1 Gaussian elimination

Explained here: https://cp-algorithms.com/linear_algebra/linear-system-gauss.html

I have to notice that you can use if you have XOR operation instead of addition because all properties of addition are kept in XOR too.

The pseudocode will be shared as a notification in the contest.

2 XOR

XOR (the notation is $\oplus$ but in C++ and Python you should use sign) is an operation defined for bits:

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

Basically, it is equal to 1 if bits are different. You can also define it for numbers by applying it bit by bit:

$$25 \oplus 13 = 11001_2 \oplus 01101_2 = 10100_2 = 20$$

Some nice properties of XOR:

1. $a \oplus b = b \oplus a$

2. $a \oplus 0 = a$

3. $a \oplus a = 0$

4. If $a = c$ then $c \oplus b = a$ and $c \oplus a = b$

XOR is also called Exclusive OR or addition modulo 2. You can actually think about it as about addition but we don't carry bits (i.e. if we have at some position two 1 we just set $(1+1)mod 2 = 0$ and ignore the fact that there now is another 1 we should add to the next bit.

2.1 Problem example 1

You're given array of n integers. You're guaranteed that one element appears once and all other appear exactly twice. Your task is to find the element that appears ones.

The answer: we can XOR all the elements. The result will be the answer.

2.2 Problem example 2

We have an array of n integers. Also we have q queries of format l, r which mean that we're asked to find $a_l \oplus a_{l+1} \oplus \ldots \oplus a_r$.

The answer: We can use prefix XORs (same way as prefix sums) $p_i = a_1 \oplus a_2 \oplus \ldots \oplus a_i$. Now the answer to the query is simply $p_r \oplus p_{l-1}$.

3 XOR basis

You can read about it here (less formal): https://codeforces.com/blog/entry/100066

And here (more formal but still good): https://codeforces.com/blog/entry/68953

The pseudocode will be shared as a notification in the contest.