B. Gena's Code

time limit per test: 0.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

It's the year 4527 and the tanks game that we all know and love still exists. There also exists Great Gena's code, written in 2016. The problem this code solves is: given the number of tanks that go into the battle from each country, find their product. If it is turns to be too large, then the servers might have not enough time to assign tanks into teams and the whole game will collapse!

There are exactly n distinct countries in the world and the i-th country added a_i tanks to the game. As the developers of the game are perfectionists, the number of tanks from each country is beautiful. A *beautiful* number, according to the developers, is such number that its decimal representation consists only of digits '1' and '0', moreover it contains **at most one** digit '1'. However, due to complaints from players, some number of tanks of **one** country was removed from the game, hence the number of tanks of this country may not remain beautiful.

Your task is to write the program that solves exactly the same problem in order to verify Gena's code correctness. Just in case.

Input

The first line of the input contains the number of countries n ($1 \le n \le 100\,000$). The second line contains n non-negative integers a_i without leading zeroes — the number of tanks of the i-th country.

It is guaranteed that the second line contains at least n - 1 beautiful numbers and the total length of all these number's representations doesn't exceed $100\,000$.

Output

Print a single number without leading zeroes — the product of the number of tanks presented by each country.

Examples

input
3 5 10 1

output
50

input
4 1 1 10 11

output
110

input
5 0 3 1 100 1

output
0

Note

In sample 1 numbers 10 and 1 are *beautiful*, number 5 is not not.

In sample 2 number 11 is not *beautiful* (contains two '`1`'s), all others are *beautiful*.

In sample 3 number 3 is not *beautiful*, all others are *beautiful*.