Transmitter

According to the scoreboard, this problem was the easiest out of all 6.

Subtask 1 $n \leq 10$ — 13 points

How can we check that a position i is broken? Let's send a sequence where all bits are 0 except the i-th one. If the result does not contain that 1 anymore, this position is definitely broken.

Subtask 2 There is exactly one broken position — 21 points

Send a sequence $[0, \ldots, 0, 1, \ldots, 1]$ where the left half is filled with zeroes and the right half is filled with ones. Then check the result. Exactly one number is missing. Is it a zero or is it a one? Now you know which half contains the broken position. Go there and split it in the middle again. Repeat the same procedure.

This resembles binary search. After $\log n = 10$ operations, you will find the broken position.

Subtask 3 There are no adjacent broken positions — 20 points

You can find the answer with only 2 queries. Send $[0, 0, 1, 1, 0, 0, 1, 1, \ldots]$ and $[0, 1, 0, 1, 0, 1, 0, 1, \ldots]$. Can you figure out all broken positions now?

The first query will tell you if each "block" of size 2 contains 0 or 1 broken positions. From the second query you can deduce which exact position is broken in each block.

Subtask 4 No additional constraints — 46 points

We will do a "binary search" on all broken bits:

1. Segments: Start with one big segment $[0, n-1]$, meaning we haven't yet separated which parts might be broken.
2. Transmit Pattern: In each round, for every segment $[l, r]$:
   - Calculate $\text{mid} = \lfloor \frac{l+r}{2} \rfloor$.
   - Assign 0 to the left half $[l, \ldots, \text{mid}]$ and 1 to the right half $[\text{mid} + 1, \ldots, r]$.
3. Count Received: We send that entire pattern of size n. In the response, bits from broken positions are missing — so we track how many bits from each segment actually made it. That tells us how many broken bits were in the left vs. right half.
4. Split and Repeat: Replace $[l, r]$ with two segments $[l, \text{mid}]$ and $[\text{mid} + 1, r]$ carrying the correct broken-count each.

After $\log n = 10$ rounds, each segment shrinks to a single position. If that position's broken count is 1, it's broken. We collect all broken positions and return them sorted.