**Exercise 5: Task Management System**

1. **Understanding Linked Lists:**
   * **Q) Explain the different types of linked lists (Singly Linked List, Doubly Linked List).**
     + **Ans)**
       - **Singly Linked List**: In a singly linked list, each node points to the next node in the sequence. Each node contains data and a reference (or pointer) to the next node. The first node is called the head, and the last node points to null, indicating the end of the list.
       - **Doubly Linked List**: In a doubly linked list, each node has an additional pointer that points to the previous node in the sequence. This means the nodes are connected in both directions—forward and backward. Each node contains data, a reference to the next node, and a reference to the previous node.
2. **Analysis:**
   * **Q) Analyze the time complexity of each operation.**
     + **Ans)**
       - **Add**: O(n) — Adding a task to the end of the list requires traversing to the end.
       - **Search**: O(n) — Searching for a task requires traversing until the task is found.
       - **Traverse**: O(n) — Traversing visits each node once.
       - **Delete**: O(n) — Deletion requires searching for the task and updating pointers.
   * **Q) Discuss the advantages of linked lists over arrays for dynamic data.**
     + **Ans)**
       - **Dynamic Size**: Linked lists can grow and shrink dynamically, eliminating the need to declare a predefined size.
       - **Efficient Insertions/Deletions**: Inserting or deleting an element at the start or in the middle of a linked list is more time-efficient compared to arrays because elements do not need to be shifted.
       - **Memory Consumption**: Linked lists use memory more efficiently for dynamic data because they do not waste extra memory reserved by the application.