

## [E] String Inversions

Time limit: 1 second  
Memory limit: 65535 kBytes

### Description

Ivan came up with an interesting idea for a programming contest problem: given a string  $S$  consisting of lowercase English letters, count the number of *inversions* in it. An inversion is a pair of indices  $i < j$  such that the letter  $S_i$  appears later in the alphabet than  $S_j$ .

However, other jury members reminded Ivan that there are already many well-known algorithms for counting inversions. To make the problem more original, Ivan suggested the following modification: in addition to  $S$ , a positive integer  $N$  is given, and the task is to compute the number of inversions in  $N \cdot S$  – the string formed by repeating  $S$  exactly  $N$  times.

The jury is now curious how to solve this new version of the problem. Help them by finding the solution.

### Input

The first line of the input contains the string  $S$  consisting of lowercase English letters.

The second line contains the integer  $N$  – the number of repetitions of  $S$ .

### Output

Output a single line with the number of inversions in the repeated string. This number can be large, so print it modulo  $10^9 + 7$ .

### Constraints

- $1 \leq |S| \leq 100\,000$
- $1 \leq N \leq 10^{12}$

### Example

Input	Output
ba 1	1
ab 3	3
zkba 1	6
cab 7	77