import java.util.\*;

class Task {

    String id, description;

    int priority;

    public Task(String id, String description, int priority) {

        this.id = id;

        this.description = description;

        this.priority = priority;

    }

    @Override

    public String toString() {

        return "[ID: " + id + ", " + description + ", Priority: " + priority + "]";

    }

}

class TaskManager {

    PriorityQueue<Task> taskQueue = new PriorityQueue<>(Comparator.comparingInt(t -> -t.priority));

    Map<String, Task> taskMap = new HashMap<>();

    void addTask(String id, String description, int priority) {

        if (taskMap.containsKey(id)) {

            System.out.println("Task ID already exists!");

            return;

        }

        Task task = new Task(id, description, priority);

        taskQueue.add(task);

        taskMap.put(id, task);

        System.out.println("Task added.");

    }

    void removeTask(String id) {

        if (taskMap.containsKey(id)) {

            taskQueue.remove(taskMap.remove(id));

            System.out.println("Task removed.");

        } else {

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

        }

    }

    void getHighestPriorityTask() {

        if (taskQueue.isEmpty()) System.out.println("No tasks available.");

        else System.out.println("Highest Priority Task: " + taskQueue.peek());

    }

    void displayTasks() {

        if (taskQueue.isEmpty()) System.out.println("No tasks available.");

        else taskQueue.forEach(System.out::println);

    }

}

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        TaskManager manager = new TaskManager();

        while (true) {

            System.out.println("\n1. Add Task  2. Remove Task  3. View Highest  4. Display All  5. Exit");

            System.out.print("Choose: ");

            int choice = sc.nextInt();

            sc.nextLine();

            switch (choice) {

                case 1:

                    System.out.print("Enter ID: ");

                    String id = sc.nextLine();

                    System.out.print("Enter Description: ");

                    String desc = sc.nextLine();

                    System.out.print("Enter Priority: ");

                    int priority = sc.nextInt();

                    manager.addTask(id, desc, priority);

                    break;

                case 2:

                    System.out.print("Enter ID to remove: ");

                    manager.removeTask(sc.nextLine());

                    break;

                case 3:

                    manager.getHighestPriorityTask();

                    break;

                case 4:

                    manager.displayTasks();

                    break;

                case 5:

                    System.out.println("Exiting...");

                    sc.close();

                    return;

                default:

                    System.out.println("Invalid choice.");

            }

        }

    }

}