UNIT 1

Short Questions [3 marks]

Q.1	Define primitive and non-primitive data structures and explain with an example.
Q.2	Define time complexity and its importance in algorithm analysis and explain with an example.
Q.3	Write the difference between space complexity and time complexity with an example.
Q.4	Explain the push and pop operations in a stack data structure with syntax and example.
Q.5	How to represent linear arrays in a memory? Explain with an example.
Q.6	Describe the role of pointers in dynamic memory allocation. Give an example in C demonstrating the allocation and deallocation of memory using pointers.
Q.7	Compare arrays and structures. When would you use an array, and when would you prefer to use a structure? Provide examples to justify your choice.

Long Questions [5 marks]

Q.1	Describe the concept of queue data structure and its application. Explain the operations on a queue, such as enqueue and dequeue with a syntax and example.
Q.2	Describe the process of dynamically allocating memory for arrays. Discuss common issues like memory leaks and how they can be prevented or managed.
Q.3	How to represent a Linear array in a memory? Discuss the advantages and challenges of using dynamic arrays compared to static arrays with an example.
Q.4	Define self-referential structures and unions and explain with an example also differentiate between self-referential structures and unions.

UNIT 2

Short Questions [3 marks]

Q.1	Describe the process of converting an infix arithmetic expression to postfix notation using a stack. Explain the advantages of postfix notation and provide an example to demonstrate the conversion process step-by-step.
Q.2	What is a circular queue and a linear queue? Explain the difference between a circular queue and a linear queue.
Q.3	Explain the operations on a queue. Discuss how queues are utilized in scenarios requiring First In First Out (FIFO) data processing, explain with an example.
Q.4	What is a recursion? Explain Tower of Hanoi with the recursion method.
Q.5	Describe the characteristics of a queue. Explain its advantages and disadvantages.
Q.6	What are the common operations that can be performed on a priority queue? Explain a few of these operations in detail.
Q.7	Explain how the performance of a priority queue is affected when implemented using an unsorted array compared to a sorted array.

Long Questions [5 marks]

Q.1	What is a priority queue and explain its typical operations. Provide an example of a priority queue.
Q.2	Discuss the concept of Polish notation (prefix notation) and its significance. Provide an example of an arithmetic expression in Polish notation and explain how it can be evaluated using a stack.
Q.3	Discuss the key characteristics of stack data structures. Explain how stacks facilitate Last In First Out (LIFO) operations with an example.
Q.4	Explain the advantages and limitations of using recursion with an example.

UNIT 3

Short Questions [3 marks]

Q.1	What is an array and linked list? Explain different types of linked lists.
Q.2	Explain the difference between singly linked list and doubly linked list with an example.
Q.3	Explain what is a linked list and how it differs from an array in terms of memory allocation and access time.
Q.4	What is the difference between traversing a singly linked list and an array?
Q.5	What is a circular linked list? How does it manage the end of the list differently compared to a singly linked list?
Q.6	How does a circular linked list differ from a doubly linked list?
Q.7	Imagine you have a music playlist where you can add new songs anywhere and remove unwanted ones. What data structure (linked list or array) would be more suitable and why?

Long Questions [5 marks]

Q.1	Compare and contrast the three main types of linked lists (singly, circular, doubly) in terms of structure, operation, and memory usage.
Q.2	Explain the concept of a head pointer and tail pointer in a linked list and its importance.
Q.3	Describe the process of deleting a node from the middle of a singly linked list with an algorithm and explain with an example.
Q.4	Search operation in a linked list is generally slower than an array. Explain why this is the case.

UNIT 4

Short Questions [3 marks]

Q.1	What is sorting? List out different types of sorting methods and explain any of it.
Q.2	Describe how searching methods work? Explain Linear and Binary search with an example.
Q.3	Explain the process of bubble sort with an algorithm.
Q.4	Explain the process of radix sort with an algorithm.
Q.5	Describe the basic steps involved in quick sort.
Q.6	Explain the difference between selection sort and insertion sort with an example.
Q.7	Explain merge sort and quick sort with an example.

Long Questions [5 marks]

Q.1	What are the advantages and disadvantages of merge sort? Explain why it might be preferred over simpler sorting algorithms.
Q.2	Describe in detail the process of linear search and binary search.

Q.3	Perform selection sort and insertion sort in given array:
	I. A[6] = {22,25,28,17,12,1}
	II. Z[5] = {11,99,12,43,6}

Q.4	Perform radix sort and bubble sort in given array:
	I. A[6] = {522,25,228,17,112,1}
	II. Z[5] = {101,99,912,43,6}

Q.5	Perform quick sort and merge sort in given array:
	I. A[6] = {522,25,228,17,112,1}
	II. Z[5] = {101,99,912,43,6}