# 📘 UNIT 2 – STACKS AND QUEUES

--------------------------------------------

(Questions 1–20)

Q1. Stack follows which principle?
a) FIFO
b) FILO
c) LIFO
d) Both b and c

Q2. The basic operations of a stack are _______.
a) insert(), delete()
b) push(), pop()
c) add(), remove()
d) enq(), deq()

Q3. The operation of inserting an element into a stack is known as _______.
a) append
b) pop
c) push
d) enqueue

Q4. Which data structure is used for recursive function calls?
a) Queue
b) Stack
c) Tree
d) Graph

Q5. Which of the following is an application of stack?
a) Function call management
b) CPU scheduling
c) Printer queue
d) Hashing

Q6. If a stack is implemented using an array, what happens when the stack overflows?
a) Memory leak
b) Runtime error
c) Program crash
d) Both b and c

Q7. Fill in the blank: The process of removing an element from stack is called _______.
a) push
b) pop
c) delete
d) remove

Q8. If the sequence of stack operations is:
PUSH(1), PUSH(2), POP(), PUSH(3), POP(), then the popped elements are _______.
a) 2, 3
b) 1, 3
c) 2, 1
d) 3, 2

Q9. In stack implementation using linked list, which pointer represents the top of the stack?
a) Head pointer
b) Tail pointer
c) Top pointer
d) Front pointer

Q10. The time complexity of push() and pop() operations in a stack is _______.
a) O(n)

b) O(log n)
c) O(1)
d) $O(n^2)$

Q11. A queue follows which principle?
a) LIFO
b) FIFO
c) FILO
d) None

Q12. In queue implementation, inserting an element is called _______.
a) enqueue
b) dequeue
c) push
d) pop

Q13. In queue implementation, removing an element is called _______.
a) enqueue
b) dequeue
c) pop
d) delete

Q14. What is the output of the following code?

```
queue<int> q;
q.push(10);
q.push(20);
q.pop();
cout << q.front();
```

a) 10
b) 20
c) 30
d) Error

Q15. Fill in the blank: In a circular queue, when rear = size - 1, the next value of rear is _______.
a) 0
b) 1
c) NULL
d) size + 1

Q16. Which of the following is not a type of queue?
a) Circular queue
b) Priority queue
c) Double-ended queue
d) Binary queue

Q17. The main disadvantage of using an array to implement a queue is _______.
a) Overflow
b) Wasted space
c) Complexity
d) Both a and b

Q18. A double-ended queue (Deque) allows insertion and deletion from _______.
a) Rear end only
b) Front end only
c) Both ends
d) Middle

Q19. Fill in the blank: In stack, if top == -1, it means the stack is _______.
a) Full
b) Empty
c) Overflow
d) Undefined

Q20. The data structure used to convert infix to postfix expression is _______.
a) Queue
b) Tree
c) Stack
d) Graph

--------------------------------------------

🔷 ANSWER KEY – UNIT 2 (Q1–Q20)

--------------------------------------------

1. d — Stack works on LIFO/FILO principle.
2. b — push() for insert, pop() for remove.
3. c — push adds element to top.
4. b — Recursion uses stack for storing function states.
5. a — Stack manages nested/recursive function calls.
6. d — Overflow may cause runtime error or crash.
7. b — pop removes element from top.
8. a — First POP removes 2, then POP removes 3.
9. c — Top pointer indicates current top node.
10. c — Stack operations are O(1).
11. b — Queue follows FIFO principle.
12. a — enqueue adds element to queue.
13. b — dequeue removes from queue front.
14. b — After pop(), front = 20.
15. a — Circular queue wraps rear to 0.
16. d — Binary queue doesn’t exist.
17. d — Arrays cause overflow and wasted space.
18. c — Deque = insertion/deletion at both ends.
19. b — top = -1 → empty stack.
20. c — Stack used in expression conversion (infix→postfix).

