

There are a few valid variations to solving this problem. I will cover 2 of them here. (There's probably something faster than both of these, so please do shoot me a message if you find something).

METHOD 1:

We can keep a dictionary of how many times each number has been submitted. For every possible submission, i where $1 \leq i \leq K$, $d[i]$ denotes the number of times a value has been submitted.

After building $d[i]$ we can simply loop i from 1 to K (inclusive) and if $d[i] = 1$ then we have a winner (it is also guaranteed that the first valid winner will be the smallest one since we are looping in ascending order). If no winner is found then we print -1 and exit.

METHOD 2:

We keep two sets of values. One containing every value we have already encountered and one containing all possible winning candidates (i.e. values we have only seen once so far).

Loop through every submission. For each submission:

- If it is in the set of seen values, then it is no longer a candidate for winning, and thus should be removed from the set of possible winners (if it has already been removed then you can skip the second operation)
- If it is not in the set of seen values, then add it to the set of potential winners and the set of seen values as well

After one loop through, simply attain the smallest value from the set of possible winners.