1. Data Structures, Classifications, and Operations

Two-Mark Questions:

1. Define a data structure.
2. What is a primitive data structure?
3. Name two non-primitive data structures and give an example of each.
4. List any three common operations performed on data structures.
5. What is the difference between linear and non-linear data structures?

Five-Mark Questions:

1. Explain the classification of data structures with examples.
2. Describe the difference between primitive and non-primitive data structures in detail.
3. What are the main operations on data structures, and how are they useful?
4. How does the choice of data structure impact algorithm efficiency?
5. Compare and contrast arrays, linked lists, stacks, and queues.

2. Arrays, Structures, Self-Referential Structures, and Unions

Two-Mark Questions:

1. Define an array.
2. What is a structure in C, and how does it differ from an array?
3. What is a union?
4. Give an example of a self-referential structure.
5. How are arrays represented in memory?

Five-Mark Questions:

1. Explain the difference between arrays and structures with examples.
2. Describe the concept of unions and discuss how they save memory.
3. What is a self-referential structure? Explain with code.
4. Compare arrays and dynamically allocated arrays.
5. Discuss the applications and limitations of unions in C.

3. Pointers and Dynamic Memory Allocation

Two-Mark Questions:

1. Define a pointer.
2. What does malloc() do?
3. How does calloc() differ from malloc()?
4. What is a dangling pointer?
5. Explain the purpose of the `free()` function.

Five-Mark Questions:

1. Describe how pointers are used in dynamic memory allocation.
2. Explain the differences between malloc(), calloc(), and realloc().
3. What are memory leaks and dangling pointers? How can they be avoided?
4. How do dynamically allocated arrays differ from static arrays?
5. Discuss the importance of freeing dynamically allocated memory with examples.

4. Performance Analysis and Algorithm Complexities

Two-Mark Questions:

1. Define time complexity.
2. What is space complexity?
3. What is asymptotic notation?
4. Explain the best-case complexity of an algorithm.
5. Define Big O notation.

Five-Mark Questions:

1. Explain time complexity and its significance in algorithm analysis.
2. Describe space complexity with an example.
3. Compare Big O, Big Theta, and Big Omega notations.
4. How would you analyze the performance of an algorithm?
5. Discuss the difference between worst-case, best-case, and average-case complexities with examples.

5. Stacks and Stack Applications

Two-Mark Questions:

1. Define a stack.
2. What is a stack overflow?
3. List two applications of stacks.
4. What is postfix notation?
5. Explain a practical use of stack in programming.

Five-Mark Questions:

1. Describe stack operations: push, pop, and peek.
2. How is a stack represented using arrays?
3. Explain how stacks are used in infix to postfix conversion.
4. What is a postfix expression? Provide an example.
5. Describe how stacks handle function call recursion.

6. Recursion

Two-Mark Questions:

1. Define recursion.
2. What is a base case in recursion?
3. Write a recursive definition of the factorial function.
4. How does recursion differ from iteration?
5. Explain one disadvantage of recursion.

Five-Mark Questions:

1. Describe the process of calculating the GCD of two numbers using recursion.
2. Explain how recursion works with the Tower of Hanoi problem.
3. How is recursion used in calculating Fibonacci numbers?
4. Compare recursion and iteration with examples.
5. Discuss the role of the call stack in recursion.

7. Queues

Two-Mark Questions:

1. Define a queue.
2. What is a circular queue?
3. Explain the concept of dequeue.
4. List two applications of queues.
5. What is the difference between a queue and a stack?

Five-Mark Questions:

1. Describe the operations performed on a linear queue.
2. Explain the working of a circular queue and its advantage over a linear queue.
3. How are priority queues implemented?
4. What are circular queues, and how do they solve queue overflow issues?
5. Discuss the advantages and limitations of priority queues.

8. Linked Lists

Two-Mark Questions:

1. Define a linked list.
2. What is a node in a linked list?
3. Explain the purpose of a header node.
4. How does a doubly linked list differ from a singly linked list?
5. What is a circular linked list?

Five-Mark Questions:

1. Describe the different types of linked lists.
2. Explain linked list traversal with an example.
3. How are insertion and deletion performed in a doubly linked list?
4. Compare circular linked lists and header linked lists.
5. What are the benefits of using linked lists over arrays?

9. Searching and Sorting Algorithms

Two-Mark Questions:

1. What is binary search?
2. Define selection sort.
3. What is the time complexity of bubble sort?
4. How does interpolation search differ from binary search?
5. Explain the concept of radix sort.

Five-Mark Questions:

1. Describe the process of quick sort with an example.
2. Explain merge sort and its time complexity.
3. Compare insertion sort and bubble sort.
4. Discuss the advantages of radix sort over other sorting techniques.
5. Describe the working of interpolation search and its applications.

10. Trees

Two-Mark Questions:

1. Define a binary tree.

2. What is a leaf node?
3. What is in-order traversal?
4. Explain the concept of a binary search tree.
5. What is a threaded binary tree?

Five-Mark Questions:

1. Describe the properties of binary trees.
2. Compare in-order, pre-order, and post-order traversals.
3. Explain binary tree representation using arrays and linked lists.
4. What is a binary search tree? Describe its insertion and deletion operations.
5. Describe threaded binary trees and their applications.

11. Binary Search Trees, Red-Black Trees, and AVL Trees

Two-Mark Questions:

1. Define an AVL tree.
2. What is the purpose of balancing in AVL trees?
3. List the properties of a red-black tree.
4. Define a binary search tree.
5. Explain the concept of tree rotations.

Five-Mark Questions:

1. Describe the properties of an AVL tree and how it is balanced.
2. What are the balancing operations performed in a red-black tree?
3. Explain the insertion process in a binary search tree.
4. Compare AVL trees and red-black trees.
5. Discuss the role of binary search trees in data retrieval.

12. Hashing

Two-Mark Questions:

1. Define hashing.
2. What is a hash function?
3. What is a hash collision?
4. Explain chaining in hashing.
5. Define open addressing.

Five-Mark Questions:

1. Describe the purpose of hashing in data structures.
2. Explain how collisions are handled in open addressing.
3. Compare static and dynamic hashing.
4. What are the properties of a good hash function?
5. Explain hashing with chaining and its advantages.

13. Graphs

Two-Mark Questions:

1. Define a graph.
2. What is an adjacency matrix?

3. Differentiate between directed and undirected graphs.
4. Define BFS (Breadth-First Search).
5. What is a cycle in a graph?

Five-Mark Questions:

1. Explain the adjacency list and adjacency matrix representations of graphs.
2. Describe Depth-First Search and its applications.
3. How does Breadth-First Search work? Give an example.
4. Compare and contrast DFS and BFS.
5. Discuss the applications of graphs in computer science.