--------------------------------------------

📘 END OF SET 1 (UNIT 2: QUESTIONS 1–20)

--------------------------------------------

📘 UNIT 2 – STACKS AND QUEUES

--------------------------------------------

(Questions 21–40)

Q21. What is the maximum number of elements that can be stored in a circular queue of size n?
a) n
b) n - 1
c) n + 1
d) Unlimited

Q22. In stack implementation using array, overflow occurs when _______.
a) top == size
b) top == size - 1
c) top == 0
d) top == NULL

Q23. Fill in the blank: In queue implementation, underflow occurs when _______.
a) front == rear
b) front == -1
c) queue is empty
d) both b and c

Q24. Which of the following operations are performed in a circular queue?
a) enqueue() and dequeue()
b) push() and pop()
c) insert() and delete()
d) None of these

Q25. In postfix expression evaluation using stack, operands are _______ and operators are _______.
a) pushed, popped
b) popped, pushed
c) deleted, inserted
d) none

Q26. Which of the following represents the correct postfix form of the expression (A + B) * (C - D)?
a) A B + C D - *
b) A + B * C - D
c) A B + * C D -
d) A B C D + - *

Q27. What is the result of the postfix expression:
`5 6 2 + * 12 4 / -`
a) 37
b) 34
c) 32
d) 40

Q28. Which of the following is NOT an application of queue?
a) Job scheduling
b) Keyboard buffer
c) Recursion
d) Printer spooling

Q29. The time complexity of enqueue and dequeue operations in a circular queue is _______.
a) O(1)
b) O(n)
c) O(log n)
d) $O(n^2)$

Q30. What is the output of this code snippet?

```
stack<int> s;
s.push(10);
s.push(20);
cout << s.top();
```

a) 10
b) 20
c) 30
d) Garbage value

Q31. Fill in the blank: When using an array to implement a queue, the front and rear pointers must be updated _______.
a) Linearly
b) Recursively
c) Circularly
d) Randomly

Q32. Which of the following is not a valid operation on a queue?
a) Enqueue
b) Dequeue
c) Push
d) Display

Q33. The condition for an empty circular queue is _______.
a) front == rear
b) front == -1
c) rear == -1
d) rear == NULL

Q34. Which of the following operations is performed first in postfix expression evaluation?
a) Leftmost operator
b) Rightmost operand
c) Leftmost operand
d) First operator encountered

Q35. Which of the following data structures is used in the *Undo* operation of text editors?
a) Queue
b) Stack
c) Deque
d) Tree

Q36. Fill in the blank: The time complexity of converting infix to postfix using a stack is _______.
a) O(1)
b) O(n)
c) O(log n)
d) $O(n^2)$

Q37. What is the time complexity for evaluating a postfix expression of n tokens?
a) O(1)
b) O(n)
c) O(n log n)
d) $O(n^2)$

Q38. Which queue gives priority to elements with higher priority values?
a) Circular queue
b) Deque
c) Priority queue
d) Double queue

Q39. In a circular queue implemented using an array of size 5,
if front = 2 and rear = 4, after one dequeue and two enqueue operations,
what will be the new front and rear?
a) front = 3, rear = 1
b) front = 4, rear = 1
c) front = 2, rear = 3
d) front = 0, rear = 4

Q40. Fill in the blank: Stack overflow occurs when we try to _______.
a) pop from empty stack
b) push into full stack
c) push into empty stack
d) pop from non-empty stack

--------------------------------------------

🔷 ANSWER KEY – UNIT 2 (Q21–Q40)

--------------------------------------------

21. b — One space is reserved to differentiate full/empty → max = n - 1.
22. b — Overflow occurs when top reaches size - 1.
23. d — Queue underflow occurs when queue is empty or front = -1.
24. a — enqueue() adds, dequeue() removes in circular queue.
25. a — Operands pushed, operators popped for evaluation.
26. a — Infix (A + B)*(C - D) → Postfix A B + C D - *.
27. b — Expression evaluates to 34.
28. c — Recursion uses stack, not queue.
29. a — Both enqueue and dequeue take O(1).
30. b — Top element = 20.
31. c — In circular queue, pointers updated circularly.
32. c — Push is used in stack, not queue.

