**Check if frequencies can be equal**

Submissions: [6543](https://practice.geeksforgeeks.org/problem_submissions.php?pid=1771)  Accuracy:

18.15%

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

Associated Course(s): [Geeks Classes in Noida](https://practice.geeksforgeeks.org/courses/geeks-classes-in-noida/)

Show Topic Tags   

Given a string s which contains lower alphabetic characters, the task is to check if its possible to  remove at most one character from this string in such a way that frequency of each distinct character becomes same in the string.  
Examples:

Input : s = “xyyz”

Output : 1

We can remove character ’y’ from above

string to make the frequency of each

character same.

Input : s = “xxxxyyzz”

Output : 0

It is not possible to make frequency of

each character same just by removing at

most one character from above string.

**Input:**  
The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains a string s.  
  
**Output:**  
For each test case in a new line print 1 if its possible to make frequencies of all characters equal else print 0.  
  
**Constraints:**  
1<=T<=1000  
1<=length of strings <=10000  
  
**Example:  
Input:**  
2  
xyyz  
xxxxyyzz  
**Output:**  
1  
0

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

[Author: Shubham Joshi 1](https://auth.geeksforgeeks.org/user/Shubham%20Joshi%201/practice/)

<https://practice.geeksforgeeks.org/problems/check-frequencies/0/?ref=self>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp3

{

class Program

{

static int CheckIfFrequenciesCanBeEqual(string s)

{

int[] hash = new int[26];

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

{

hash[s[i] - 'a']++;

}

int[] cont = new int[s.Length + 1];

for (int i = 0; i < 26; i++)

{

if (hash[i] > 0)

{

cont[hash[i]]++;

}

}

List<int> cantGrupos = new List<int>();

//int pre\_i = -1;

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

{

if (cont[i] > 0)

{

cantGrupos.Add(cont[i]);

//if (pre\_i != -1 && i - pre\_i > 1) return 0;

//pre\_i = i;

}

}

if (cantGrupos.Count > 2) return 0;

if(cantGrupos.Count == 2)

{

if (cantGrupos.First() == 1) return 1;

if (cantGrupos.Last() > 1) return 0;

}

////if (cantGrupos.Sum() > 2) return 0;

//for (int i = 0; i + 1 < cantGrupos.Count; i++)

//{

// if (cantGrupos[i] <= cantGrupos[i + 1]) return 0;

//}

return 1;

}

static void Main(string[] args)

{

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

//while (t-- > 0)

//{

// string s = Console.ReadLine().Trim();

// Console.WriteLine(CheckIfFrequenciesCanBeEqual(s));

//}

//string s = "xxxxyyzz";

//string s = "clbhhycls";

//string s = "xxxxyyzz";

//string s = "xyyz";

//string s = "evjxpnqgmvfjl";

//string s = "ehuuroaidj"; //1

//string s = "evjxpnqgmvfjl"; //0

string s = "cceea";//1

Console.WriteLine(CheckIfFrequenciesCanBeEqual(s));

Console.ReadLine();

}

}

}

static int CheckIfFrequenciesCanBeEqual(string s)

{

int[] hash = new int[26];

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

{

hash[s[i] - 'a']++;

}

int[] cont = new int[s.Length + 1];

for (int i = 0; i < 26; i++)

{

if (hash[i] > 0)

{

cont[hash[i]]++;

}

}

//List<int> cantGrupos = new List<int>();

int cantGrupos = 0;

int a = -1, b = -1;

int pre\_i = -1;

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

{

if (cont[i] > 0)

{

//cantGrupos.Add(cont[i]);

cantGrupos++;

if (pre\_i != -1 && i - pre\_i > 1) return 0;

pre\_i = i;

if(a == -1)

{

a = cont[i];

}

else if(a > -1 && b == -1)

{

b = cont[i];

}

}

}

if (cantGrupos > 2) return 0;

if(cantGrupos == 2)

{

if (a == 1) return 1;

if (b > 1) return 0;

}

return 1;