

COMPRESSION

The reset password link only works in incognito!

My last name is hyphenated and the form won't accept it!

The Enemy-o-matic keeps reporting that my only enemy is myself!

After launching nationwide, you're drowning in bug reports - you knew it was a bad idea to trust the intern to push to production... To triage the bugs and figure out which are highest priority, you wish to determine how often each bug appears.

The Enemy-o-matic spews out log information at an impressive rate, and to reduce volume it is passed through a compressor. After compression, the log to search consists of N upper-case letters, $c_1 c_2 \dots c_N$.

For the i 'th of B bugs, you know an indicator string s_i , such that **each occurrence of s_i in the log is a separate instance of the bug**. Note that these occurrences **can overlap**. For the indicator string s_i with length l_i , an occurrence is a position j such that the string $c_j c_{j+1} \dots c_{j+l_i-1}$ equals the string s_i .

Given the log and the indicator strings, find the number of times each bug has manifested.

Input

The first line contains two space-separated integers, N and B .

The next line contains N uppercase letters $c_1 c_2 \dots c_N$ which is the compressed log.

The next B lines contain the indicator strings for each bug. The i 'th of these contains a single string with l_i uppercase letters.

Output

For each bug i , let ans_i be the number of manifestations of the i 'th bug, which is the number of times the indicator string appears in the log. Find the sum $\sum_{i=1}^B ans_i$.

Constraints

Let $L = \sum_{i=1}^B l_i$ be the sum of the lengths of the indicator strings. In all test cases, $1 \leq N, B, L \leq 10^5$. Beyond the sample input, the tests are divided into batches with additional constraints. Time limits below are for C/C++; Ocaml gets 2x, Java 3x, and Python 10x.

- 27 points satisfy $N \cdot L \leq 3 \cdot 10^5$. TL: 100ms.
- 30 points satisfy $N \cdot Q \leq 3 \cdot 10^5$. TL: 100ms.
- 43 points satisfy no further constraints. TL: 1200ms.

Sample explanation

C appears one time at position 3, BB appears 3 times at positions 4, 5, 8, and ABCB appears one time at position 1. The answer is $1 + 3 + 1 = 5$.

[View submissions \(https://cs124.seas.harvard.edu/problem/COMPRESSION/code-submission\)](https://cs124.seas.harvard.edu/problem/COMPRESSION/code-submission)

Test cases

Input	Output	Points	Timeout
9 3 ABCBBBABBB C ABCB	5	0	100 ms
<i>Hidden</i>	<i>Hidden</i>	9	100 ms
<i>Hidden</i>	<i>Hidden</i>	9	100 ms
<i>Hidden</i>	<i>Hidden</i>	9	100 ms
<i>Hidden</i>	<i>Hidden</i>	10	100 ms
<i>Hidden</i>	<i>Hidden</i>	10	100 ms
<i>Hidden</i>	<i>Hidden</i>	10	100 ms
<i>Hidden</i>	<i>Hidden</i>	11	1200 ms
<i>Hidden</i>	<i>Hidden</i>	11	1200 ms
<i>Hidden</i>	<i>Hidden</i>	11	1200 ms
<i>Hidden</i>	<i>Hidden</i>	10	500 ms

[Download \(https://cs124.seas.harvard.edu/problem/COMPRESSION/test-cases\)](https://cs124.seas.harvard.edu/problem/COMPRESSION/test-cases)

Inspired by the "Ultra Cool Programming Contest Control Centre" by Sonny Chan.
Modified for CS 124 by Neal Wu (<https://github.com/nealwu>), with design help from Martin Camacho.
Further refined by Nikhil Benesch (<https://github.com/benesch>).