**DIGITAL NURTURE 4.0 DEEP SKILLING JAVA FSE-WEEK1**

**NAME: SIVITHA GUNASEKARAN**

**SUPERSET ID: 6413354**

**WEEK 1: ALGORITHM DATA STRUCTURES**

**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).
2. **Setup:**
   * Create a class **Task** with attributes like **taskId**, **taskName**, and **status**.
3. **Implementation:**
   * Implement a singly linked list to manage tasks.
   * Implement methods to **add**, **search**, **traverse**, and **delete** tasks in the linked list.
4. **Analysis:**
   * Analyze the time complexity of each operation.
   * Discuss the advantages of linked lists over arrays for dynamic data.

**CODE SAMPLES:**

class Task {

int taskId;

String taskName;

String status;

Task next;

public Task(int id, String name, String status) {

this.taskId = id;

this.taskName = name;

this.status = status;

this.next = null;

}

public String toString() {

return taskId + " | " + taskName + " | " + status;

}

}

public class TaskManagementLinkedList {

Task head = null;

public void addTask(int id, String name, String status) {

Task newTask = new Task(id, name, status);

if (head == null) head = newTask;

else {

Task temp = head;

while (temp.next != null) temp = temp.next;

temp.next = newTask;

}

System.out.println("Added: " + newTask);

}

public void traverse() {

System.out.println("\nAll Tasks:");

Task temp = head;

while (temp != null) {

System.out.println(temp);

temp = temp.next;

}

}

public void searchTask(int id) {

Task temp = head;

while (temp != null) {

if (temp.taskId == id) {

System.out.println("Found: " + temp);

return;

}

temp = temp.next;

}

System.out.println("Task not found.");

}

public void deleteTask(int id) {

if (head == null) return;

if (head.taskId == id) {

head = head.next;

System.out.println("Deleted task with ID: " + id);

return;

}

Task temp = head;

while (temp.next != null) {

if (temp.next.taskId == id) {

temp.next = temp.next.next;

System.out.println("Deleted task with ID: " + id);

return;

}

temp = temp.next;

}

System.out.println("Task not found.");

}

public static void main(String[] args) {

TaskManagementLinkedList list = new TaskManagementLinkedList();

list.addTask(1, "Design UI", "Pending");

list.addTask(2, "Develop API", "In Progress");

list.addTask(3, "Test Feature", "Pending");

list.traverse();

list.searchTask(2);

