Quiz Answers

**Section 1:**

1. **In a stack, what is the result of applying the following sequence of operations: PUSH(1), PUSH(2), POP, PUSH(3), POP, POP, PUSH(4)? Answer: The stack contains [4].**
2. **In a queue, what is the purpose of a circular queue? Answer: To efficiently use the available memory and avoid wasting space.**
3. **Which of the following is NOT a typical application of a priority queue? Answer: d) Undo functionality in a text editor.**
4. **Which of the following statements about a deque (double-ended queue) is correct? Answer: b) In a deque, elements can be added or removed from both ends.**
5. **What is the time complexity of finding the maximum element in a priority queue implemented as a binary heap? Answer: a) O(log n)**

**Section 2:**

1. **What is the primary advantage of using a stack data structure? Answer: The primary advantage of using a stack data structure is its ability to efficiently manage function calls and maintain a Last-In-First-Out (LIFO) order, making it useful for tasks like function call management, expression evaluation, and backtracking.**
2. **What data structure follows the First-In-First-Out (FIFO) principle? Answer: A queue data structure follows the First-In-First-Out (FIFO) principle, where the element that is added first is the one to be removed first.**
3. **What is a priority queue, and how does it differ from a regular queue? Answer: A priority queue is a data structure that stores elements along with their associated priorities and allows for the retrieval of elements with the highest priority first. It differs from a regular queue in that elements are not necessarily processed in the order they were added but rather in order of priority.**
4. **How can you efficiently find the length of a singly linked list without traversing the entire list? Answer: You can efficiently find the length of a singly linked list by maintaining a count variable while traversing the list during insertions and deletions. This count variable represents the length of the list, and you can update it whenever you add or remove elements from the list. This way, you don't need to traverse the entire list to find its length.**
5. **What is the fundamental difference between an array and a linked list? Answer: The fundamental difference between an array and a linked list is their underlying data structure. An array is a contiguous block of memory that stores elements of the same data type, and its size is typically fixed. In contrast, a linked list is a data structure composed of nodes, where each node contains data and a reference (or link) to the next node in the sequence. Linked lists can dynamically grow or shrink in size.**