

UNIT II – Question Bank: Stack ADT & Queue ADT

Part A – 2 Marks (Short Answer Questions)

1. 1. Define Stack ADT and list its operations.
2. 2. What is the purpose of the 'top' pointer in a stack?
3. 3. Mention any two real-time applications of stacks.
4. 4. What is the difference between a stack and a queue?
5. 5. Define Queue ADT and list its operations.
6. 6. What is a circular queue?
7. 7. Give an example of a postfix expression.
8. 8. Define double-ended queue (deque).
9. 9. State the use of stack in expression evaluation.
10. 10. What are the limitations of linear queue implementation?
11. 11. Differentiate between linear queue and circular queue.
12. 12. What is postfix evaluation?
13. 13. List the steps for converting infix to postfix expression.
14. 14. What are the applications of deque?
15. 15. Define balanced parentheses with example.

Part B – 13 Marks (Descriptive/Problem Solving Questions)

16. 1. Explain Stack ADT and describe its array implementation with all operations.
17. 2. Write an algorithm to convert an infix expression to postfix expression with an example.
18. 3. Write an algorithm for evaluating a postfix expression and explain it with an example.
19. 4. Discuss the use of stack in checking balanced parentheses. Write the algorithm.
20. 5. Explain function call mechanism using stack. Provide an example to illustrate recursion.
21. 6. Explain Queue ADT and describe its array implementation.
22. 7. Describe the working of circular queue. How does it overcome the limitation of linear queue?
23. 8. Write a C program to implement circular queue using arrays with enqueue and dequeue operations.
24. 9. Compare stack and queue with examples. When should one be preferred over the other?
25. 10. Explain double-ended queue (deque). Write a program to implement input-restricted deque.
26. 11. Describe applications of queue in real-time systems.
27. 12. Write a menu-driven program for stack operations using linked list implementation.

28. 13. Design a menu-driven program to implement a queue using singly linked list.

Part C – 15 Marks (Case Study / Application-Based Questions)

29. 1. Expression Evaluation System: Design a system that takes infix expressions and converts them to postfix form using stack. Further, evaluate the postfix expression and return the result. Justify your choice of data structures and provide implementation details.
30. 2. Compiler Function Tracker: A compiler uses stack to manage function calls and returns. Create a simulation of function calls and returns using a stack. Track recursive function behavior and explain how stack helps in managing memory.
31. 3. Browser History System: Design a browser history navigation system using two stacks – one for back and one for forward navigation. Implement key operations: visit, back, forward.
32. 4. Call Center Task Scheduler: Implement a task queue system for a call center. Tasks must be processed in the order they arrive (FIFO). Choose a suitable queue structure and explain how circular queue improves performance.
33. 5. Simulation of Printer Queue: Model a printer job queue where jobs can be inserted and removed from both ends based on priority. Implement this using double-ended queue and justify its use.
34. 6. Balanced Code Validator: Create a code validator that checks whether parentheses, braces, and brackets are balanced in a given code snippet. Use stack-based approach and explain the logic with implementation.