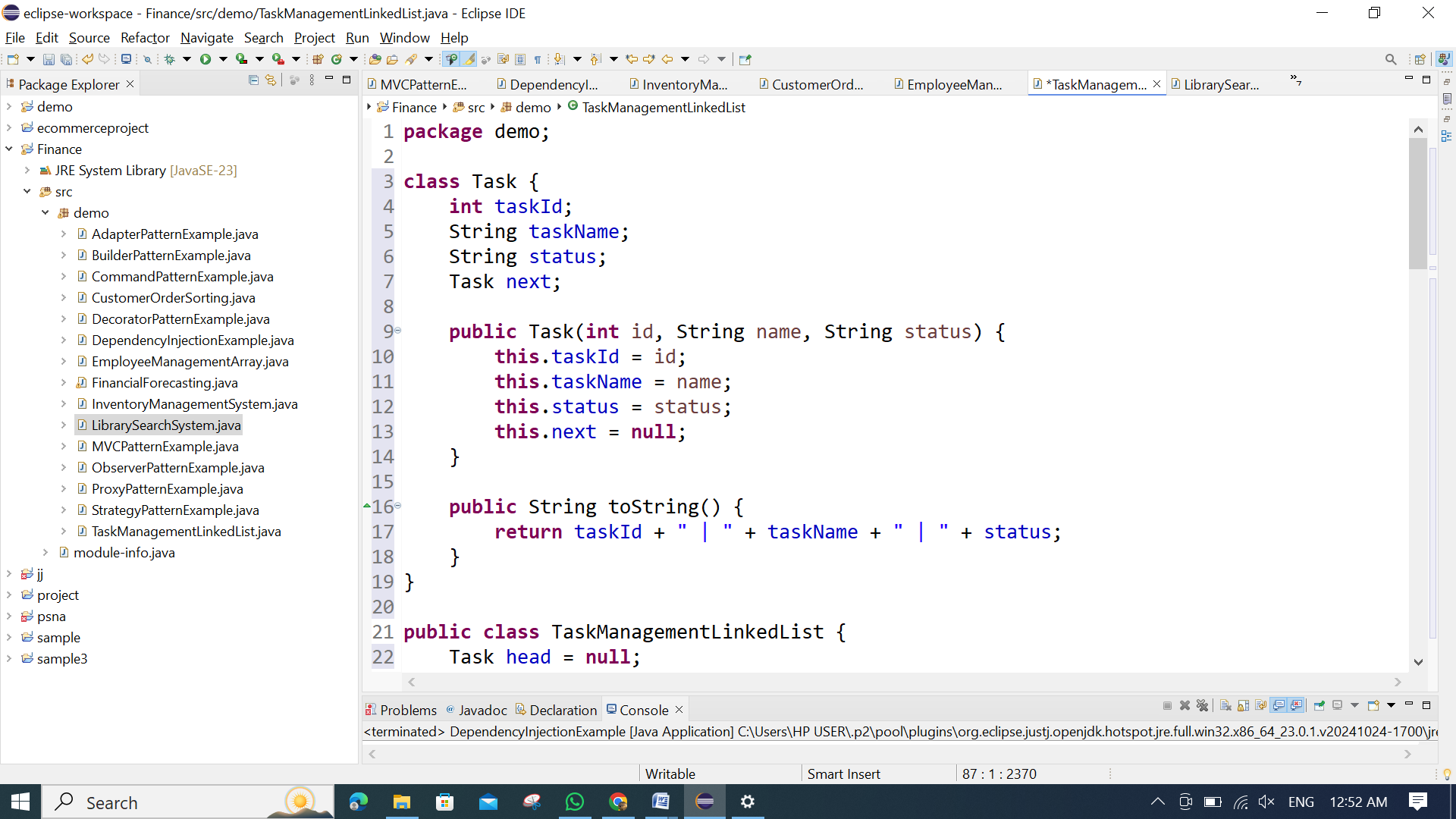
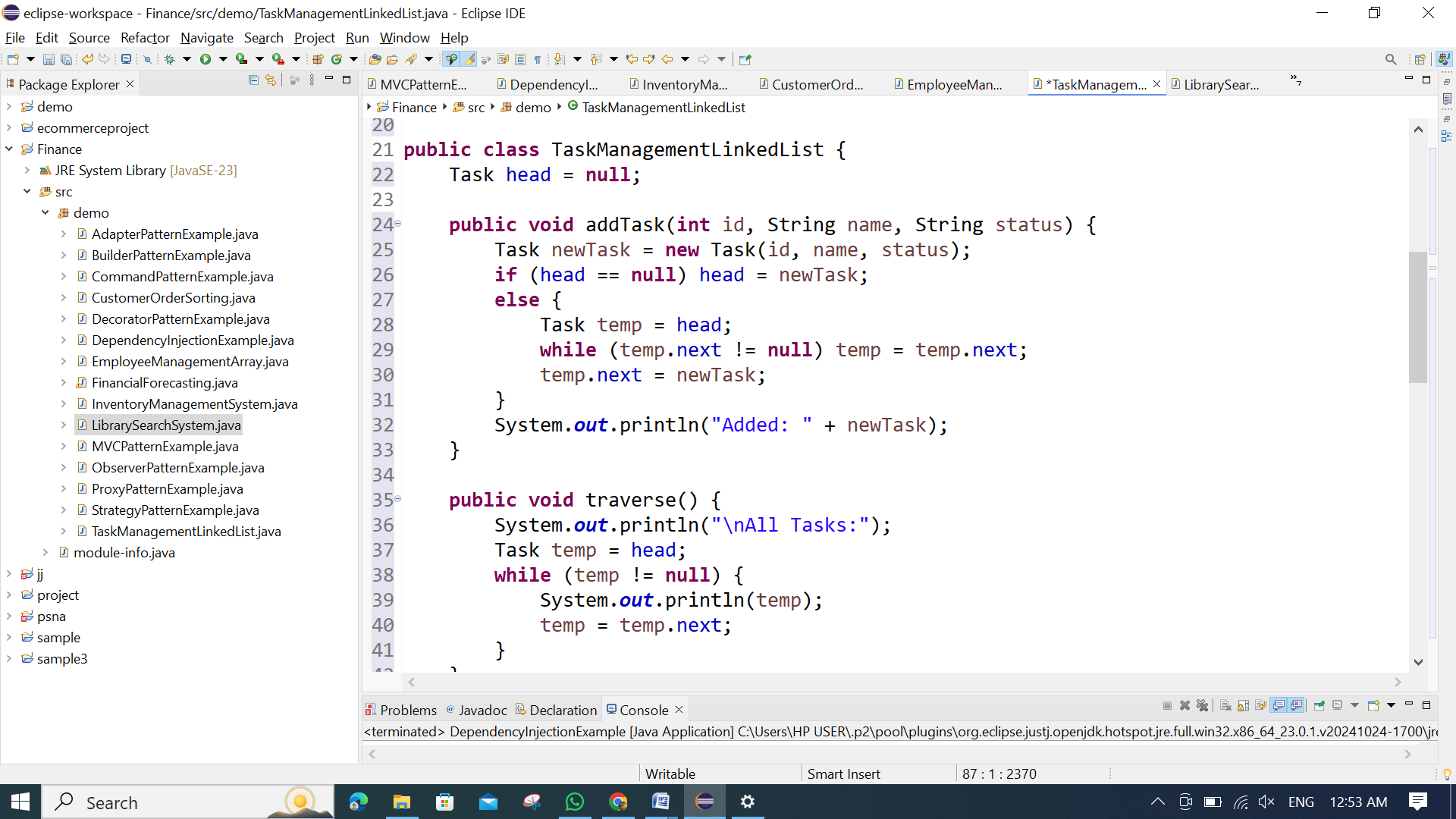
