Practice Compete Jobs Rank Leaderboard nac

Dashboard > Algorithms > Strings > Sherlock and Anagrams

Badge Progress (Details)

Points: 2392.04 Rank: 4032

Sherlock and Anagrams

by darkshadows

Problem Submissions Leaderboard Discussions Editorial Topics

Given a string S, find the number of "unordered anagrammatic pairs" of substrings. In other words, find the number of *unordered* pairs of substrings of S that are anagrams of each other.

Two strings are **anagrams** of each other if the letters of one string can be rearranged to form the other string.

Input Format
First line contains T, the number of testcases. Each testcase consists of string S in one line.

Constraints
$1 \leq T \leq 10$
$2 \leq length(S) \leq 100$
String S contains only the lowercase letters of the English alphabet.

Output Format
For each testcase, print the required answer in one line.

Sample Input#00

```
2
abba
abcd
```

Sample Output#00

```
4
0
```

Sample Input#01

```
5
ifailuhkqq
hucpoltgty
ovarjsnrbf
pvmupwjjjf
iwwhrlkpek
```

Sample Output#01

```
3
2
2
6
3
```

Explanation

Sample00

Let's say $S[i, j]$ denotes the substring $S_i, S_{i+1}, \cdots, S_j$.

testcase 1:
For S= abba , anagrammatic pairs are: $\{S[1,1], S[4,4]\}$, $\{S[1,2], S[3,4]\}$, $\{S[2,2], S[3,3]\}$ and $\{S[1,3], S[2,4]\}$.

testcase 2:
No anagrammatic pairs.

Sample01

Left as an exercise to you.

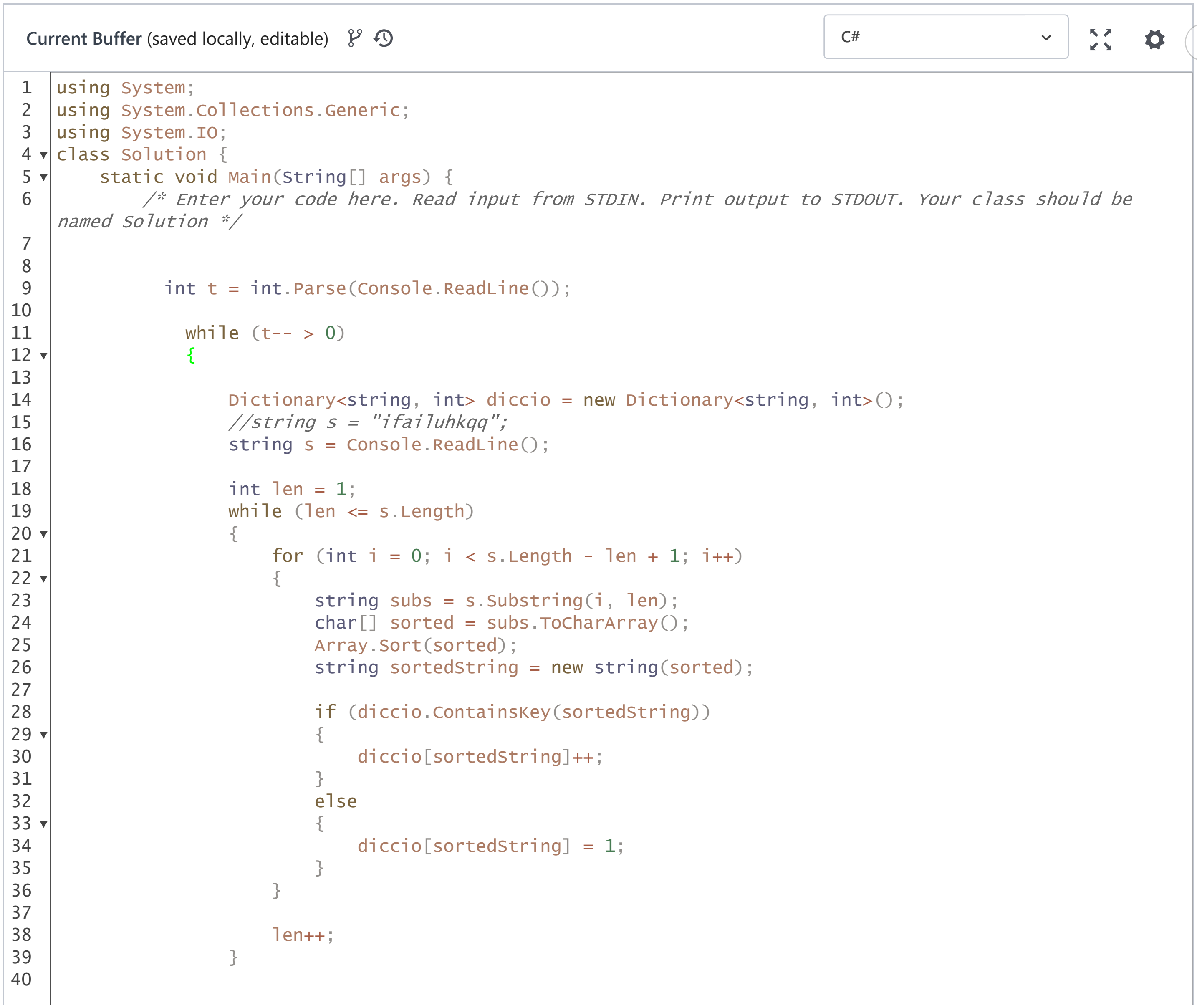
f 🐦 in

Submissions: 11888
Max Score: 50
Difficulty: Medium

Rate This Challenge:
★★★★★ Thanks!

Need Help?
Anagram

More



Current Buffer (saved locally, editable)

C#

```
using System;
using System.Collections.Generic;
using System.IO;
class Solution {
    static void Main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution */


          int t = int.Parse(Console.ReadLine());

            while (t-- > 0)
            {

                Dictionary<string, int> diccio = new Dictionary<string, int>();
                //string s = "ifailuhkqq";
                string s = Console.ReadLine();

                int len = 1;
                while (len <= s.Length)
                {
                    for (int i = 0; i < s.Length - len + 1; i++)
                    {
                        string subs = s.Substring(i, len);
                        char[] sorted = subs.ToCharArray();
                        Array.Sort(sorted);
                        string sortedString = new string(sorted);

                        if (diccio.ContainsKey(sortedString))
                        {
                            diccio[sortedString]++;
                        }
                        else
                        {
                            diccio[sortedString] = 1;
                        }
                    }

                    len++;
                }

                int contPares = 0;
                foreach (KeyValuePair<string, int> kvp in diccio)
                {
                    //Console.WriteLine(kvp.Key + " " + kvp.Value);
                    contPares += (kvp.Value * (kvp.Value - 1)) / 2;
                }

                Console.WriteLine(contPares);

            }


    }
}
```

Line: 50 Col: 14

Upload Code as File ☐ Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

✓ Test Case #0 ✓ Test Case #1 ✓ Test Case #2

✓ Test Case #3 ✓ Test Case #4 ✓ Test Case #5

Next Challenge

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature