Exercise 5: Task Management System

1. Understanding Linked Lists:

Linked Lists:

- A linked list is a dynamic data structure where each element (node) points to the next.

- Unlike arrays, linked lists do not require contiguous memory blocks, allowing them to grow and shrink dynamically.

Types of Linked Lists:

1. Singly Linked List:

- Each node contains data and a reference to the next node.

- Simple structure, low memory overhead.

- Operations such as insertion and deletion at the head are efficient.

- Cannot traverse backwards.

2. Doubly Linked List:

- Each node contains data, a reference to the next node, and a reference to the previous node.

- Allows traversal in both directions.

- Slightly higher memory usage due to the additional pointer.

3. Circular Linked List:

- Last node points back to the first node.

- Can be singly or doubly circular.

When to Use Linked Lists:

- When data size changes frequently.

- When insertions and deletions are frequent (especially at the start).

- When random access is not needed.

Key Differences from Arrays:

- Arrays require resizing or declaration of a fixed size; linked lists grow as needed.

- Arrays provide O(1) access by index; linked lists require O(n) traversal.

- Arrays can waste memory if preallocated; linked lists allocate memory per node.