Problem Statement

You're creating a new programming language with some exciting new features! Any programming language can check if two strings are matching, but you'd like yours to be able to check if they're almost matching.

More specifically, we'll say two strings are almost matching if they're equal in length and all of their corresponding characters are the same except for one.

For example:
- "cat" and "bat" are almost matching
- but "cat" and "dog" are not.

For the sake of efficiency, you're planning on testing the feature by using a single string and comparing its substrings.

Given a string s and an integer k, your task is to find the number of pairs of substrings of s that are almost matching but differ at their k-th character (0-based).

It is necessary that the length of both substrings exceeds k (otherwise the strings wouldn't have a k-th character).

Also note that substrings are determined by their indices, so there could potentially be multiple instances of the same word.

For example, in the word "ingratiating", the substring "ing" beginning at index 0 is considered distinct from the one at index 9 (and there are also two distinct "ati" substrings).

Example

For s = "abacaba" and k = 1, the output should be:

solution(s, k) = 8

Using (i, j) to represent the start and end indices of the first substring, and (l, m) to represent the indices of the second substring, the 8 pairs are:

- "aba" vs "aca" -> (i=0, j=2, l=1, m=3)
- "aba" vs "aca" -> (i=4, j=6, l=1, m=3)
- "aca" vs "aba" -> (i=1, j=3, l=0, m=2)
- "aca" vs "aba" -> (i=1, j=3, l=4, m=6)
- "ac" vs "ab" -> (i=1, j=2, l=0, m=1)
- "ac" vs "ab" -> (i=1, j=2, l=4, m=5)
- "ab" vs "ac" -> (i=0, j=1, l=1, m=2)
- "ab" vs "ac" -> (i=4, j=5, l=1, m=2)

Input / Output

[execution time limit] 3 seconds (Java)

[memory limit] 1 GB

Input

- string s

  A string consisting only of lowercase English letters.

  Constraints:

  1 <= s.length <= 200

- integer k

  Constraints:

  0 <= k < s.length

Output

- integer

  The number of different substring pairs as described above.

[Java] Syntax Tips

```
// Prints help message to the console
// Returns a string
// Globals declared here will cause a compilation error
// Declare variables inside the function instead!

String helloWorld(String name) {
    System.out.println("This prints to the console");
```

```
    return "Hello, " + name;
}
```