which allows insertion at both ends of the list but deletion at only one end. **Output Restricted Queue**.
29. Identify the data structure which allows insertion and deletion are performed at both ends. **Dequeue**.
30. A circular queue is implemented using an array of size 10. The array index starts with 0, front is 6, and rear is 9. The insertion of next element takes place at the array index. **0**.
31. Evaluate Postfix expression from given infix expression.  $A + B * (C + D) / F + D * E$ . **ABCD+\*F/+DE\*+**.
32. The postfix expression corresponding to the infix expression  $a + b \times c - d \wedge e \wedge f$  is **abc × + def ^ ^ -**.
33. The five items: A,B,C,D and E are pushed in stack, one after the other starting from A. The stack is popped four items and each element is

## Stack and Queue

- inserted in a queue. Then two elements are deleted from the queue and pushed back on the stack. Now one item is popped from the stack. **D**.
34. Consider the following pseudocode that uses a stack declare a stack of characters. What is output for input "Infosys"? **sysofnI**.
35. The following postfix expression with single digit operands is evaluated using a stack:  $8\ 2\ 3\ ^\wedge\ /\ 2\ 3\ *\ +\ 5\ 1\ *\ -$ . Note that  $^\wedge$  is the exponentiation operator. The top two elements of the stack after the first  $*$  is evaluated are: **6, 1**.
36. The result evaluating the postfix expression  $10\ 5\ +\ 60\ 6\ /\ *\ 8\ -$  is **142**.
37. What is the result of the following operation: Top (Push (S[], 50))? **50**.
38. If the elements "A", "B", "C" and "D" are placed in a queue and are deleted one at a time, in what order will they be removed? **ABCD**.
39. **Double ended Queue** is the type of queue in which insertion and removal of elements can be performed from either from the front or rear.
40. The conditions used for checking the fullness of the circular queue are: **FRONT = 0 && REAR == MAXSIZE - 1**.
41. If the MAX\_SIZE is the size of the array used in the implementation of circular queue. How is rear manipulated while inserting an element in the queue? **rear=(rear+1)%MAX\_SIZE**.
42. A circular queue is implemented using an array of size 10. The array index starts with 0, front is 6, and rear is 9. The insertion of next element takes place at the array index. **0**.
43. If the MAX\_SIZE is the size of the array used in the implementation of circular queue, array index start with 0, front point to the first element in the queue, and rear point to the last element in the queue. Which of the following condition specify that circular queue is EMPTY? **Front=**  
**rear=-1**.

## Stack and Queue

44. In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into EMPTY queue? **Both front and rear pointer.**
45. An array of size MAX\_SIZE is used to implement a circular queue. Front, Rear, and count are tracked. Suppose front is 0 and rear is MAX\_SIZE - 1. How many elements are present in the queue? **MAX\_SIZE.**

### **Reference:**

1. <https://www.careerride.com/view/mcqs-on-queue-with-answers-19633.aspx>
2. <https://affairsccloud.com/data-structures-questions-set-10/>