Given a string `word`, return *the **sum of the number of vowels** (`'a'`, `'e'`, `'i'`, `'o'`, and `'u'`) in every substring of* `word`.

A **substring** is a contiguous (non-empty) sequence of characters within a string.

Note: Due to the large constraints, the answer may not fit in a signed 32-bit integer. Please be careful during the calculations.

Example 1:

```
Input: word = "aba"
Output: 6
Explanation:
All possible substrings are: "a", "ab", "aba", "b", "ba", and "a".
- "b" has 0 vowels in it
- "a", "ab", "ba", and "a" have 1 vowel each
- "aba" has 2 vowels in it
Hence, the total sum of vowels = 0 + 1 + 1 + 1 + 1 + 2 = 6.
```

Example 2:

```
Input: word = "abc"
Output: 3
Explanation:
All possible substrings are: "a", "ab", "abc", "b", "bc", and "c".
- "a", "ab", and "abc" have 1 vowel each
- "b", "bc", and "c" have 0 vowels each
Hence, the total sum of vowels = 1 + 1 + 1 + 0 + 0 + 0 = 3.
```

Example 3:

```
Input: word = "ltcd"
Output: 0
Explanation: There are no vowels in any substring of "ltcd".
```

Constraints:

- `1 <= word.length <= 10^5`
- `word` consists of lowercase English letters.

```python
class Solution(object):
    def countVowels(self, word):
        """
        :type word: str
        :rtype: int
        """
        # n = len(word)
        # char[0] is in n substrings
        # char[1] is in (n-1) + (n-1)
        # char[2] is in (n-2) + (n-2) + (n-2)

        total = 0
        n = len(word)
        vowels = {'a', 'e', 'i', 'o', 'u'}
        for i in range(n):
            if word[i] in vowels:
                total += (n-i) * (i+1)

        return total

```