Given a string s, return *the number of* ***unique palindromes of length three*** *that are a* ***subsequence*** *of* s.

Note that even if there are multiple ways to obtain the same subsequence, it is still only counted **once**.

A **palindrome** is a string that reads the same forwards and backwards.

A **subsequence** of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

* For example, "ace" is a subsequence of "abcde".

**Example 1:**

Input: s = "aabca"  
Output: 3  
Explanation: The 3 palindromic subsequences of length 3 are:  
- "aba" (subsequence of "aabca")  
- "aaa" (subsequence of "aabca")  
- "aca" (subsequence of "aabca")

**Example 2:**

Input: s = "adc"  
Output: 0  
Explanation: There are no palindromic subsequences of length 3 in "adc".

**Example 3:**

Input: s = "bbcbaba"  
Output: 4  
Explanation: The 4 palindromic subsequences of length 3 are:  
- "bbb" (subsequence of "bbcbaba")  
- "bcb" (subsequence of "bbcbaba")  
- "bab" (subsequence of "bbcbaba")  
- "aba" (subsequence of "bbcbaba")

**Constraints:**

* 3 <= s.length <= 105
* s consists of only lowercase English letters.