33. a — Empty circular queue when front == rear.
34. a — In postfix, evaluation begins from leftmost operator.
35. b — Undo operation uses stack (LIFO).
36. b — Infix → Postfix takes linear time O(n).
37. b — Postfix evaluation processes each token once → O(n).
38. c — Priority queue orders by priority value.
39. a — After dequeue (front=3), two enqueue → rear wraps → rear=1.
40. b — Overflow when pushing into full stack.

--------------------------------------------

📘 END OF SET 2 (UNIT 2: QUESTIONS 21–40)

--------------------------------------------

## 📘 UNIT 2 – STACKS AND QUEUES

--------------------------------------------

(Questions 41–60)

Q41. Which of the following data structures can be used to implement a queue?
a) Stack
b) Linked List
c) Tree
d) Both a and b

Q42. Fill in the blank: In linked list implementation of stack, insertion and deletion are performed at the _______.
a) Rear end
b) Front end
c) Middle
d) Random position

Q43. What will be the output of this code?

```
stack<int> s;
s.push(1);
s.push(2);
s.push(3);
s.pop();
cout << s.top();
```

a) 1
b) 2
c) 3
d) 0

Q44. The time complexity for inserting an element into a stack implemented using linked list is _______.
a) O(n)
b) O(1)
c) O(n log n)
d) $O(n^2)$

Q45. Which of the following is NOT an application of stack?
a) Balancing symbols
b) Undo/Redo
c) CPU scheduling
d) Expression evaluation

Q46. In queue implementation using linked list,
where is insertion performed?
a) Front
b) Rear
c) Both ends
d) None

Q47. What happens when we perform dequeue operation on an empty queue?
a) Overflow

b) Underflow
c) Random output
d) None

Q48. Which data structure is used in level-order traversal of a tree?
a) Stack
b) Queue
c) Linked list
d) Array

Q49. Fill in the blank: In circular queue, when front == (rear + 1) % size, the queue is _______.
a) Empty
b) Full
c) Half full
d) Overflowed

Q50. Which of the following statements is true about queues?
a) Deletion at front, insertion at rear
b) Insertion at front, deletion at rear
c) Insertion and deletion both at front
d) None

Q51. What is the time complexity of push and pop in stack using array?
a) O(1)
b) O(n)
c) O(n log n)
d) $O(n^2)$

Q52. What is the time complexity for enqueue and dequeue operations in a queue implemented using linked list?
a) O(1)
b) O(n)
c) O(log n)
d) $O(n^2)$

Q53. Which of the following statements is FALSE about a priority queue?
a) Highest priority element is deleted first
b) Elements are processed according to arrival time
c) Can be implemented using heaps
d) Used in Dijkstra’s algorithm

Q54. The number of pointers required to represent a circular linked list queue is _______.
a) 1
b) 2
c) 3
d) None

Q55. Fill in the blank: When we use a linked list to implement a stack, we avoid _______ problem of arrays.
a) Overflow
b) Underflow
c) Both a and b
d) None

Q56. Which of the following is not a real-world example of queue?
a) Ticket counter line
b) Printer job queue
c) Browser history
d) Call waiting system

Q57. The postfix expression “6 3 2 + * 4 -” evaluates to _______.
a) 30

b) 25
c) 28
d) 32

Q58. What is the time complexity for deleting the front element of a queue implemented using array?
a) O(1)
b) O(n)
c) $O(n^2)$
d) O(log n)

Q59. Which of the following best describes a double-ended queue (Deque)?
a) A queue that allows insertion/deletion at both ends
b) A stack with two tops
c) A queue with only one end
d) None

