

# Pattern Matching in A Minor “Low Space”

Input file:           standard input  
Output file:         standard output  
Time limit:          5 seconds  
Memory limit:       4 megabytes

Bobo is asked to solve the following classical pattern matching problem:

Given two strings  $s$  and  $t$ , calculate how many times  $s$  appears in  $t$  as a substring?

The correct memory limit for this problem is 1 megabytes.

## Input

The first line contains two integers  $n, m$  ( $1 \leq n, m \leq 10^7$ ), denoting the length of strings  $s$  and  $t$ , respectively.

The next line contains a string  $s$  of length  $n$ , consisting of only lowercase English letters.

The next line contains a string  $t$  of length  $m$ , consisting of only lowercase English letters.

Please refer to the “Note” section for more specific restrictions.

## Output

Output an integer in a line, denoting the number of times  $s$  appears in  $t$  as a substring.

## Examples

standard input	standard output
3 7 aba abababc	2
1 11 a abracadabra	5
8 10 possible impossible	1
10 21 pleasenote theunusualmemorylimit	0

## Note

Due to technical issues, this problem is implemented as an interactive problem. You can safely treat it as a normal traditional problem, except that your program is only allowed to read the input sequentially at most once. For example, it is explicitly prohibited to read the entire input, then rewind the input stream to the beginning and re-read the input. **Also, you should read all input before outputting the answer, otherwise you may not get the correct verdict.**