**Exercise 5: Task Management System**

**1. Understand Linked Lists**

**Singly Linked List** is a linear data structure where each node points to the next node. It allows dynamic memory usage, as nodes can be created or removed without shifting other elements.

**Doubly Linked List** is similar but with two pointers in each node: one to the next node and one to the previous. This allows traversal in both directions, but with more memory overhead due to the extra pointer.

In this task, a **singly linked list** is sufficient and more memory-efficient.

**2. Analysis**

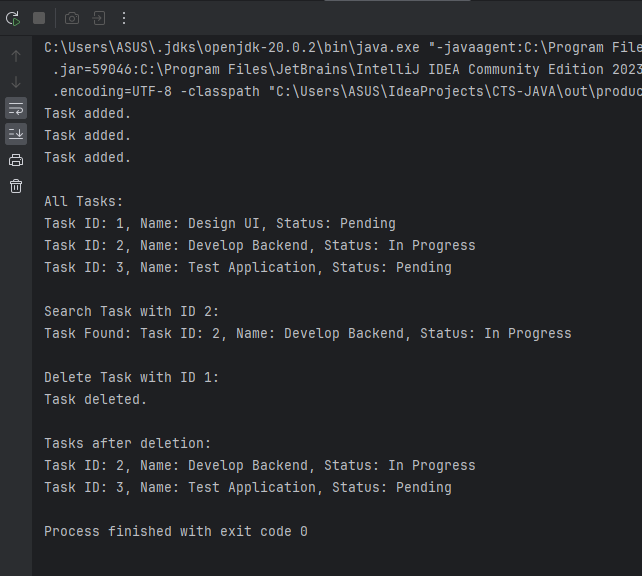
* **Add Operation:** Adding a task at the end of the list takes **O(n)** time, as it involves traversing the list to the last node.
* **Search Operation:** Searching for a task requires scanning the list node by node, which also takes **O(n)** time in the worst case.
* **Traverse Operation:** Displaying all tasks involves visiting each node once, which is **O(n)**.
* **Delete Operation:** Deletion requires locating the task first, then adjusting links, resulting in **O(n)** time.

**Advantages of Linked Lists over Arrays**

* Linked lists allow **dynamic memory allocation**, so tasks can be added or removed without worrying about array size.
* No need to shift elements when adding or deleting in the middle, unlike arrays.
* Memory is allocated only when needed, avoiding wasted space.

However, linked lists do not support direct access to elements by index, which can make certain operations slower compared to arrays.

**3. Output**

****