**Exercise 5: Task Management System**

**Scenario:**

You are developing a task management system where tasks need to be added, deleted, and traversed efficiently.

**Steps:**

1. **Understand Linked Lists:**

Explain the different types of linked lists (Singly Linked List, Doubly Linked List).

**Singly Linked List:**

* A singly linked list is a data structure consisting of a sequence of elements, each containing a reference to the next element in the sequence.
* Each node contains two fields: one for storing the data and another for storing the reference to the next node.
* The last node in the list points to null.

**Doubly Linked List:**

* A doubly linked list is similar to a singly linked list but with an additional reference in each node to the previous node.
* Each node contains three fields: one for storing the data, one for the next node, and one for the previous node.
* This allows traversal in both forward and backward directions.

1. **Setup:**

A class **Task** is created with attributes like **taskId**, **taskName**, and **status**. The code is provided in the wordpad

1. **Implementation:**

The implementation code is provided in the wordpad**.**

1. **Analysis:**
   1. **Analyze the time complexity of each operation.**

Time Complexity:

Add Task: O(n) in the worst case (traversing to the end of the list), O(1) if adding at

the beginning.

Search Task: O(n) in the worst case (traversing the entire list).

Traverse Tasks: O(n) (visiting each node).

Delete Task: O(n) in the worst case (traversing to find the node).

* 1. **Discuss the advantages of linked lists over arrays for dynamic data**.

Advantages of Linked Lists Over Arrays:

Dynamic Size: Linked lists grow and shrink dynamically, providing better flexibility compared to arrays with a fixed size.

Efficient Insertions/Deletions: Insertions and deletions are more efficient (O(1) if the position is known) compared to arrays, where shifting elements can be costly (O(n)).

Memory Utilization: Linked lists do not require contiguous memory allocation, making them more efficient in terms of memory usage compared to arrays.

This Java implementation allows for efficient management of tasks with the flexibility of linked lists.