



Sherlock and Anagrams ☆

Problem

Submissions

Leaderboard

Editorial

Topics

Two strings are *anagrams* of each other if the letters of one string can be rearranged to form the other string. Given a string, find the number of pairs of substrings of the string that are anagrams of each other.

For example $s = mom$, the list of all anagrammatic pairs is $[m, m]$, $[mo, om]$ at positions $[[0], [2]]$, $[[0, 1], [1, 2]]$ respectively.

Function Description

Complete the function *sherlockAndAnagrams* in the editor below. It must return an integer that represents the number of anagrammatic pairs of substrings in s .

sherlockAndAnagrams has the following parameter(s):

- s : a string .

Input Format

The first line contains an integer q , the number of queries.

Each of the next q lines contains a string s to analyze.

Constraints

$$1 \leq q \leq 10$$

$$2 \leq |s| \leq 100$$

String s contains only lowercase letters $\in \text{ascii}[a-z]$.

Output Format

For each query, return the number of unordered anagrammatic pairs.

Sample Input 0

```
2
abba
abcd
```

Sample Output 0

```
4
0
```

Explanation 0

The list of all anagrammatic pairs is $[a, a]$, $[ab, ba]$, $[b, b]$ and $[abb, bba]$ at positions $[[0], [3]]$, $[[0, 1], [2, 3]]$, $[[1], [2]]$ and $[[0, 1, 2], [1, 2, 3]]$ respectively.

No anagrammatic pairs exist in the second query as no character repeats.

Sample Input 1

```
2
ifailuhkqq
kkkk
```



Sample Output 1

3
10

Explanation 1

For the first query, we have anagram pairs $[i, i]$, $[q, q]$ and $[ifa, fai]$ at positions $[[0], [3]]$, $[[8], [9]]$ and $[[0, 1, 2], [1, 2, 3]]$ respectively.

For the second query:

There are 6 anagrams of the form $[k, k]$ at positions $[[0], [1], [0], [2]], [[0], [3]], [[1], [2]], [[1], [3]]$ and $[[2], [3]]$.

There are 3 anagrams of the form $[kk, kk]$ at positions $[[0, 1], [1, 2]], [[0, 1], [2, 3]]$ and $[[1, 2], [2, 3]]$.

There is 1 anagram of the form $[kkk, kkk]$ at position $[[0, 1, 2], [1, 2, 3]]$.

Sample Input 2

1
cdcd

Sample Output 2

5

Explanation 2

There are two anagrammatic pairs of length 1: $[c, c]$ and $[d, d]$.

There are three anagrammatic pairs of length 2: $[cd, dc]$, $[cd, cd]$, $[dc, cd]$ at positions $[[0, 1], [1, 2]], [[0, 1], [2, 3]], [[1, 2], [2, 3]]$ respectively.

Python 3



```

1  #!/bin/python3
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8  from collections import Counter
9
10 # Complete the sherlockAndAnagrams function below.
11 def sherlockAndAnagrams(s):
12     buckets = {}
13     for i in range(len(s)):
14         for j in range(1, len(s) - i + 1):
15             key = frozenset(Counter(s[i:i+j]).items()) # O(N) time key extract
16             buckets[key] = buckets.get(key, 0) + 1
17     count = 0
18     for key in buckets:
19         count += buckets[key] * (buckets[key]-1) // 2
20     return count
21
22 if __name__ == '__main__':
23     fptr = open(os.environ['OUTPUT_PATH'], 'w')
24 
```