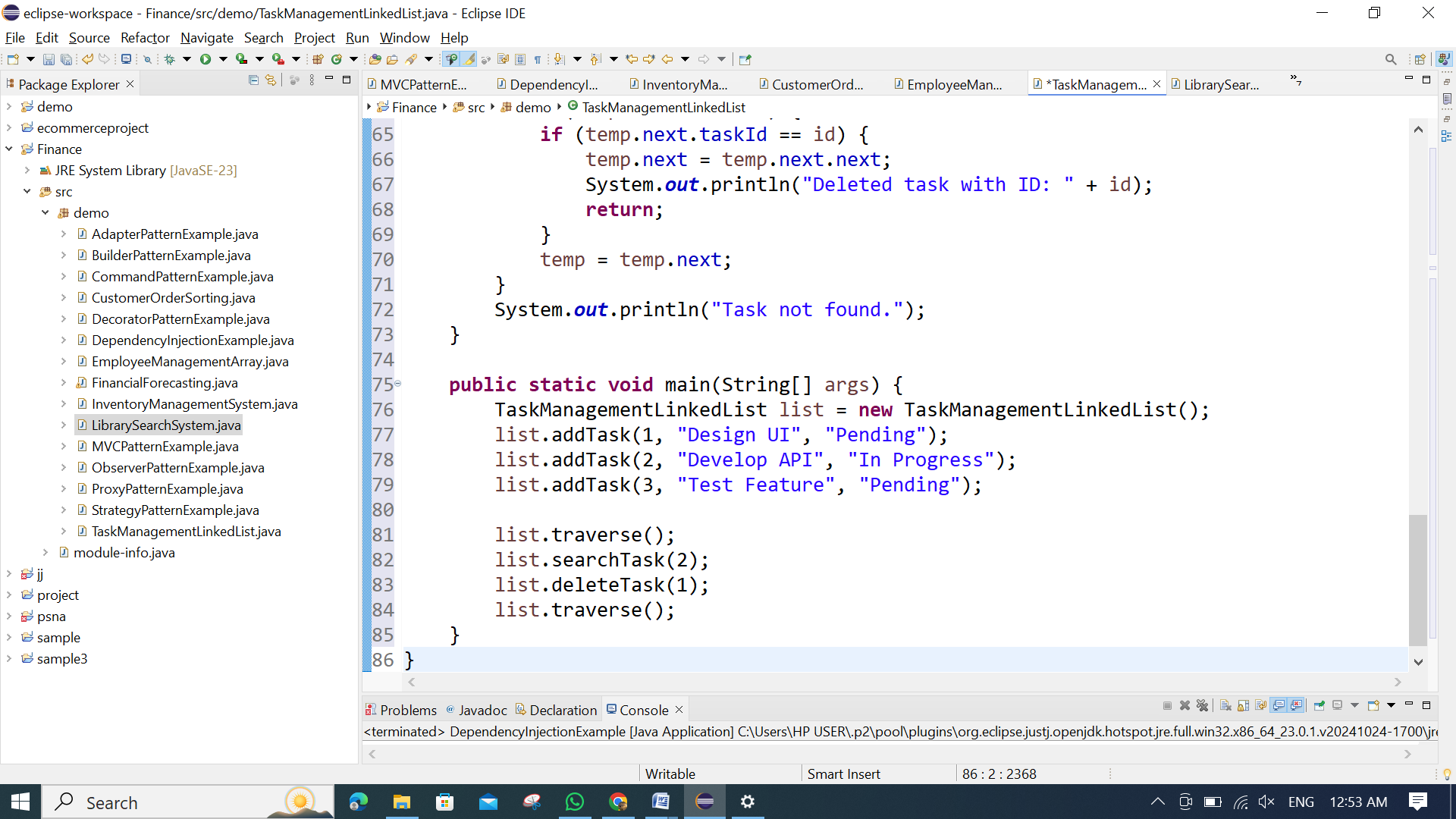
list.deleteTask(1);