Q60. In a circular queue of size 5,
if front = 0 and rear = 4,
and we perform dequeue() twice and enqueue(7), enqueue(8),
then new front and rear are _______.
a) front = 2, rear = 1
b) front = 1, rear = 3
c) front = 2, rear = 0
d) front = 3, rear = 4

--------------------------------------------
🔷 ANSWER KEY – UNIT 2 (Q41–Q60)
--------------------------------------------

41. d — Queue can be implemented using stack or linked list.
42. b — Stack operations occur at front (top) in linked list.
43. b — After pop(), top = 2.
44. b — Linked list insertion/deletion at top → O(1).
45. c — CPU scheduling uses queues, not stacks.
46. b — Queue insertion always at rear.
47. b — Dequeue on empty queue = underflow.
48. b — Level order uses queue (FIFO).
49. b — In circular queue, this condition means full.
50. a — Queue inserts at rear, deletes from front.
51. a — Stack push/pop → constant time O(1).
52. a — Enqueue/dequeue in linked list → O(1).
53. b — Priority queue processes based on priority, not arrival time.
54. b — One pointer for front, one for rear.
55. a — Linked list avoids array overflow (dynamic size).
56. c — Browser history uses stack (LIFO), not queue.
57. c — 6*(3+2)-4 = 6*5-4 = 30-4 = 26 (Correction → answer is 26).
57. (Corrected) — 6*(3+2)-4 = 26.
58. b — Array deletion shifts elements → O(n).
59. a — Deque allows insertion/deletion at both ends.
60. a — After 2 dequeues → front=2; enqueue twice → rear wraps to 1.

--------------------------------------------
📘 END OF SET 3 (UNIT 2: QUESTIONS 41–60)
--------------------------------------------

📘 UNIT 2 – STACKS AND QUEUES
--------------------------------------------
(Questions 61–80)

Q61. Which of the following is the correct prefix form of the expression (A + B) * (C + D)?
a) * + A B + C D

b) + * A B C D
c) A B + * C D +
d) A B C D + + *

Q62. Which of the following infix expressions corresponds to the postfix expression “A B + C D - *”?
a) (A + B) * (C - D)
b) A + (B * (C - D))
c) ((A + B) * C) - D
d) (A * B) + (C - D)

Q63. Fill in the blank: To convert infix to postfix, we use a _______ data structure.
a) Queue
b) Stack
c) Tree
d) Array

Q64. In prefix expression evaluation, operators are scanned from _______.
a) Left to right
b) Right to left
c) Randomly
d) Both directions

Q65. Which of the following expressions is the postfix form of “A * (B + C) / D”?
a) A B C + * D /
b) A B * C + D /
c) A * B C + / D
d) A B C D * + /

Q66. The time complexity of evaluating a postfix expression of n elements using stack is _______.
a) O(1)
b) O(n)
c) O(n log n)
d) O($n^2$)

Q67. Which data structure is used in the evaluation of arithmetic expressions?
a) Queue
b) Stack
c) Tree
d) Graph

Q68. Fill in the blank: Stack memory is used to store _______ in recursion.
a) Parameters and return addresses
b) Static variables
c) Constants
d) File pointers

Q69. What will be the output of this code?

```
stack<int> s;
s.push(5);
s.push(10);
s.pop();
s.push(15);
cout << s.top();
```

a) 5
b) 10
c) 15
d) None

Q70. The time complexity of deleting an element from a queue implemented using linked list is _______.

a) O(1)
b) O(n)
c) O(n log n)
d) O($n^2$)

Q71. Which of the following best describes a circular queue?
a) A queue that connects the last position back to the first
b) A queue that grows dynamically
c) A queue that reverses order
d) A queue with two fronts

Q72. Fill in the blank: In a queue of size n, the maximum number of elements that can be inserted is _______.
a) n
b) n - 1
c) n + 1
d) None

