

XOR Multiset

You are given an integer n and $n - 1$ non-negative integers a_1, a_2, \dots, a_{n-1} .

Find a multiset S of integers from $\{1, 2, \dots, n - 1\}$ such that:

- $\sum_{x \in S} x \equiv 0 \pmod{n}$
- $\bigoplus_{x \in S} a_x$ is maximized, where \bigoplus denotes the bitwise XOR operation. The bitwise XOR operator works on the binary representation of two numbers and performs the logical exclusive OR operation on each pair of corresponding bits; so 5 (binary representation 0101) XOR 3 (binary representation 0011) gives 6 (binary representation 0110). The operator is `^` in C++, Java, and Python.

If there are multiple such multisets, you can return any of them.

Implementation Details

You need to implement the following function:

```
(int64, int32[]) find_multiset(int32 n, int64[] a)
```

- n : the modulus value
- a : array of length $n - 1$, where $a[i]$ corresponds to a_{i+1}
- The function should return a pair with:
 - **First element:** an integer representing the optimal value of $\bigoplus_{x \in S} a_x$ among all valid multisets S
 - **Second element:** a vector representing any optimal multiset S . The elements of the vector should be integers from 1 to $n - 1$, and the size of S has to be at most $2n$.

Constraints

- $1 \leq n \leq 10^5$
- $0 \leq a_i < 2^{62}$ for each $i = 1, 2, \dots, n - 1$

Scoring

- **Subtask 1** (20 points): $n \leq 10$
- **Subtask 2** (40 points): n is odd
- **Subtask 3** (40 points): No additional constraints

In each subtask, you can obtain a partial score if your program determines the optimal value of $\bigoplus_{x \in S} a_x$ among all valid multisets S . More precisely, you get the whole score of a subtask if in all of its test cases, the first element of the pair returned by `find_multiset` is exactly the same as the first element of the pair returned by the official grader and the second element is a valid multiset (i.e. it satisfies the conditions above) that achieves this optimal value. You get 60% of the score of a subtask if in all of its test cases, the first element of the pair returned by `find_multiset` is exactly the same as the first element of the pair returned by the official grader (regardless of the second element) and you get 0% of the score of a subtask otherwise.

Examples

The following `find_multiset(3, {5, 10})` call should return `{15, {1, 2}}`

- We have $n = 3$ and $a = [5, 10]$ (corresponding to $a_1 = 5, a_2 = 10$).
- We need to find a multiset $S \subseteq \{1, 2\}$ such that $\sum_{x \in S} x \equiv 0 \pmod{3}$.
- Valid multisets include: \emptyset (sum = 0), $\{1, 2\}$ (sum = 3 $\equiv 0$), $\{1, 1, 1\}$ (sum = 3 $\equiv 0$), $\{2, 2, 2\}$ (sum = 6 $\equiv 0$), etc.
- For $S = \{1, 2\}$: XOR is $a_1 \oplus a_2 = 5 \oplus 10 = 15$.
- For $S = \{1, 1, 1\}$: XOR is $a_1 \oplus a_1 \oplus a_1 = 5 \oplus 5 \oplus 5 = 5$.
- The maximum XOR value is 15, achieved by $S = \{1, 2\}$.

The following `find_multiset(4, {8, 12, 6})` call should return `{14, {1, 3}}`

- We have $n = 4$ and $a = [8, 12, 6]$ (corresponding to $a_1 = 8, a_2 = 12, a_3 = 6$).
- We need to find a multiset $S \subseteq \{1, 2, 3\}$ such that $\sum_{x \in S} x \equiv 0 \pmod{4}$.
- For $S = \{1, 3\}$: XOR is $a_1 \oplus a_3 = 8 \oplus 6 = 14$.
- For $S = \{2, 2\}$: XOR is $a_2 \oplus a_2 = 12 \oplus 12 = 0$.
- The maximum XOR value is 14, achieved by $S = \{1, 3\}$.

Sample Grader

The sample grader reads the input in the following format:

- Line 1: One integer n
- Line 2: $n - 1$ integers a_1, a_2, \dots, a_{n-1}

The sample grader calls `find_multiset(n, a)` and prints the returned multiset in the following format:

- First line: the value returned as the first element of the pair
- Second line: the size of the multiset
- Third line: the elements of the multiset (if any), separated by spaces

Note: The sample grader provided with this problem is just for testing your solution locally. The actual grader used during the contest may be different.