Problem F. Shuffle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given a string of even length $s = s_1 \dots s_n$, we define shuffle operation which transforms a string into a new string according to the following rule:

$$\text{shuffle}(s) = s_1 s_3 \dots s_{n-1} s_2 s_4 \dots s_n$$

For example, shuffle(`abcdef`) = `acebdf`.

You are given two strings of equal even length, s and t. How many times do you have to apply shuffle operation to s in order to get t as a result?

In the other words, find minimum k such that $\underbrace{\text{shuffle}(\text{shuffle}(\dots \text{shuffle}}_{k \text{ times}}(s)\dots)) = t$ or report that it is not possible to reach t in any number of operations.

Input

The first line of input contains a string s, the second contains a string t ($|s| = |t|$, $2 \le |s| \le 10^6$, $|s|$ is even). Both strings consist of lowercase English characters.

Output

Print minimum non-negative k such that it is possible to obtain t from s by applying shuffle operation k times (or maybe not applying at all if $k = 0$), or print `-1` if it is impossible.

Examples

standard input	standard output
abcdef aedcbf	2
petrozavodsk poztsvoedark	3
qwerty ytrewq	-1