Q73. What is the postfix expression for the infix “(A + B) / (C + D * E)” ?
a) A B + C D E * + /
b) A B + C D + E * /
c) A B C D + * + /
d) A B + C + D E * /

Q74. In the infix to postfix conversion algorithm, when an operator with lower precedence is found, what should be done?
a) Push it directly
b) Pop operators from the stack until precedence is higher
c) Ignore it
d) Replace existing operators

Q75. Which of the following applications uses both stack and queue together?
a) CPU scheduling
b) Recursion
c) Expression parsing
d) None

Q76. Fill in the blank: Queue overflow occurs when _______.
a) front == -1
b) rear == size - 1
c) front == rear
d) rear == NULL

Q77. The auxiliary space required for postfix evaluation of n tokens is _______.
a) O(1)
b) O(n)
c) O(log n)
d) O($n^2$)

Q78. In which case is stack preferred over queue?
a) When elements must be processed in reverse order
b) When elements must be processed in arrival order
c) When priority matters
d) None

Q79. Which of the following is a real-life example of a stack?
a) Ticket booking system
b) Call stack in function execution
c) Print queue
d) CPU scheduling

Q80. Fill in the blank: In an expression tree, leaves represent _______ and internal nodes represent _______.
a) Operators, Operands

b) Operands, Operators
c) Variables, Constants
d) Functions, Loops

--------------------------------------------
🔷 ANSWER KEY – UNIT 2 (Q61–Q80)
--------------------------------------------

61. a — (A + B)*(C + D) → Prefix = * + A B + C D.
62. a — Postfix A B + C D - * → Infix (A + B)*(C - D).
63. b — Stack is used for expression conversion.
64. b — Prefix expressions scanned right to left.
65. a — A*(B+C)/D → Postfix A B C + * D /.
66. b — Each element processed once → O(n).
67. b — Stack handles operators and operands.
68. a — Stack stores parameters, return addresses, local vars in recursion.
69. c — Final top = 15.
70. a — Linked list queue deletion → O(1).
71. a — Circular queue links last to first.
72. b — One cell kept empty → n - 1 usable.
73. a — (A+B)/(C+D*E) → A B + C D E * + /.
74. b — Pop till higher precedence found.
75. c — Expression parsing uses both stack (operators) & queue (operands).
76. b — Queue overflow when rear = size - 1.
77. b — Postfix evaluation needs O(n) auxiliary stack.
78. a — Stack used for reverse order (LIFO).
79. b — Call stack maintains function calls.
80. b — Operands = leaves, operators = internal nodes.

--------------------------------------------
📘 END OF SET 4 (UNIT 2: QUESTIONS 61–80)
--------------------------------------------

📘 UNIT 2 – STACKS AND QUEUES
--------------------------------------------
(Questions 81–100)

Q81. Which of the following statements is TRUE about stacks and queues?
a) Stack is LIFO, Queue is FIFO
b) Stack is FIFO, Queue is LIFO
c) Both are LIFO
d) Both are FIFO

Q82. Fill in the blank: The queue used in Breadth First Search (BFS) algorithm is a _______.
a) Priority Queue
b) Simple Queue
c) Circular Queue
d) Deque

Q83. The postfix expression for the infix “A + B * C – D / E” is _______.
a) A B C * + D E / -
b) A B + C * D E / -
c) A B C + * D E / -
d) A B + C D * / -

Q84. In which case will a circular queue be more efficient than a linear queue?
a) When frequent enqueue and dequeue operations are needed
b) When no elements are deleted
c) When implemented using arrays
d) Never

Q85. The time complexity for checking whether a stack is empty is _______.

a) O(1)
b) O(n)
c) O(n log n)
d) $O(2^n)$

Q86. Which of the following operations is not allowed in a simple queue?
a) Insertion at rear
b) Deletion at front
c) Insertion at front
d) Traversal

