**Group Anagrams Together**

Submissions: [5371](https://practice.geeksforgeeks.org/problem_submissions.php?pid=2071)  Accuracy:

45.18%

   Difficulty: [Easy](https://practice.geeksforgeeks.org/Easy/0/0/)   Marks: 2

Show Topic Tags   

[Amazon](https://practice.geeksforgeeks.org/company/Amazon/)[Goldman Sachs](https://practice.geeksforgeeks.org/company/Goldman%20Sachs/)[Snapdeal](https://practice.geeksforgeeks.org/company/Snapdeal/)

Given an array of words, print the count of all anagrams together in sorted order (increasing order of counts).  
For example, if the given array is {“cat”, “dog”, “tac”, “god”, “act”}, then grouped anagrams are “(dog, god) (cat, tac, act)”. So the output will be 2 3.

**Input:**  
First line consists of T test case. First line of every test case consists of N, denoting the number of words in array. Second line of every test case consists of words of array.

**Output:**  
Single line output, print the counts of anagrams in increasing order.

**Constraints:**  
1<=T<=100  
1<=N<=50

**Example:  
Input:**  
2  
5  
act cat tac god dog  
3  
act cat tac  
**Output:**  
2 3  
3

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/k-anagrams-1/0/?ref=self#ExpectOP) option \*\*

Contributor: Saksham Raj Seth  
[Author: saksham seth](https://auth.geeksforgeeks.org/user/saksham%20seth/practice/)

<https://practice.geeksforgeeks.org/problems/k-anagrams-1/0/?ref=self>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

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

while (t-- > 0)

{

int n = int.Parse(Console.ReadLine().Trim());

string[] s = Console.ReadLine().Trim().Split(' ');

Dictionary<string, int> dic = new Dictionary<string, int>();

for (int i = 0; i < s.Length; i++)

{

char[] ch = s[i].ToCharArray();

Array.Sort(ch);

string key = new string(ch);

if (dic.ContainsKey(key))

{

dic[key]++;

}

else

{

dic[key] = 1;

}

}

int[] vals = dic.Values.ToArray();

Array.Sort(vals);

for (int i = 0; i < vals.Length; i++)

{

Console.Write(vals[i] + " ");

}

Console.WriteLine();

}

//string[] s = { "cat", "dog", "tac", "god", "act" };

Console.ReadLine();

}

}

}