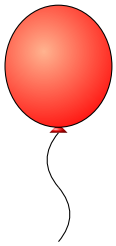
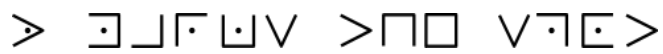


Coded



As a spy for a secret international agency, you know that all classical codes are breakable. With your handler you decide on a new scheme to get warnings from the agency that you think is safe.

As usual, your handler will send you numbers (that are themselves codes) by hiding them into newspapers. To retrieve the message, you first extract the text of the newspaper. For this you only keep the 26 lowercase letters from 'a' to 'z' as well as the 26 uppercase letters 'A' to 'Z' (all the spaces, punctuation, and accented characters, etc., are removed).

Your scheme then relies on a secret sequence of letters S. The secret number your handler sends is the number of times an anagram of S appears. Write a program that, given the secret word S and the text T, counts the number of possible appearances of S in T; that is, every sequence of $|S|$ consecutive letters in T that is a permutation of letters in S.

Limits

- S and T are only made of the letters 'a' to 'z' and 'A' to 'Z'.
- $1 \leqslant |S| \leqslant 3\,000$
- $1 \leqslant |T| \leqslant 3\,000\,000$

Note that both S and T are not necessarily made of real English words.

Input

The first line of the input consists of two space-separated integers $|S|$ and $|T|$, the lengths of S and T. The second line contains S the secret word and T contains the text.

Output

The output should contain a single integer M which is the number of occurrences of an anagram of S in T.

Sample Input 1

```
4 20
abba
bababaisnotarealword
```

Sample Output 1

```
3
```

Sample Explanation 1

"baba" is an anagram of "abba" and there are two occurrences of "baba" (starting at the first position of the word and at the third), and "abab" is also an anagram of "abba" and there is one occurrence of it starting at the second position of the word.

Sample Input 2

```
1 13
a
thecakeisalie
```

Sample Output 2

```
2
```