Swapnanil Dutta Roll: 12

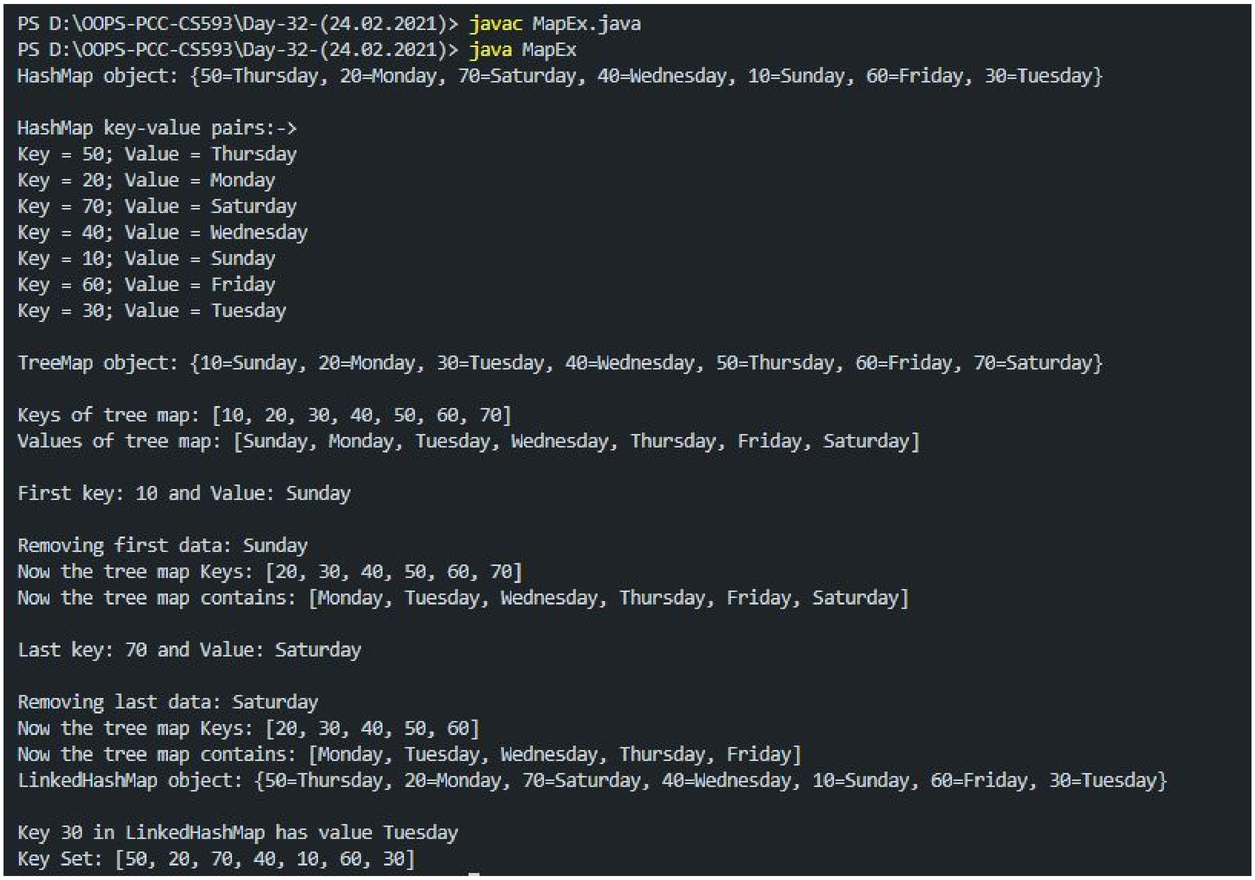
1.