Q87. The best data structure to check whether a given string is a palindrome is _______.
a) Stack
b) Queue
c) Tree
d) Graph

Q88. Which of the following code fragments correctly implements stack push operation using an array?
a) top++; stack[top] = value;
b) stack[top++] = value;
c) stack[top] = value; top--;
d) stack[top] = ++value;

Q89. Which of the following is NOT a limitation of using array-based stacks?
a) Fixed size
b) Overflow
c) Random access
d) Underflow

Q90. Fill in the blank: The process of removing all elements from a stack is known as _______.
a) Dequeue
b) Emptying
c) Popping all
d) Stack clearing

Q91. Which of the following is an example of a real-life queue?
a) Browser history
b) Print spooling
c) Undo operation
d) Recursion

Q92. If we reverse the order of operations in a stack, it will behave like a _______.
a) Queue
b) Tree
c) Graph
d) Deque

Q93. Which data structure is used to evaluate a prefix expression?
a) Stack
b) Queue
c) Linked List
d) Array

Q94. Fill in the blank: In a stack implemented using linked list, the “top” pointer points to the _______.
a) Last inserted node
b) First node
c) Null node
d) Second node

Q95. Which of the following represents the correct prefix form of the expression "A – B + C * D"?
a) + - A B * C D
b) - + A B * C D
c) + A - B * C D
d) + - * C D A B

Q96. The auxiliary space used in a recursive function depends on _______.
a) Input data size
b) Recursion depth
c) Compiler
d) Function name

Q97. In a priority queue, which element is deleted first?
a) The smallest element
b) The element with the highest priority
c) The element inserted last
d) The middle element

Q98. The time complexity of reversing a stack using recursion is _______.
a) O(1)
b) O(n)
c) O(n log n)
d) O($n^2$)

Q99. Fill in the blank: In a circular queue, the next position of rear is given by _______.
a) (rear + 1)
b) (rear + 1) % size
c) (rear - 1)
d) size - rear

Q100. Which of the following statements is TRUE?
a) Stack is used in recursion, Queue in scheduling
b) Stack is used in scheduling, Queue in recursion
c) Both used for sorting
d) Both used in BFS

--------------------------------------------

🔷 ANSWER KEY – UNIT 2 (Q81–Q100)

--------------------------------------------

81. a — Stack uses LIFO, Queue uses FIFO.
82. b — BFS uses a simple queue.
83. a — A + B*C – D/E → Postfix: A B C * + D E / -.
84. a — Circular queue efficiently reuses freed spaces.
85. a — Checking empty stack uses O(1) time.
86. c — Simple queue does not allow insertion at front.
87. a — Stack checks palindrome efficiently by reversal.
88. a — Push increments top, then inserts value.
89. c — Random access is not a limitation (stacks don't require it).
90. d — Removing all elements → stack clearing.
91. b — Print spooling follows FIFO principle.
92. a — Reverse LIFO → behaves like FIFO (queue).
93. a — Stack used for evaluating prefix and postfix.
94. a — Top points to the most recently inserted node.
95. a — "A – B + C * D" → Prefix = + - A B * C D.
96. b — Depends on recursion depth (stack frames).
97. b — Highest-priority element deleted first.
98. b — Each element processed once → O(n).
99. b — Circular queue wraps using (rear + 1) % size.
100. a — Stack → recursion; Queue → scheduling tasks.

--------------------------------------------

## 📘 END OF SET 5 (UNIT 2: QUESTIONS 81–100)

--------------------------------------------

## ✅ UNIT 2 COMPLETE — STACKS AND QUEUES (100 QUESTIONS)

------------------------------------------------------

Topics Covered:

- Stack operations (push, pop, overflow, underflow)
- Queue operations (enqueue, dequeue, circular queues)
- Infix, Postfix, Prefix conversions
- Expression evaluation
- Linked list & array implementations
- Real-world applications & complexities