Problem L. Array Elimination

Time limit 2000 ms
Mem limit 524288 kB

You are given array $a_1, a_2, \ldots, a_n$, consisting of non-negative integers.

Let's define operation of "elimination" with integer parameter k ($1 \le k \le n$) as follows:

- Choose k distinct array indices $1 \le i_1 < i_2 < \ldots < i_k \le n$.
- Calculate $x = a_{i_1} \ \& \ a_{i_2} \ \& \ \ldots \ \& \ a_{i_k}$, where $\&$ denotes the bitwise AND operation (notes section contains formal definition).
- Subtract x from each of $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$; all other elements remain untouched.

Find all possible values of k, such that it's possible to make all elements of array a equal to 0 using a finite number of elimination operations with parameter k. It can be proven that exists at least one possible k for any array a.

Note that you **firstly choose k and only after that perform elimination operations with value k you've chosen initially**.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). Description of the test cases follows.

The first line of each test case contains one integer n ($1 \le n \le 200\,000$) — the length of array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{30}$) — array a itself.

It's guaranteed that the sum of n over all test cases doesn't exceed $200\,000$.

Output

For each test case, print all values k, such that it's possible to make all elements of a equal to 0 in a finite number of elimination operations with the given parameter k.

Print them in increasing order.

Sample 1

Input	Output
5 4 4 4 4 4 4 13 7 25 19 6 3 5 3 1 7 1 1 1 5 0 0 0 0 0	1 2 4 1 2 1 1 1 2 3 4 5

Note

In the first test case:

- If $k = 1$, we can make four elimination operations with sets of indices $\{1\}, \{2\}, \{3\}, \{4\}$. Since $\&$ of one element is equal to the element itself, then for each operation $x = a_i$, so $a_i - x = a_i - a_i = 0$.
- If $k = 2$, we can make two elimination operations with, for example, sets of indices $\{1, 3\}$ and $\{2, 4\}$: $x = a_1 \ \& \ a_3 = a_2 \ \& \ a_4 = 4 \ \& \ 4 = 4$. For both operations $x = 4$, so after the first operation $a_1 - x = 0$ and $a_3 - x = 0$, and after the second operation — $a_2 - x = 0$ and $a_4 - x = 0$.
- If $k = 3$, it's impossible to make all a_i equal to 0. After performing the first operation, we'll get three elements equal to 0 and one equal to 4. After that, all elimination operations won't change anything, since at least one chosen element will always be equal to 0.
- If $k = 4$, we can make one operation with set $\{1, 2, 3, 4\}$, because $x = a_1 \ \& \ a_2 \ \& \ a_3 \ \& \ a_4 = 4$.

In the second test case, if $k = 2$ then we can make the following elimination operations:

- Operation with indices $\{1, 3\}$: $x = a_1 \ \& \ a_3 = 13 \ \& \ 25 = 9$. $a_1 - x = 13 - 9 = 4$ and $a_3 - x = 25 - 9 = 16$. Array a will become equal to $[4, 7, 16, 19]$.
- Operation with indices $\{3, 4\}$: $x = a_3 \ \& \ a_4 = 16 \ \& \ 19 = 16$. $a_3 - x = 16 - 16 = 0$ and $a_4 - x = 19 - 16 = 3$. Array a will become equal to $[4, 7, 0, 3]$.
- Operation with indices $\{2, 4\}$: $x = a_2 \ \& \ a_4 = 7 \ \& \ 3 = 3$. $a_2 - x = 7 - 3 = 4$ and $a_4 - x = 3 - 3 = 0$. Array a will become equal to $[4, 4, 0, 0]$.
- Operation with indices $\{1, 2\}$: $x = a_1 \ \& \ a_2 = 4 \ \& \ 4 = 4$. $a_1 - x = 4 - 4 = 0$ and $a_2 - x = 4 - 4 = 0$. Array a will become equal to $[0, 0, 0, 0]$.

Formal definition of bitwise AND:

Let's define bitwise AND ($\&$) as follows. Suppose we have two non-negative integers x and y, let's look at their binary representations (possibly, with leading zeroes): $x_k \dots x_2 x_1 x_0$ and

$y_k \dots y_2 y_1 y_0$. Here, x_i is the i-th bit of number x, and y_i is the i-th bit of number y. Let $r = x \ \& \ y$ is a result of operation $\&$ on number x and y. Then binary representation of r will be $r_k \dots r_2 r_1 r_0$, where:

$$r_i = \begin{cases} 1, \text{ if } x_i = 1 \text{ and } y_i = 1 \\ 0, \text{ if } x_i = 0 \text{ or } y_i = 0 \end{cases}$$