```
package HW4;

import java.util.*;

class Task {
    int id;
    String name;
    int priority;

    public Task(int id, String name, int priority) {
        this.id = id;
        this.name = name;
        this.priority = priority;
    }

    public void displayTaskDetails() {
        System.out.println("Task ID: " + id + ", Name: \"" + name + "\", Priority:
" + priority);
    }
}

class TaskManager {
    private List<Task> tasksArray = new ArrayList<>();
    private Queue<Task> taskQueue = new LinkedList<>();
    private Stack<Task> completedTasks = new Stack<>();
    private LinkedList<Task> highPriorityTasks = new LinkedList<>();

    public void inputTasks() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of tasks: ");
        int numTasks = scanner.nextInt();
        scanner.nextLine();

        for (int i = 0; i < numTasks; i++) {
            System.out.print("Enter task ID: ");
            int id = scanner.nextInt();
            scanner.nextLine();

            System.out.print("Enter task name: ");
            String name = scanner.nextLine();

            System.out.print("Enter task priority (1-5): ");
            int priority = scanner.nextInt();
            scanner.nextLine();

            Task task = new Task(id, name, priority);
            tasksArray.add(task);

            if (priority == 1) {
                highPriorityTasks.add(task);
            } else {
                taskQueue.add(task);
            }
        }
    }

    public void displayTasks() {
        System.out.println("\nList of All Tasks:");
        for (Task task : tasksArray) {
```

```
            task.displayTaskDetails();
        }
    }

    public void processTasks() {
        System.out.println("\nTasks in queue (Priority 1 tasks handled first):");

        while (!highPriorityTasks.isEmpty()) {
            Task highPriorityTask = highPriorityTasks.removeFirst();
            System.out.println("Processed Task: " + highPriorityTask.name);
            completedTasks.push(highPriorityTask);
        }

        while (!taskQueue.isEmpty()) {
            Task task = taskQueue.poll();
            System.out.println("Processed Task: " + task.name);
            completedTasks.push(task);
        }
    }

    public void displayTaskHistory() {
        System.out.println("\nTask History (Most Recent First):");
        while (!completedTasks.isEmpty()) {
            Task task = completedTasks.pop();
            task.displayTaskDetails();
        }
    }

    public void simulateTaskManager() {
        inputTasks();
        displayTasks();
        processTasks();
        displayTaskHistory();
    }
}
```