A *palindrome* is a string that reads the same left-to-right and right-to-left. For example, "Madam, I'm Adam" and "Poor Dan is in a droop" are both *palindromes*. Note that letter case and non-alphanumeric characters should be ignored when deciding whether a string is a *palindrome* or not.

А string x is an *anagram* of another string y if you can obtain y by rearranging the letters of x. For example, "cinema" is an *anagram* of "iceman", and vice versa. Note that the string and its *anagram* must have the same length. By definition, the string is **not** considered as an *anagram* of itself. In *anagrams*, non-alphanumeric characters and letter case are important. For instance, "Oo" is not the same as "oO", making "Oo" an *anagram* of "oO" and vice versa.

Given a message, your task is to determine whether there is an *anagram* of the message that is also a *palindrome*.

**Example**

* For message = "abab", the output should be  
  hasPalindromicAnagram(message) = true.

Among the *anagrams* of "abab", there are two strings that are also *palindromes* ("abba" and "baab"), so the answer is true.

* For message = "bob", the output should be  
  hasPalindromicAnagram(message) = false.

The only rearrangement of the letters in the string "bob" that is a *palindrome* is the word itself, but this is not an *anagram* as defined above. Therefore, the answer is false.

* For message = "heh!", the output should be  
  hasPalindromicAnagram(message) = true.

"!heh", "h!eh" and "he!h" are all *palindromes* and all of them are *anagrams* of "heh!". Remember that according to the rules laid out above, non-alphanumeric characters are ignored in *palindromes* but need to be considered in *anagrams*.

* **[time limit] 3000ms (cs)**
* **[input] string message**

A string containing at least one alphanumeric character.

0 < message.length ≤ 20

* **[output] boolean**

<https://codefights.com/challenge/gRkR2xSSNnnAWADmj>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static bool esPalin(string message)

{

char[] rev = message.ToCharArray();

Array.Reverse(rev);

return new string(rev) == message;

}

static bool hasPalindromicAnagram(string message)

{

string message\_limpio = "";

foreach (char ch in message)

{

if (char.IsLetterOrDigit(ch))

{

message\_limpio += char.ToLower(ch);

}

}

//si no es palindromo me fijo si PUEDE SER palindromo

Dictionary<char, int> frec = message\_limpio.ToCharArray().GroupBy(x => x)

.ToDictionary(x => x.Key, x => x.Count());

int cant\_impares = 0;

int cant\_pares = 0;

foreach (KeyValuePair<char, int> kvp in frec)

{

if (kvp.Value % 2 != 0)

{

cant\_impares++;

}

else

{

cant\_pares++;

}

}

if (cant\_impares > 1)//no puede ser palindromo

{

return false;

}

else if (cant\_impares <= 1) //si puede ser palindromo

{

if (!esPalin(message\_limpio))

{

return true;

}

bool tiene\_mayusculas = false;

bool tiene\_minusculas = false;

bool tiene\_caracteres\_especiales = false;

// bool tiene\_digitos = false;

foreach (char ch in message)

{

if (char.IsLower(ch))

{

tiene\_minusculas = true;

}

if (char.IsUpper(ch))

{

tiene\_mayusculas = true;

}

if (!char.IsLetterOrDigit(ch))

{

tiene\_caracteres\_especiales = true;

}

}

if (esPalin(message) && cant\_pares < 2 && !tiene\_caracteres\_especiales && !(tiene\_mayusculas && tiene\_minusculas))

{

return false;

}

if (tiene\_mayusculas && tiene\_minusculas)

{

return true;

}

if (tiene\_caracteres\_especiales)

{

return true;

}

//return true;

}

if (cant\_pares > 1)

{

return true;

}

return false;

}

static void Main(string[] args)

{

//string message = "bob";//false

//string message = "heh!";

//string message = "aA"; //true

// string message = "a0099A"; //true

//string message = "!a!"; // => true?

//string message = "A!"; //, true;

// string message = "Ada";//, true;

//string message = "Aadaa";//, true;

// string message = "aadaa";//, false;

//string message = "adda";//, true.

//string message = "aboooba";

string message = "988";

//string message = "!?~~";

//string message = "aaa";

// string message = "baoooab";

//string message = "z";

//string message = " Hey! Yehoo! 111";

// string message = " ";

//string message = "aabb";

// string message = "abcba";

// string message = "adda"; //, true.

//string message = "abcba"; //true

Console.WriteLine(hasPalindromicAnagram(message));

Console.ReadLine();

}

}

}