**Exercise 5: Task Management System**

**Linked Lists**

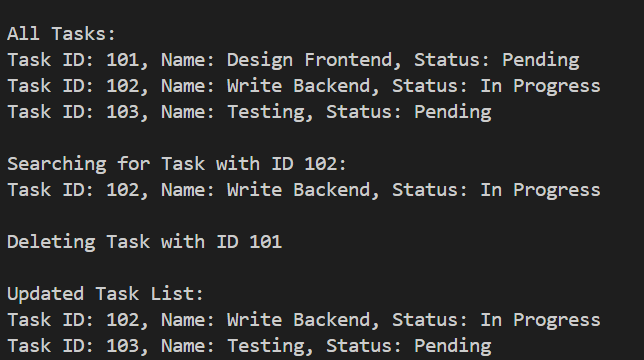
**Singly Linked List:**

* A data structure where each node contains data and a reference (or pointer) to the **next** node.
* Traversal is **one-directional** (from head to tail).

**Doubly Linked List:**

* Each node contains data, a **next** pointer, and a **previous** pointer.
* Allows traversal in **both directions** (forward and backward).
* Requires **more memory** than singly linked lists.

**Output**



### Time Complexity

* **Add Task – O(n):**  
  To add a task, we must go through the list to find the end.
* **Search Task – O(n):**  
  We check each task one by one until we find a match.
* **Traverse Tasks – O(n):**  
  We go through all tasks from start to end.
* **Delete Task – O(n):**  
  We need to locate the task before removing it, which takes time.

### Why Linked Lists are better than Arrays:

* **Dynamic Size:**  
  Linked lists can grow or shrink as needed, while arrays cannot change size after creation.
* **Insertion/Deletion:**  
  Linked lists allow easier and faster insertion or deletion when the position is known, unlike arrays that may need shifting.
* **Memory Usage:**  
  Linked lists use memory only when needed; arrays reserve a fixed amount in advance.