CSW ASSIGNMENT 2

PART A

Q1.
```
import java.util.Scanner;
import java.util.TreeSet;

public class TreeSetExample {
    public static void main(String[] args) {
        // Create a TreeSet of Integer type
        TreeSet<Integer> treeSet = new TreeSet<>();

        // Add some elements to the set
        treeSet.add(10);
        treeSet.add(20);
        treeSet.add(30);
        treeSet.add(40);
        treeSet.add(50);

        // Display the TreeSet
        System.out.println("TreeSet: " + treeSet);

        // Ask the user to enter a number and search for it
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number to search: ");
        int num = scanner.nextInt();
        if (treeSet.contains(num)) {
            System.out.println(num + " is present in the TreeSet.");
        } else {
            System.out.println(num + " is not present in the TreeSet.");
        }

        // Remove an element from the TreeSet
        System.out.print("Enter an element to remove from the TreeSet: ");
        int removeNum = scanner.nextInt();
        if (treeSet.remove(removeNum)) {
            System.out.println(removeNum + " was removed from the TreeSet.");
            System.out.println("Updated TreeSet: " + treeSet);
        } else {
            System.out.println(removeNum + " was not found in the TreeSet.");
        }
    }
}
```

Q2.
```
import java.util.TreeMap;

class Address {
    private String plotNo;
    private String area;
    private String postOffice;

    public Address(String plotNo, String area, String postOffice) {
        this.plotNo = plotNo;
        this.area = area;
        this.postOffice = postOffice;
    }

    public String getPlotNo() {
        return plotNo;
    }

    public String getArea() {
        return area;
    }

    public String getPostOffice() {
        return postOffice;
    }

    public String toString() {
        return "Plot No: " + plotNo + ", Area: " + area + ", Post Office: " + postOffice;
    }
}

public class TreeMapExample {
    public static void main(String[] args) {
        // Create a TreeMap to store addresses with person names as keys
        TreeMap<String, Address> addressBook = new TreeMap<String, Address>();

        // Insert addresses in the TreeMap
        addressBook.put("John", new Address("123", "Park Street", "Kolkata"));
        addressBook.put("Mary", new Address("456", "Lake Avenue", "New York"));
        addressBook.put("Peter", new Address("789", "Garden Road", "London"));

        // Display the TreeMap
        System.out.println("Address Book:");
        for (String name : addressBook.keySet()) {
            System.out.println(name + ": " + addressBook.get(name));
        }
    } }
```

Q3.
```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Comparator;
import java.util.PriorityQueue;

class Process {
    private int id;
    private int burstTime;
    private int remainingTime;

    public Process(int id, int burstTime) {
        this.id = id;
        this.burstTime = burstTime;
        this.remainingTime = burstTime;
    }

    public int getId() {
        return id;
    }

    public int getBurstTime() {
        return burstTime;
    }

    public int getRemainingTime() {
        return remainingTime;
    }

    public void setRemainingTime(int remainingTime) {
        this.remainingTime = remainingTime;
    }

    public String toString() {
        return "Process " + id + " (Burst Time: " + burstTime + ", Remaining Time: " + remainingTime
+ ")";
    }
}

public class SRTN {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of processes: ");
        int n = scanner.nextInt();

        // Create a list of processes with their burst times
        ArrayList<Process> processes = new ArrayList<Process>();
        for (int i = 1; i <= n; i++) {
            System.out.print("Enter the burst time of process " + i + ": ");
            int burstTime = scanner.nextInt();
            processes.add(new Process(i, burstTime));
        }

        // Create a priority queue to store processes based on their remaining times
        PriorityQueue<Process> pq = new PriorityQueue<Process>(new Comparator<Process>() {
            public int compare(Process p1, Process p2) {
                return p1.getRemainingTime() - p2.getRemainingTime();
            }
        });

        int time = 0;
        while (!processes.isEmpty() || !pq.isEmpty()) {
            // Add processes to the priority queue whose arrival time is less than or equal to the
current time
            while (!processes.isEmpty() && processes.get(0).getBurstTime() <= time) {
                pq.offer(processes.remove(0));
            }

            // If the priority queue is not empty, pick the process with the shortest remaining time
            if (!pq.isEmpty()) {
                Process p = pq.poll();
                System.out.println("Time " + time + "-" + (time + 1) + ": " + p);
                p.setRemainingTime(p.getRemainingTime() - 1);
                if (p.getRemainingTime() > 0) {
                    pq.offer(p);
                }
            } else {
                System.out.println("Time " + time + "-" + (time + 1) + ": Idle");
            }

            time++;
        }

        // Close the scanner object
        scanner.close();
    }
}
```

Q4. import java.util.HashSet;

```
public class HashSetExample {
    public static void main(String[] args) {
        HashSet<String> hashSet = new HashSet<>();

        // Inserting elements into the HashSet
        hashSet.add("apple");
        hashSet.add("banana");
        hashSet.add("orange");
        hashSet.add("grape");

        // Displaying the contents of the HashSet
        System.out.println("Elements of HashSet: " + hashSet);
    }
}
```

Q5.
```
import java.util.LinkedHashSet;

public class LinkedHashSetExample {
    public static void main(String[] args) {
        // Create a LinkedHashSet of type double
        LinkedHashSet<Double> set = new LinkedHashSet<>();

        // Insert some elements into the set
        set.add(3.14);
        set.add(2.71);
        set.add(1.23);
        set.add(4.56);
        set.add(5.67);

        // Display the set
        System.out.println("Set elements: " + set);
    }
}
```

Q6.
```
import java.util.HashMap;
import java.util.Map;

public class HashMapExample {
    public static void main(String[] args) {
        // Create a HashMap
        Map<String, Integer> map = new HashMap<>();

        // Insert some elements into the map
        map.put("John", 25);
        map.put("Alice", 30);
        map.put("Bob", 20);
        map.put("Charlie", 35);
        map.put("David", 28);

        // Display the map
        System.out.println("Map elements: " + map);
    }
}
```

Q7.
```
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class NoDuplicates {
    public static void main(String[] args) {
        // Create a Set to hold the numbers
        Set<Integer> numbers = new HashSet<>();

        // Read N numbers from the user
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        System.out.println("Enter " + n + " numbers:");
        for (int i = 0; i < n; i++) {
            int num = scanner.nextInt();
            numbers.add(num); // add the number to the set
        }

        // Display the unique numbers
        System.out.println("Unique numbers: " + numbers);
    }
}
```