list.traverse();

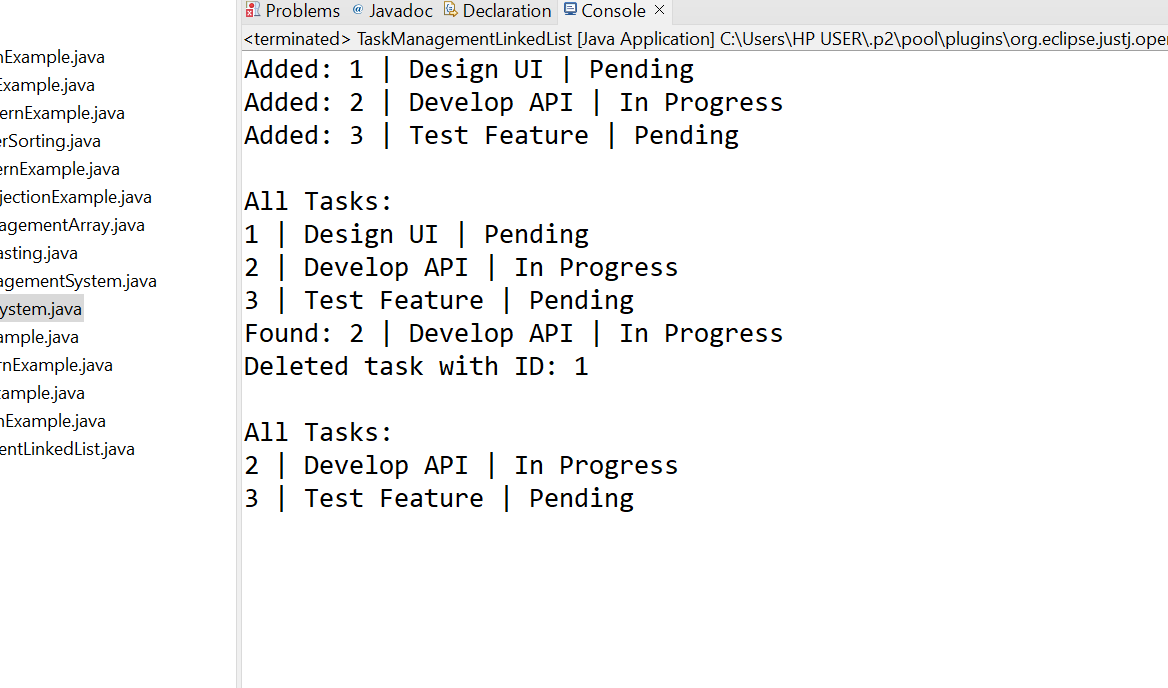
}

}

**MY SCREENSHOT PROOF:**

**  **

**OUTPUT:**

****