

Xor subsequence

You are given **N** integers: $A_1, A_2, A_3, \dots, A_N$. We pick up a consecutive subsequence of integers from the given series. $A[i], A[i+1], \dots, A[j-1], A[j]$, ($1 \leq i \leq j \leq N$). For example, if **N = 3**:
The subsequences we may consider are

```
A[1]
A[2]
A[3]
A[1],A[2]
A[2],A[3]
A[1],A[2],A[3]
```

For each subsequence, we apply the bitwise **XOR** operation to all the integers and record the value of this **XOR** operation. Since there are **$N \times (N+1)/2$** subsequences, we obtain **$N \times (N + 1) / 2$** numbers.
Your task is to find the most frequent number in the **recorded list** and how many times it appears.

Input Format

The first line contains an integer **N** ($1 \leq N \leq 100000$). This is followed by **N** lines, each containing one integer A_i per line. ($1 \leq A_i < 2^{16}$).

Output Format

Output one line contains the most frequent number and how many times it appears. If there is multiple number that has the most frequency, choose the minimum number.

Sample Input

```
4
2
1
1
3
```

Sample output

```
1 3
```

Explanation

Finding the XOR in all the consecutive subsequences:

```
2 = 2
2 ^ 1 = 3
2 ^ 1 ^ 1 = 2
2 ^ 1 ^ 1 ^ 3 = 1
1 = 1
1 ^ 1 = 0
1 ^ 1 ^ 3 = 3
1 = 1
1 ^ 3 = 2
3 = 3
```

1, 2, 3 are all repeated three times. Since we are looking for the minimum number, 1 is the answer.

