**Count Occurences of Anagrams**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)[Microsoft](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Microsoft&isCmp=1)

Given a word and a text, return the count of the occurences of anagrams of the word in the text(For eg: anagrams of word **for** are **for, ofr, rof** etc.))

**Input:**  
The first line of input contains an integer T denoting the number of test cases. The description of T test cases follows.  
The first line of each test case contains a text  consisting of only lowercase Latin Letters.  
The second line contains a word consisting of only lowercase Latin Letters.

**Output:**  
Print the count of the occurences of anagrams of the word in the text.

**Constraints:**  
1 <= T <= 50  
1 <= |word|<= |text| <= 50  
here |word| denotes the size of word and |text| denotes the size of text

**Example:**  
Input:  
2  
forxxorfxdofr  
for  
aabaabaa  
aaba

Output:  
3  
4

**Explaination:  
for, orf and ofr**appears in the first test case and hence answer is 3.

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=259>

#include <iostream>

#include <stdio.h>

#include <vector>

#include <algorithm>

#include <limits>

using namespace std;

int main() {

    int t;

    scanf("%d", &t);

    while(t--) {

       string text, word;

       cin >> text;

       cin >> word;

       int ans =0;

       for(int i =0; i < text.size() - word.size() + 1; i++) {

            std::string subs = text.substr(i, word.size());

            std::sort(subs.begin(), subs.end());

            std::sort(word.begin(), word.end());

            if(subs == word) {

                ans++;

            }

       }

       cout << ans << endl;

    }

     //system("pause");

    return 0;

}