Friendly Note (neighborly)

Italian people have a very peculiar way to permanently settle a dispute: they prepare a *friendly note*, politely describing their personal point of view about the matter, and deliver it secretly to each other. In order to add a dash of fun, they usually prepare the note by cutting and pasting sequences of characters from newspapers.

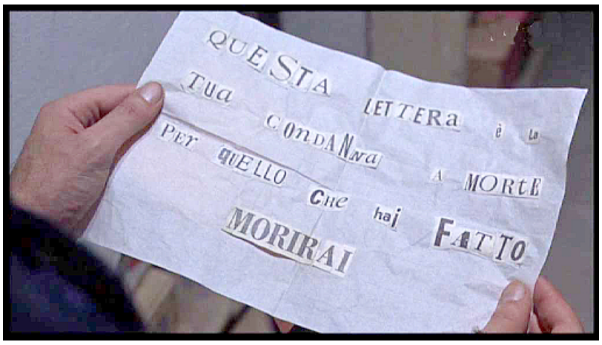

Figure 1: A typical Italian friendly note.

Giorgio makes no exception, and his long-standing dispute with his neighbour (about noise complaints) requires him to finally take action and defend his honour. He thus collected *a lot* of copies of his favourite newspaper, *La Stampa*, and is now ready to prepare his friendly note N.

Since Giorgio feels unconfortable with scissors, he would prefer to cut and paste the smallest number of character sequences from the newspaper content S. Help Giorgio produce text N by juxtaposing **as few** contiguous subsequences of text S as possible, assuming that these subsequences can freely overlap (remember: he collected *a lot* of copies of that newspaper).

☞ Among the attachments of this task you may find a template file `neighborly.*` with a sample incomplete implementation.

Input

The first line contains string N. The second line contains string S.

Output

You need to write a single line with an integer: the least number of subsequences Giorgio needs to cut and paste.

Constraints

- Strings N and S consist of ASCII printable characters only.
- $1 \leq \text{len}(N) \leq 1\,000\,000$.
- $1 \leq \text{len}(S) \leq 10\,000$.
- It is always possible to obtain the string N from substrings of S.

Scoring

Your program will be tested against several test cases grouped in subtasks. The score in each subtask will be calculated as the **minimum** score obtained in any of its test cases, multiplied by the value of the subtask. The score in a test case will be **0** if you output a number that is lower than the optimal solution or higher than the length of N. Otherwise, it will be calculated as:

$$\frac{\Delta_{\text{max}} - \Delta_{\text{out}}}{\Delta_{\text{max}} + 10\Delta_{\text{out}}}$$

where Δ_{out} is the difference between your solution and the optimal solution, and Δ_{max} is the difference between the length of N and the optimal solution.

- **Subtask 1** (0 points) Examples.
  ●○○○○
- **Subtask 2** (15 points) N consists of a single character repeated multiple times.
  ●○○○○
- **Subtask 3** (20 points) $\text{len}(N), \text{len}(S) \leq 20$.
  ●●●○○
- **Subtask 4** (25 points) $\text{len}(N) \leq 1000$.
  ●●●○○
- **Subtask 5** (25 points) $\text{len}(S) \leq 1000$.
  ●●●○○
- **Subtask 6** (15 points) No additional limitations.
  ●●●○○

Examples

input.txt	output.txt
@dd10 v1cin* 0 v@d1ci*n	6
caro vicino vuoi smetterla di fare casino Il caro vita sale ancora: se sei mancino vuoi smettere di comprare oggetti appositi, per esempio. Fare cassa e' sempre piu' difficile; parola di fiscalista.	5

Explanation

In the **first sample case**, the friendly note can be composed by the following fragments: `"@d"`, `"d1"`, `"0 v"`, `"1ci"`, `"n"`, `"*"`.

In the **second sample case**, the friendly note can be composed by the following fragments: `"caro vi"`, `"cino vuoi smetter"`, `"la di f"`, `"are cas"`, `"ino"`.