6695 Computing Inverses

Finite fields are used in many modern cryptosystems, such AES. In this problem, we consider the finite field $\mathbf{GF}(2^m)$ for some positive integer m.

Let's start with $m = 1$. In $\mathbf{GF}(2)$, there are two elements: 0 and 1. The following tables shows the rule of addition (+) and the rule of multiplication (×).

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

It can be seen that addition in $\mathbf{GF}(2)$ is the *exclusive-or* of the two input bits; and multiplication is the same as ordinary multiplication of integers.

Now consider the case when $m > 1$. The elements of $\mathbf{GF}(2^m)$ can be represented by binary strings of length m. Thus, addition is the *excursive-or* of the two input strings.

To define multiplication in $\mathbf{GF}(2^m)$, it is more convenient to represent elements in $\mathbf{GF}(2^m)$ by a polynomial of degree $m - 1$.

$$a_{m-1}a_{m-2}\cdots a_0 \quad \rightarrow \quad a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_0$$

The following is an example for $m = 4$.

$$\begin{aligned} 0000 &\rightarrow 0 \\ 0001 &\rightarrow 1 \\ 0010 &\rightarrow x \\ 0011 &\rightarrow x+1 \\ &\vdots \\ 1111 &\rightarrow x^3 + x^2 + x + 1 \end{aligned}$$

Let $f(x) = \sum_{i=0}^{m-1} a_i x^i$ and $g(x) = \sum_{i=0}^{m-1} b_i x^i$ be two elements in $\mathbf{GF}(2^m)$. The addition of $f(x)$ and $g(x)$ can also be defined as $f(x)+g(x)$. Note that in doing the addition of the coefficients of the polynomials, it is required to use the addition rule as defined in $\mathbf{GF}(2)$. For example, $(x^3+x+1)+(x^3+x^2+1) = x^2+x$. It is easy to verify that this is the same as the *exclusive-or* as defined above.

To multiply two elements $f(x)$ and $g(x)$ in $\mathbf{GF}(2^m)$, we first multiply $f(x)$ and $g(x)$. In doing the addition and the multiplication of the coefficients of the polynomials, it is required to use the addition rule and the multiplication rule as define in $\mathbf{GF}(2)$. However, the degree of $f(x) \times g(x)$ can be $2m - 2$, and it may not be an element of $\mathbf{GF}(2^m)$. Therefore, it is required to take the remainder of $f(x) \times g(x)$ after dividing by a polynomial $h(x)$ of degree m. Thus, the multiplication of $f(x)$ and $g(x)$ is defined by

$$(f(x) \times g(x)) \bmod h(x).$$

There are rules in selecting the polynomial $h(x)$ to make $\mathbf{GF}(2^m)$ a field. In this problem, you may assume that the polynomial $h(x)$ has been properly chosen so that $\mathbf{GF}(2^m)$ with addition and multiplication operations defined above is a field. Your task is to write a program to compute multiplicative inverses in $\mathbf{GF}(2^m)$. Formally, given a nonzero element $a \in \mathbf{GF}(2^m)$ and $h(x)$, write a program to find an element $b \in \mathbf{GF}(2^m)$ satisfying $a \times b = 1$.

Input

The input to the problem contains many test cases. Each test case is written in one line, which contains two binary strings a and h. The length of h is no more than 20. Note that there are no leading zeros in a and h. The last line of the input file contains a single '`0`'. You do not need to process this line.

Output

The output for each test case is an element $b \in \mathbf{GF}(2^m)$, satisfying $a \times b = 1$. The element b should be printed as a binary string without leading zeros. Print the output of each test case in one line with no leading and tailing spaces.

Sample Input

```
1001 100011011
1100110 100101011
0
```

Sample Output

```
1001111
10011011
```