A. Minimum Binary Number

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

String can be called *correct* if it consists of characters "0" and "1" and there are no redundant leading zeroes. Here are some examples: "0", "10", "1001".

You are given a *correct* string s.

You can perform two different operations on this string:

1. swap any pair of adjacent characters (for example, "101" "110");
2. replace "11" with "1" (for example, "110" "10").

Let $val(s)$ be such a number that s is its binary representation.

Correct string a is less than some other *correct* string b iff $val(a) < val(b)$.

Your task is to find the minimum *correct* string that you can obtain from the given one using the operations described above. You can use these operations any number of times in any order (or even use no operations at all).

Input

The first line contains integer number n $(1 \le n \le 100)$ — the length of string s.

The second line contains the string s consisting of characters "0" and "1". It is guaranteed that the string s is *correct*.

Output

Print one string — the minimum *correct* string that you can obtain from the given one.

Examples

input

```
4
1001
```

output

```
100
```

input

```
1
1
```

output

```
1
```

Note

In the first example you can obtain the answer by the following sequence of operations: "1001" "1010" "1100" "100".

In the second example you can't obtain smaller answer no matter what operations you use.