D. Obsessive String

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

Hamed has recently found a string t and suddenly became quite fond of it. He spent several days trying to find all occurrences of t in other strings he had. Finally he became tired and started thinking about the following problem. Given a string s how many ways are there to extract $k \geq 1$ non-overlapping substrings from it such that each of them contains string t as a substring? More formally, you need to calculate the number of ways to choose two sequences $a_1, a_2, \ldots, a_k$ and $b_1, b_2, \ldots, b_k$ satisfying the following requirements:

- $k \geq 1$
- t is a substring of string $s_{a_i}s_{a_i+1}\ldots s_{b_i}$ (string s is considered as 1-indexed).

As the number of ways can be rather large print it modulo $10^9 + 7$.

Input

Input consists of two lines containing strings s and t ($1 \leq |s|, |t| \leq 10^5$). Each string consists of lowercase Latin letters.

Output

Print the answer in a single line.

Examples

input

```
ababa
aba
```

output

```
5
```

input

```
welcometoroundtwohundredandeightytwo
d
```

output

```
274201
```

input

```
ddd
d
```

output

```
12
```