Problem A. The Bubble Potion Conundrum

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

In the grand alchemical laboratory of Bubbleland, one of the most esteemed competitions among the sorcerers is the art of potion-making. The sorcerers aim to craft the most perfect and refined potion possible, but there's a twist—each potion starts as a mixture of N ingredients, and their goal is to combine and reduce the ingredients into the smallest possible essence.

The sorcerers have discovered a magical operation they can perform repeatedly on their mixture. Each ingredient is represented by a digit in a special number. You are given this mixture as an N-digit number, and the operation works as follows:

Choose any two positions i and j in the mixture's ingredient list. Let A_i and A_j represent the digits (ingredients) at those positions. You can replace both A_i and A_j with the result of $(A_i + A_j) \bmod 10$. Interestingly, you can even select the same index for both i and j, effectively transforming the single digit A_i into $(A_i + A_i) \bmod 10$.

Through this powerful transformation, the sorcerers can refine their potion over and over. Their ultimate goal is to produce the smallest possible potion mixture number that can be obtained through any number of transformations.

You, as a seasoned potion master, must determine the smallest possible number that can be created by performing this operation on the initial N-digit number any number of times.

Input

The first line contains a single integer N: the number of digits in the initial potion mixture ($1 \le N \le 10^6$).

The second line contains a string of length N consisting of digits from 0 to 9, representing the N-digit potion mixture. The first digit is non-zero.

Output

Output a string of length N representing the smallest possible potion mixture number that can be obtained after applying the transformation operation any number of times. The result may contain leading zeros.

Examples

standard input	*standard output*
1 9	2
2 37	00

Note

In the first example, the only operation we can do is to choose to double the single digit that we have. Doing this, we will go from 9 to 8 to 6 to 2, and we can show that 2 is the lowest we can go.

In the second example, we can choose to do the operation on the first and second digits of the number, transforming them both into $(3 + 7) \bmod 10 = 0$.