2) Find the second most repeating number in an array without using extra storage.

E.g.. [6,8,1,2,6,3,6,2,3,2,6]

**Program:-**

**(KeyValueModelClass.java)**

**package** org.pojos;

**public** **class** KeyValueModelClass {

**int** key,value;

**public** **int** getKey() {

**return** key;

}

**public** **void** setKey(**int** key) {

**this**.key = key;

}

**public** **int** getValue() {

**return** value;

}

**public** **void** setValue(**int** value) {

**this**.value = value;

}

}

**(CountComparator.java)**

**package** org.comparators;

**import** java.util.Comparator;

**import** org.pojos.KeyValueModelClass;

**public** **class** CountComparator **implements** Comparator<KeyValueModelClass>{

@Override

**public** **int** compare(KeyValueModelClass k1, KeyValueModelClass k2) {

**if**(k1.getValue()<k2.getValue())

**return** 1;

**else** **if**(k1.getValue()>k2.getValue())

**return** -1;

**else**

**return** 0;

}

}

**(SecondHighestRepeatedNumber.java)**

**package** org.tasks;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.HashMap;

**import** java.util.HashSet;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Map.Entry;

**import** java.util.Scanner;

**import** java.util.Set;

**import** org.comparators.CountComparator;

**import** org.pojos.KeyValueModelClass;

**public** **class** SecondHighestRepeatedNumber {

**public** **static** **void** main(String[] args) {

Scanner in = **new** Scanner(System.***in***);

System.***out***.println("Enter the no. of integers");

**int** number = in.nextInt();

System.***out***.println("Enter the integers");

**int** a[] = **new** **int**[number];

**for**(**int** i=0;i<number;i++)

a[i]=in.nextInt();

Set<Integer> si = **new** HashSet<Integer>();

**for**(**int** i=0;i<a.length;i++)

si.add(a[i]);

HashMap<Integer, Integer> hsi = **new** HashMap<Integer, Integer>();

Iterator<Integer> ii =si.iterator();

**while**(ii.hasNext())

{

**int** r =ii.next();

**int** count=0;

**for**(**int** i=0;i<a.length;i++)

{

**if**(r==a[i])

{

count++;

}

}

hsi.put(r, count);

}

Set<Entry<Integer, Integer>> sei =hsi.entrySet();

Iterator<Entry<Integer, Integer>> ie =sei.iterator();

List<KeyValueModelClass> list = **new** ArrayList<KeyValueModelClass>();

**while**(ie.hasNext())

{

Entry<Integer, Integer> eii =ie.next();

KeyValueModelClass keyValueModelClass = **new** KeyValueModelClass();

keyValueModelClass.setKey(eii.getKey());

keyValueModelClass.setValue(eii.getValue());

list.add(keyValueModelClass);

}

Collections.*sort*(list, **new** CountComparator());

Iterator<KeyValueModelClass> ikv= list.iterator();

**int** c=0;

**while**(ikv.hasNext())

{

KeyValueModelClass kvmc= ikv.next();

c++;

**if**(c==2)

System.***out***.println("Second Most Repeating number is : "+kvmc.getKey()+" \nNo. of times repeated is: "+kvmc.getValue());

}

}

}