

E. Lock Puzzle

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Welcome to another task about breaking the code lock! Explorers Whitfield and Martin came across an unusual safe, inside of which, according to rumors, there are untold riches, among which one can find the solution of the problem of discrete logarithm!

Of course, there is a code lock is installed on the safe. The lock has a screen that displays a string of n lowercase Latin letters. Initially, the screen displays string s . Whitfield and Martin found out that the safe will open when string t will be displayed on the screen.

The string on the screen can be changed using the operation «`shift x`». In order to apply this operation, explorers choose an integer x from 0 to n inclusive. After that, the current string $p = \alpha\beta$ changes to $\beta^R\alpha$, where the length of β is x , and the length of α is $n - x$. In other words, the suffix of the length x of string p is reversed and moved to the beginning of the string. For example, after the operation «`shift 4`» the string «`abcacb`» will be changed with string «`bcacab`», since $\alpha = ab$, $\beta = cacb$, $\beta^R = bcac$.

Explorers are afraid that if they apply too many operations «`shift`», the lock will be locked forever. They ask you to find a way to get the string t on the screen, using no more than 6100 operations.

Input

The first line contains an integer n , the length of the strings s and t ($1 \leq n \leq 2\,000$).

After that, there are two strings s and t , consisting of n lowercase Latin letters each.

Output

If it is impossible to get string t from string s using no more than 6100 operations «`shift`», print a single number -1 .

Otherwise, in the first line output the number of operations k ($0 \leq k \leq 6100$). In the next line output k numbers x_i corresponding to the operations «`shift x_i` » ($0 \leq x_i \leq n$) in the order in which they should be applied.

Examples

input
6 abacbb babcba
output
4 6 3 2 3

input
3 aba bba
output
-1

In the first example, after applying the operations, the string on the screen will change as follows:

1. abacbb bbcaba

2. bbcaba ababbc
3. ababbc cbabab
4. cbabab babcba