[A - Repeated Numbers](https://vjudge.net/problem/HackerRank-si-repeated-numbers" \t "_blank)

 You are given an array of N elements. All elements of the array are in range 1 to N-2. All elements occur once except two numbers, which occur twice. Your task is to find the two repeating numbers.

**Input Format**

First line of input contains T - number of test cases. Its followed by 2T lines, the first line contains N - the size of the array and second line contains the elements of the array.

**Constraints**

30 points  
1 <= T <= 100  
4 <= N <= 103

70 points  
1 <= T <= 100  
4 <= N <= 106

**Output Format**

Print the 2 repeated numbers in sorted manner, for each test case, separated by new line.

**Sample Input 0**

2

8

1 3 2 3 4 6 5 5

10

1 5 2 8 1 4 7 4 3 6

**Sample Output 0**

3 5

1 4

**Explanation 0**

Self Explanatory

### **Repeated Numbers**

### **Bit Manipulation Approach**

What if we do XOR of all array elements with numbers from [1, N-2] and assign it to X.

Let's say N = 7, A = {1, 2, 2, 3, 4, 5, 5}

X = [1 ^ **2 ^ 2** ^ 3 ^ 4 ^ **5 ^ 5**] ^ [1 ^ **2** ^ 3 ^ 4 ^ **5**] =   This will cause all elements to nullify except two repeating elements.  X = 2 ^ 5 = 7.

Now somehow we need to separate these two values and print them.

OBSERVATION TO USE - In Binary Form of X, every bit that is 1 is guaranteed to be different in two repeated numbers. Let's assume two numbers are A and B and X = A ^ B and we want to separate out A and B.

Let's assume we can create a mask from binary pattern of X that has all bits 0 except one bit that is set (Any bit in fact).    We already know it can be done easily using MASK = X ^ -X.

Since X = 7, our mask will be

X = 7 ^ 6 = 1

Now using this mask, array elements and numbers in range [1 to N-2] it is easy to separate out two numbers.  Here is how.

// skipping calculation of x...  
  
a = 0  
b = 0  
  
mask = x & -x   
  
for i = [0, n-1]  
    if( (arr[i] & mask) == 0)  
        a = a ^ arr[i]  
    else  
        b = b ^ arr[i]  
  
  
for i = [1, n-2]  
    if( (i & mask) == 0)  
        a = a ^ i  
    else  
        b = b ^ i  
  
// now a and b contain answer

### **Map Based Solution**

We can solve this problem using map/dictionary,  Take a map and store each element and its corresponding frequency.

Now elements whose frequency is 2 are our answers.

### **Set Based Solution**

We can take a set of integers.  Loop over array elements and do the following for each element.

set st;  
  
for i = [0, n-1]  
    if st contains a[i]  
        remove a[i] from st  
    else  
        add a[i] to st

Now set has all elements except two repeated ones.  Since we know elements are in range 1 to N-2 only so we can just check which two elements are missing from set.

a = 0  
b = 0  
for i = [1, n-2]  
    if st does not contain i  
        if a == 0   
            a = i     // first missing element   
        else  
            b = i     // second missing element

So finally a and b are answers.