Code:

```
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.Map;
import java.util.TreeMap;
import java.util.Set;

public class MapEx {
    public static void main(String[] args) {
        HashMap<Integer, String> hMap = new HashMap<>();
        hMap.put(60, "Friday");
        hMap.put(40, "Wednesday");
        hMap.put(30, "Tuesday");
        hMap.put(50, "Thursday");
        hMap.put(20, "Monday");
        hMap.put(70, "Saturday");
        hMap.put(10, "Sunday");
        System.out.println("HashMap object: " + hMap + "\n");
        System.out.println("HashMap key-value pairs:-> ");
        for (Map.Entry<Integer, String> entry : hMap.entrySet()) {
            Integer key = entry.getKey();
            String value = entry.getValue();
            System.out.println("Key = " + key + "; Value = " + value);
        }
        System.out.println();
        TreeMap<Integer, String> tMap = new TreeMap<>(hMap);
        System.out.println("TreeMap object: " + tMap + "\n");
        System.out.println("Keys of tree map: " + tMap.keySet());
        System.out.println("Values of tree map: " + tMap.values());
        System.out.println("\nFirst key: " + tMap.firstKey() + " and
Value: " + tMap.get(tMap.firstKey()) + "\n");
        System.out.println("Removing first data: " +
tMap.remove(tMap.firstKey()));
        System.out.println("Now the tree map Keys: " + tMap.keySet());
        System.out.println("Now the tree map contains: " + tMap.values() +
"\n");
        System.out.println("Last key: " + tMap.lastKey() + " and Value: "
+ tMap.get(tMap.lastKey()) + "\n");
```

```
        System.out.println("Removing last data: " +
tMap.remove(tMap.lastKey()));
        System.out.println("Now the tree map Keys: " + tMap.keySet());
        System.out.println("Now the tree map contains: " + tMap.values());
        LinkedHashMap<Integer, String> lhMap = new LinkedHashMap<>(hMap);
        System.out.println("LinkedHashMap object: " + lhMap + "\n");
        Integer key = 30;
        if (lhMap.containsKey(key)) {
            System.out.println("Key " + key + " in LinkedHashMap has value
" + lhMap.get(key));
        } else {
            System.out.println("Key " + key + " is not found!");
        }
        Set<Integer> set = lhMap.keySet();
        System.out.println("Key Set: " + set);
    }
}
```

Output:



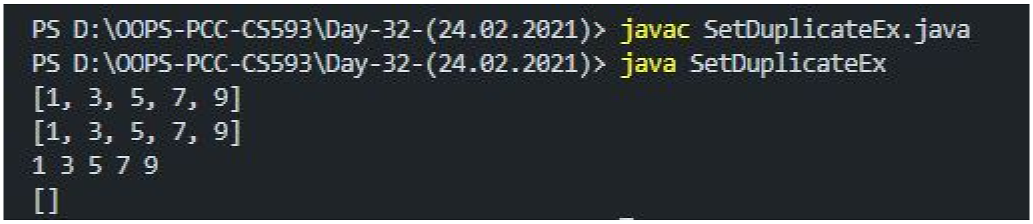
```
PS D:\OOPS-PCC-CS593\Day-32-(24.02.2021)> javac MapEx.java
PS D:\OOPS-PCC-CS593\Day-32-(24.02.2021)> java MapEx
HashMap object: {50=Thursday, 20=Monday, 70=Saturday, 40=Wednesday, 10=Sunday, 60=Friday, 30=Tuesday}

HashMap key-value pairs:->
Key = 50; Value = Thursday
Key = 20; Value = Monday
Key = 70; Value = Saturday
Key = 40; Value = Wednesday
Key = 10; Value = Sunday
Key = 60; Value = Friday
Key = 30; Value = Tuesday

TreeMap object: {10=Sunday, 20=Monday, 30=Tuesday, 40=Wednesday, 50=Thursday, 60=Friday, 70=Saturday}

Keys of tree map: [10, 20, 30, 40, 50, 60, 70]
Values of tree map: [Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday]

First key: 10 and Value: Sunday

Removing first data: Sunday
Now the tree map Keys: [20, 30, 40, 50, 60, 70]
Now the tree map contains: [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday]

Last key: 70 and Value: Saturday

Removing last data: Saturday
Now the tree map Keys: [20, 30, 40, 50, 60]
Now the tree map contains: [Monday, Tuesday, Wednesday, Thursday, Friday]
LinkedHashMap object: {50=Thursday, 20=Monday, 70=Saturday, 40=Wednesday, 10=Sunday, 60=Friday, 30=Tuesday}

Key 30 in LinkedHashMap has value Tuesday
Key Set: [50, 20, 70, 40, 10, 60, 30]
```

2.

Code:

```java
import java.util.Arrays;
import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;

public class SetDuplicateEx {
    public static void main(String[] args) {
        Integer list[] = { 7, 3, 5, 9, 1, 3, 9 };
        Set<Integer> set = new HashSet<>(Arrays.asList(list));
        System.out.println(set);
        System.out.println(set);
        Iterator iterator = set.iterator();
        while (iterator.hasNext()) {
            System.out.print(iterator.next() + " ");
            ;
        }
        System.out.println();
        set.clear();
        System.out.println(set);
    }
}
```

Output:

