

# Binary String

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

Given a binary string  $S = S_1S_2 \dots S_n$  of length  $n$ , you need to insert a bitwise operator between every two adjacent digits  $S_i$  and  $S_{i+1}$  (there are  $n - 1$  such positions), so that the final value of the resulting expression is 0.

There are three operators to choose from, with the following rules:

- Bitwise AND ( $\&$ ): for the expression  $a \& b$ , its value is 1 if and only if  $a = b = 1$ ; otherwise, it is 0.
- Bitwise XOR ( $\wedge$ ): for the expression  $a \wedge b$ , its value is 1 if and only if  $a = 1, b = 0$  or  $a = 0, b = 1$ ; otherwise, it is 0.
- Bitwise OR ( $|$ ): for the expression  $a | b$ , its value is 0 if and only if  $a = b = 0$ ; otherwise, it is 1.

Note that in this problem, the precedence of bitwise operators is the same as in C, according to the following rules:

1. Bitwise AND ( $\&$ ) has the highest precedence.
2. Bitwise XOR ( $\wedge$ ) has lower precedence than bitwise AND, but higher precedence than bitwise OR.
3. Bitwise OR ( $|$ ) has the lowest precedence.
4. Operators with the same precedence are evaluated from left to right.

You need to construct a valid sequence of operators such that the final value of the whole expression is 0. If there are multiple valid solutions, output any of them. It can be proved that for every valid input, at least one valid construction always exists.

## Input

The first line contains a positive integer  $n$  ( $2 \leq n \leq 10^5$ ), which denotes the length of the string.

The second line contains a string  $S$  of length  $n$  consisting only of 0 and 1.

## Output

Output one line containing a string of length  $n - 1$ , representing the operators inserted in order. If there are multiple valid solutions, output any of them.

## Example

standard input	standard output
3 000	$\&\&$

## Note

In the sample, the original string is 000. Two bitwise AND operators can be inserted in the middle to make the expression  $0\&0\&0$ , whose result is 0. Therefore, output  $\&\&$ .