An error-detection technique used widely in today's computer networks and data storage devices is based on Cyclic Redundancy Check (CRC) codes. A systematic CRC encoder takes two binary inputs, a data word and a generator polynomial, and carries out the necessary calculation to produce the encoded word.

Your task is to write a program to decode a systematic CRC encoded message. In the case of error detection your program should return an ERROR message.

Systematic CRC

Given a data word $D(x)$ of length k, a Systematic Encoder generates the encoded data word $E(x)$ according to the expression:

$$E(x) = X^{n-k}D(x) + R(x)$$

where n is the size of the encoded message, $G(x)$ is the generator polynomial of length $(n-k+1)$ bits, X^{n-k} is the $(n-k)$ term of $G(x)$ and $R(x)$ is the remainder of the modulo-2 division of $X^{n-k}D(x)$ by $G(x)$.

Remark: we can obtain $E(x)$ by shifting the data word that represents $D(X)$ $n-k$ bits to the left (identical to multiplying it by X^{n-k}), and then adding $R(x)$ (where $R(x)$ is obtained by dividing the left-shifted word by $G(x)$).

Example

Let the binary data word 110 represent the original polynomial $D(x) = X^2 + X$, and 11101 represent the generator polynomial $G(x) = X^4 + X^3 + X^2 + 1$. Thus, 1100000 represents $X^4D(x) = X^6 + X^5$, and 1100000 mod 11101=1001 represents the remainder $R(x) = X^4D(x)$ mod $G(x)$. Finally, 1100000+1001=1101001 represents the generated encoded word $E(x) = X^4D(x) + R(x)$.

Input

The input file contains several test cases, each of them as described below.

Three lines containing:

- An integer k representing the length of the original data word ($k \leq 16$);
- A binary sequence (string with caracters '0' and '1') representing the encoded message $E(x)$;
- A binary sequence (string with caracters '0' and '1') representing the generator polynomial $G(x)$.

The binary sequences have maximum length 200.

Output

For each test case, output a single line containing the decoded message, or the word `ERROR` if your program detects that the given $E(x)$ could not have been generated by the given generator polynomial.

Sample Input

```
3
1101001
11101
3
1101011
11101
```

Sample Output

```
110
ERROR
```