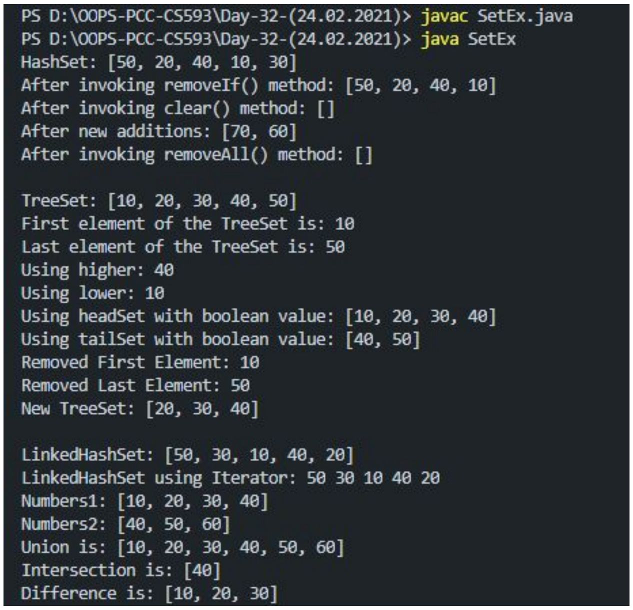


```
PS D:\OOPS-PCC-CS593\Day-32-(24.02.2021)> javac SetDuplicateEx.java
PS D:\OOPS-PCC-CS593\Day-32-(24.02.2021)> java SetDuplicateEx
[1, 3, 5, 7, 9]
[1, 3, 5, 7, 9]
1 3 5 7 9
[]
```

3.
Code:

```
import java.util.Arrays;
import java.util.HashSet;
import java.util.Iterator;
import java.util.LinkedHashSet;
import java.util.TreeSet;
public class SetEx {
    public static void main(String[] args) {
        HashSet<Integer> hSet = new HashSet<Integer>();
        TreeSet<Integer> tSet = new TreeSet<>();
        LinkedHashSet<Integer> lhSet = new LinkedHashSet<>();
        for (Integer val : Arrays.asList(50, 30, 10, 40, 20)) {
            hSet.add(val);
            tSet.add(val);
            lhSet.add(val);
        }
        System.out.println("HashSet: " + hSet);
        hSet.removeIf(val -> (val == 30));
        System.out.println("After invoking removeIf() method: " + hSet);
        hSet.clear();
        System.out.println("After invoking clear() method: " + hSet);
        hSet.add(60);
        hSet.add(70);
        System.out.println("After new additions: " + hSet);
        hSet.removeAll(hSet);
        System.out.println("After invoking removeAll() method: " + hSet);
        System.out.println("\nTreeSet: " + tSet);
        System.out.println("First element of the TreeSet is: " +
tSet.first());
        System.out.println("Last element of the TreeSet is: " +
tSet.last());
        System.out.println("Using higher: " + tSet.higher(30));
        System.out.println("Using lower: " + tSet.lower(20));
        System.out.println("Using headSet with boolean value: " +
tSet.headSet(40, true));
        System.out.println("Using tailSet with boolean value: " +
tSet.tailSet(30, false));
        System.out.println("Removed First Element: " + tSet.pollFirst());
        System.out.println("Removed Last Element: " + tSet.pollLast());
        System.out.println("New TreeSet: " + tSet);
        System.out.println("\nLinkedHashSet: " + lhSet);
        Iterator<Integer> iterate = lhSet.iterator();
        System.out.print("LinkedHashSet using Iterator: ");
        while (iterate.hasNext()) {
            System.out.print(iterate.next() + " ");
        }
        System.out.println();
        LinkedHashSet<Integer> result;
        LinkedHashSet<Integer> numbers1 = new
LinkedHashSet<>(Arrays.asList(10, 20, 30, 40));
        LinkedHashSet<Integer> numbers2 = new
LinkedHashSet<>(Arrays.asList(40, 50, 60));
        System.out.println("Numbers1: " + numbers1);
        System.out.println("Numbers2: " + numbers2);
        result = numbers1;
        result.addAll(numbers2);
        System.out.println("Union is: " + result);
        result = new LinkedHashSet<>(Arrays.asList(10, 20, 30, 40));
        result.retainAll(numbers2);
        System.out.println("Intersection is: " + result);
        result = new LinkedHashSet<>(Arrays.asList(10, 20, 30, 40));
        result.removeAll(numbers2);
        System.out.println("Difference is: " + result);
    }
}
```

Output:



