**The Monk and Prateek**

Attempted by: **321**

/

Accuracy: **80%**

/

Maximum Score: **30**

/

3 Votes

Tag(s):

Easy-Medium, Easy-Medium, Hashing, Implementation

The Monk is extremely fond of Prateek, the new HackerEarth moderator. This time, Prateek had to deal with a crazy hash function, f(n)f(n), called the **r3gz3n function**, which is as follows: NN [xor](https://en.wikipedia.org/wiki/Bitwise_operation" \l "XOR) (sumofdigitsofN)(sumofdigitsofN). For example, f(81)f(81) = 8181 ^ (88+11) = 8888.

Now to test the efficiency of his friend, The Monk gives Prateek a list of **N** numbers, and asks him to find out the following information about this list:

* The value of the **r3gz3n function** which occurs the maximum number of times.
* Number of collisions with the hash function.

**Note-1:** If all the values of the function are unique, print the **maximum** value which occurs in the list of the hashed values.  
**Note-2:** If there are multiple hashed values which occur the maximum number of times, print the **smallest hashed value with the maximum count.**

**Input format:**  
The first line will contain a single digit integer, **N**, denoting the number of numbers in a list. The second line will contain **N** integers, each separated by a space.

**Output format:**  
Print two integers separated by a space, where the first integer denotes the value of the function which occurs the maximum number of times. Remember that if all the values of the function are unique, print the **maximum** value which occurs in the list. The second integer would denote the number of collisions.

**Constraints:**  
11 ≤ N ≤ 105105   
11 ≤ Ni ≤ 107107

**Sample Input-0:**  
44  
1010 1111 1212 1313

**Sample Output-0:**  
99 11

**Sample Explanation-0:**  
1010^11 = 1111.  
1111^22 = 99.  
1212^33 = 1515.  
1313^44 = 99.

The value which occurs the maximum number of times is: 99, and the number of collisions is: 11.

**Sample Input-1:**  
44  
11 11 2121 2121

**Sample Output-1:**  
00 22

**Sample Explanation-1:**  
11^11 = 00.  
11^11 = 00.  
2121^33 = 2222.  
2121^33 = 2222.

The value which occurs the maximum number of times: 00, and 2222; but since 00 is the smaller one, that is our answer.The number of collisions are: 22.

<https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/the-monk-and-prateek/#c84461>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

long n = long.Parse(Console.ReadLine());

long[] arr = Array.ConvertAll(Console.ReadLine().Split(' '), e => long.Parse(e));

//string n = "12";

//int sum = n.Sum(e => e - '0');

//Console.WriteLine(sum);

//int[] arr = { 10, 11, 12, 13 };

//int[] arr = { 1, 1, 21, 21 };

// int[] arr = { 40, 50 };

//10

//long[] arr = Array.ConvertAll("49 50 25 48 22 3 4 8 14 10".Split(' '), e => long.Parse(e));

//0 4

Dictionary<long,long> diccio = new Dictionary<long,long>();

for (int i = 0; i < arr.Length; i++)

{

long sum\_dig = arr[i].ToString().Sum(e => e - '0');

long xor = arr[i] ^ sum\_dig;

if (diccio.ContainsKey(xor))

{

diccio[xor]++;

}

else

{

diccio[xor] = 1;

}

}

long max\_value = diccio.Values.Max();

long number\_of\_collitions = 0;

List<long> max\_keys = new List<long>();

foreach (KeyValuePair<long,long> kvp in diccio)

{

if (kvp.Value == max\_value)

{

max\_keys.Add(kvp.Key);

}

if (kvp.Value > 1)

{

number\_of\_collitions += kvp.Value - 1;

}

}

max\_keys = max\_keys.Distinct().ToList();

if (max\_value == 1)

{

Console.WriteLine(max\_keys.Max() + " " + number\_of\_collitions);

}

else

{

Console.WriteLine(max\_keys.Min() + " " + number\_of\_collitions);

}

//foreach (int elem in max\_keys)

//{

// Console.Write(elem + " ");

//}

Console.ReadLine();

}

}

}