**Hacker Rank Challenges**

**Lonely Integer:**

**import java.io.\*;**

**import java.util.\*;**

**import java.text.\*;**

**import java.math.\*;**

**import java.util.regex.\*;**

**public class Solution {**

**static int lonelyinteger(int[] a) {**

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

**int finalValue = 0;**

**for(int value : a){**

**if(finalMap.containsKey(value)) {**

**finalMap.put(value, finalMap.get(value)+1);**

**} else {**

**finalMap.put(value, 1);**

**}**

**}**

**for(Map.Entry<Integer,Integer> entry: finalMap.entrySet()){**

**if(entry.getValue() == 1){**

**finalValue = entry.getKey();**

**}**

**}**

**return finalValue;**

**}**

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

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

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

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

**for(int a\_i = 0; a\_i < n; a\_i++){**

**a[a\_i] = in.nextInt();**

**}**

**int result = lonelyinteger(a);**

**System.out.println(result);**

**}**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Recursive Digit Sum:**

**import java.io.\*;**

**import java.util.\*;**

**import java.text.\*;**

**import java.math.\*;**

**import java.util.regex.\*;**

**public class Solution {**

**static int superDigit(String n, int k) {**

**String finalString = "";**

**for (int i =0; i < k; i++){**

**finalString = finalString+n;**

**}**

**int sum =0;**

**int finalSum = 0;**

**do{**

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

**sum = sum + (finalString.charAt(i) - 48);**

**finalSum = sum;**

**}**

**finalString = String.valueOf(finalSum);**

**sum =0;**

**} while(finalString.length() > 1);**

**return finalSum;**

**}**

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

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

**String n = in.next();**

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

**int result = superDigit(n, k);**

**System.out.println(result);**

**in.close();**

**}**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**SUM vs XOR:**

**import java.io.\*;**

**import java.util.\*;**

**import java.text.\*;**

**import java.math.\*;**

**import java.util.regex.\*;**

**public class Solution {**

**static long solve(long n) {**

**int count = 0;**

**while (n > 0) {**

**if ((n & 1) == 0) {**

**count++;**

**}**

**n >>= 1; // divides by 2**

**}**

**return 1L << count ;**

**}**

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

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

**long n = in.nextLong();**

**long result = solve(n);**

**System.out.println(result);**

**}**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Flipping Bits:**

**import java.io.\*;**

**import java.util.\*;**

**import java.text.\*;**

**import java.math.\*;**

**import java.util.regex.\*;**

**public class Solution {**

**static int t;**

**static long[] results;**

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

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

**t = in.nextInt();**

**results = new long[t];**

**for (int i = 0; i < t; i++) {**

**long n = in.nextLong();**

**results[i] = flipBits(n);**

**}**

**for (int i = 0; i < t; i++) {**

**System.out.println(results[i]);**

**}**

**}**

**private static long flipBits(long n) {**

**long mask = 0xffffffffl;**

**long result = n ^ mask;**

**return result;**

**}**

**}**