Problem D. Dangerous Numbers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

One day, oToToT received a secret hacker mission, instructing him to hack into NTU's school system. After a simple investigation, he discovered that NTU's system had many vulnerabilities. As a seasoned hacker, he easily bypassed them and reached the final stage: password verification.

Despite his efforts, oToToT could not find the password. Since he could see anything accessible online, he suspected that the security personnel might have written the password on paper.

Luckily, oToToT discovered something peculiar about the password verifier. First, the machine only accepts numbers as passwords. For any input number, the verifier checks against M numbers called "dangerous" numbers. If any **substring** of the input number matches any of the dangerous numbers, the machine will raise an error. Otherwise, the input passes. For example, if `12` is a dangerous number, inputs such as `123` and `3212` will trigger an error, while `2142` and `456` will not.

Of course, it's not that simple. In addition to avoiding dangerous numbers, oToToT found that the password must also be **not less than** a minimum value N. That is, if the input number is smaller than N, the machine will also raise an error.

At this point, oToToT was called away for training by Teacher Chou. Can you help oToToT find the **smallest** number that will pass the password verifier without triggering an error?

Input

The first line of input contains a large integer N representing the minimum value for the password ($1 \le |N| \le 3 \cdot 10^5$; here and forth, $|X|$ is the number of digits in X).

The second line contains an integer M representing the number of dangerous numbers ($1 \le M \le 3 \cdot 10^5$).

Each of the next M lines contains a large integer d_i representing the i-th dangerous number ($\sum_{i=1}^{M} |d_i| \le 3 \cdot 10^5$).

Zero is represented as "`0`", and none of the other numbers start with a zero.

Output

Output a single large integer on one line: the **smallest** number that will pass the password verifier without triggering an error. If it's impossible to construct such a number, output `-1` on a single line. Zero should be represented as "`0`", and none of the other numbers should start with a zero.

Example

standard input	*standard output*
339 4 9 34 51 0	352