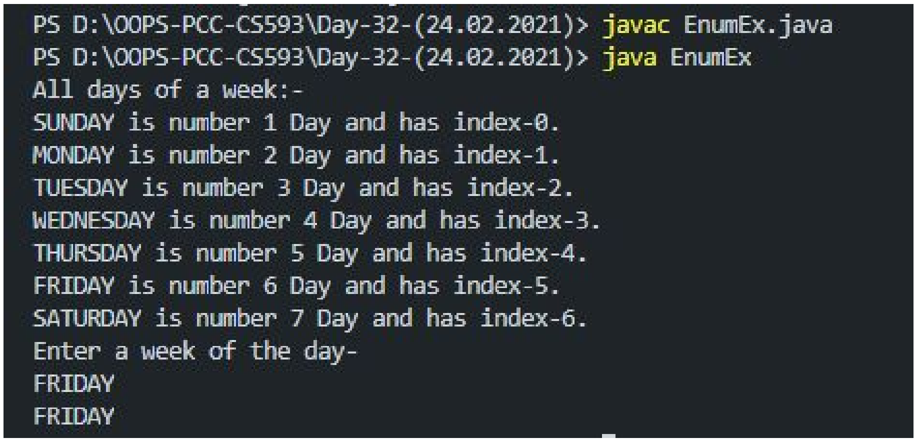
```
PS D:\OOPS-PCC-CS593\Day-32-(24.02.2021)> javac SetEx.java
PS D:\OOPS-PCC-CS593\Day-32-(24.02.2021)> java SetEx
HashSet: [50, 20, 40, 10, 30]
After invoking removeIf() method: [50, 20, 40, 10]
After invoking clear() method: []
After new additions: [70, 60]
After invoking removeAll() method: []

TreeSet: [10, 20, 30, 40, 50]
First element of the TreeSet is: 10
Last element of the TreeSet is: 50
Using higher: 40
Using lower: 10
Using headSet with boolean value: [10, 20, 30, 40]
Using tailSet with boolean value: [40, 50]
Removed First Element: 10
Removed Last Element: 50
New TreeSet: [20, 30, 40]

LinkedHashSet: [50, 30, 10, 40, 20]
LinkedHashSet using Iterator: 50 30 10 40 20
Numbers1: [10, 20, 30, 40]
Numbers2: [40, 50, 60]
Union is: [10, 20, 30, 40, 50, 60]
Intersection is: [40]
Difference is: [10, 20, 30]
```

4.

Code:

```
import java.io.*;
public class EnumEx {
    public static void main(String[] args) throws IOException {
        System.out.println("All days of a week:- ");
        for (Week w : Week.values()) {
            System.out.println(w + " is number " + w.getVal() + " Day and
has index-" + w.ordinal() + ".");
        }
        System.out.println("Enter a week of the day-");
        BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
        String str = in.readLine();
        Week w1 = Week.valueOf(str);
        System.out.print(w1);
    }
}
enum Week {
    SUNDAY(1), MONDAY(2), TUESDAY(3), WEDNESDAY(4), THURSDAY(5),
FRIDAY(6), SATURDAY(7);
    private int val;
    Week(int n) {
        this.val = n;
    }
    public int getVal() {
        return val;
    }
}
```

Output:



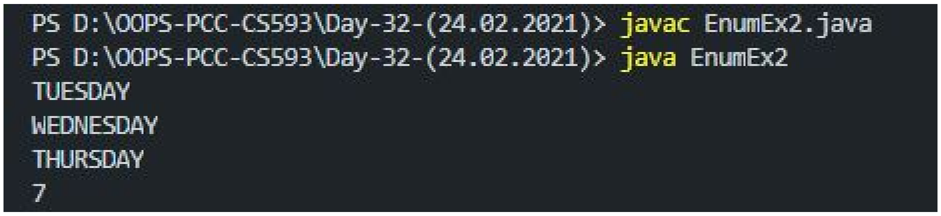
```
PS D:\OOPS-PCC-CS593\Day-32-(24.02.2021)> javac EnumEx.java
PS D:\OOPS-PCC-CS593\Day-32-(24.02.2021)> java EnumEx
All days of a week:-
SUNDAY is number 1 Day and has index-0.
MONDAY is number 2 Day and has index-1.
TUESDAY is number 3 Day and has index-2.
WEDNESDAY is number 4 Day and has index-3.
THURSDAY is number 5 Day and has index-4.
FRIDAY is number 6 Day and has index-5.
SATURDAY is number 7 Day and has index-6.
Enter a week of the day-
FRIDAY
FRIDAY
```

5.

Code:

```
import java.util.EnumMap;
import java.util.EnumSet;
import java.util.Iterator;

public class EnumEx2 {
    public static void main(String[] args) {
        EnumSet<Week> enumSet = EnumSet.of(Week.THURSDAY, Week.TUESDAY,
Week.WEDNESDAY);
        Iterator<Week> it = enumSet.iterator();
        while (it.hasNext()) {
            System.out.println(it.next());
        }
        EnumMap<Week, String> enumMap = new EnumMap<Week,
String>(Week.class);
        enumMap.put(Week.SUNDAY, "1");
        enumMap.put(Week.TUESDAY, "5");
        enumMap.put(Week.THURSDAY, "6");
        enumMap.put(Week.WEDNESDAY, "9");
        enumMap.put(Week.FRIDAY, "7");
        String day = enumMap.get(Week.FRIDAY);
        System.out.println(day);
    }
}
```

Output:



```
PS D:\OOPS-PCC-CS593\Day-32-(24.02.2021)> javac EnumEx2.java
PS D:\OOPS-PCC-CS593\Day-32-(24.02.2021)> java EnumEx2
TUESDAY
WEDNESDAY
THURSDAY
7
```