UNIT I – Question Bank: Data Structures

Part A – 2 Marks (Short Answer Questions)

1. 1. Define Abstract Data Type (ADT) with an example.
2. 2. What is asymptotic notation? Name any three types.
3. 3. State the difference between recursion and iteration.
4. 4. What is tail recursion?
5. 5. Write the time complexity of binary search using recursion.
6. 6. Define singly linked list and give an example.
7. 7. What is the advantage of using a circular linked list over a singly linked list?
8. 8. Define doubly linked list and mention any two applications.
9. 9. What are the drawbacks of array-based list implementation?
10. 10. What is the Big-O notation and where is it used?
11. 11. Mention two use cases of linked list over arrays.
12. 12. Write the general form of a recursive function.
13. 13. What is meant by stack overflow in recursion?
14. 14. State the purpose of header node in a linked list.
15. 15. Mention one real-world application for each: singly linked list and circular linked list.

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

16. 1. Explain the concept of Abstract Data Types (ADT) and discuss how it helps in software development.
17. 2. Describe asymptotic notations (Big O, Ω, and Θ) with examples and corresponding graphs.
18. 3. Write a recursive algorithm to find the factorial of a number and analyze its time complexity.
19. 4. Differentiate between singly, circular, and doubly linked lists with diagrams and examples.
20. 5. Implement a List ADT using array. Explain each operation (insert, delete, search).
21. 6. Describe the advantages and disadvantages of array-based list implementation over linked list.
22. 7. Write a recursive function for Fibonacci series and analyze its time complexity.
23. 8. Implement a doubly linked list with insert at front and delete at rear operations using C.
24. 9. Design a menu-driven program to implement a circular linked list.
25. 10. Compare recursion and iteration. Give scenarios where recursion is preferred.
26. 11. Develop a program to reverse a singly linked list iteratively and recursively.
27. 12. Analyze a recursive algorithm using recurrence relations (e.g., Merge Sort).
28. 13. Write an algorithm to delete a node in a circular singly linked list given its key value.

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

29. 1. Hospital Management System: A hospital uses a dynamic list to maintain patient records in the order of their arrival. Suggest a suitable data structure. Justify your choice. Implement basic operations like add new patient, remove discharged patient, and display patient list using a singly linked list.
30. 2. Music Playlist App: In a music player, users can play songs in a loop (repeat mode). Model this using a circular linked list. Write functions for adding a song, removing a song, and moving to the next song in loop mode.
31. 3. Undo Feature in Text Editor: A text editor implements the undo feature using a doubly linked list where each edit is a node. Explain how this can be implemented. Write pseudo-code to traverse back and forth through the edits.
32. 4. Recursive Analysis: You are designing a recursive solution to solve the Tower of Hanoi problem. Analyze its complexity. Suggest whether it can be implemented non-recursively and if so, how?
33. 5. Library Management System: The system uses a linked list to store books issued to a student. The list should support addition of new books, returning books (deletion), and displaying currently issued books. Develop an implementation using a doubly linked list.
34. 6. Memory Efficient Scheduler: A task scheduler wants to implement a memory-efficient dynamic queue of tasks where tasks can be removed from both ends. Propose a data structure. Justify your choice and implement necessary operations.