Problem D : Lost Frequencies

Deep in the middle of a vast desert, an isolated research station runs a long–term experiment on a mysterious species of beetles that communicate using ultrasonic frequencies. Each beetle is assigned a unique **non-negative frequency ID**, allowing the monitoring system to track individuals in the wild.

For years, the central computer at the station has been quietly recording these frequencies. Every time a beetle is detected, its frequency ID is stored in a log. But one night, a violent sandstorm hits the observatory. Power flickers, antennas bend under the wind, and the main recording unit is forced to reboot.

When the system finally comes back online, the researchers realize that the frequency log has been corrupted:

- Some recorded values have turned into **negative numbers**, which cannot possibly be valid IDs.
- Some entries have become **enormous integers**, with up to 100 digits, likely the result of overflow or bit flips.
- Only values that form valid **non-negative integers** can still be trusted as real frequency IDs.

Despite the damage, the experiment must continue. Whenever a **new** beetle is detected, the system must assign it the **smallest non-negative frequency ID** that is **not currently used** by any known beetle. This ensures that IDs remain compact and easy to manage.

The old allocator module, which used to handle this task, was destroyed in the reboot. The researchers now need you to re-implement this logic using the corrupted log they managed to recover.

Your job is to read the list of recorded values, filter out the invalid ones, and determine the smallest non-negative integer that does **not** appear among the valid frequency IDs. This is the ID that will be assigned to the next newly detected beetle.

Input

The input consists of the following lines:

- The first line contains an integer N, the number of recorded values.
- Each of the next N lines contains an integer X_i, written in decimal form. Each X_i may be:
  - a negative integer,
  - zero,
  - a positive integer,
  - a very large positive integer (up to 100 digits).

An entry X_i is considered a **valid frequency** if and only if it represents a non-negative integer. All other values (negative numbers or any value outside the allowed range) must

be ignored when determining which frequencies are already used.

Output

Print a single integer — the **smallest non-negative integer** that does not appear among the valid recorded frequencies.

Constraints

$$0 \leq N \leq 10^6$$

Each integer X_i is given as a decimal string of fewer than 100 digits. The input is well-formed: each line represents an integer in standard decimal notation, possibly with a leading minus sign.

Example

Standard Input	Standard Output
5 1 -1 3 0 10	2

In this example, the valid non-negative frequencies are $\{0, 1, 3, 10\}$. The smallest non-negative integer that does not appear in this set is 2.