

Problem H. Query Matching

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds (*3 seconds for Java*)
Memory limit: 128 mebibytes

One can tell a lot about a person by the search queries this person issues.

But what is even more characteristic of a person is the way she organizes her search queries into a query session. Even typos she makes and her manner of correcting them matters!

Having acknowledged this fact, Yandex is developing a new dating service with an unprecedented feature: couples are matched basing on their search engine query sessions.

You are given query session logs of two persons. Your task is to compute the score of the match between them.

In the context of this problem, you can assume that a query session is a sequence of keyboard presses by a user. There are only two types of key-presses in each query session. When the user hits a lowercase English letter, this letter is appended to the end of the query in the query input box. When the user hits backspace, the last letter in the query input box (the most recently entered one) disappears. If the user hits backspace when the query input box is empty, nothing happens.

Given a log of key presses in a search query session, one could consider a set of all nonempty strings of lowercase English letters that appeared in the query input box during this search query session. Moreover, one could construct the set of all non-empty substrings of such strings.

Consider two sets of such substrings for two particular users. The score of the match between them is the number of common substrings in these two sets.

Input

Input consists of two search query session logs presented on two separate lines. Each log is a string that consists entirely of lowercase English letters and “<” characters. The “less-than” character denotes a backspace hit by the user.

Each log contains no less than 1 and no more than 100 000 English letters. The total length of each log does not exceed 300 000 characters.

Output

Output a single integer that is the answer to this problem.

Example

standard input	standard output
g<yandex yand<<hoo	10