**Isograms**

3626689% of3,5721,388 of 26,848[chunjef](https://www.codewars.com/users/chunjef)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/isograms/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/isograms/solutions/csharp)

[Forks (9)](https://www.codewars.com/kata/isograms/forks/csharp)

[Discourse (83)](https://www.codewars.com/kata/isograms/discuss/csharp)

* Add to Collection
* |
* Share this kata:

An isogram is a word that has no repeating letters, consecutive or non-consecutive. Implement a function that determines whether a string that contains only letters is an isogram. Assume the empty string is an isogram. Ignore letter case.

isIsogram "Dermatoglyphics" == true

isIsogram "moose" == false

isIsogram "aba" == false

<https://www.codewars.com/kata/isograms/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static bool IsIsogram(string str)

{

// Code on you crazy diamond!

str = str.ToLower();

HashSet<char> hs = new HashSet<char>();

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

{

if (hs.Contains(str[i])) return false;

hs.Add(str[i]);

}

return true;

}

static void Main(string[] args)

{

}

}

}

//------------------- C++ ----------------------

#include <iostream>

#include <stdio.h>

#include <set>

using namespace std;

//std::tolower('T', std::locale());

char minuscula(char ch) {

return tolower(ch, std::locale());

}

bool is\_isogram(std::string str) {

string minusc = "";

for(int i =0; i<str.length(); i++) {

minusc += tolower(str[i]);

}

std::set<char> hs;

for(int i =0; i<minusc.length(); i++) {

if(hs.find(minusc[i]) != hs.end()) {

return false;

}

hs.insert(minusc[i]);

}

return true;

}

/\*

Assert::That(is\_isogram("Dermatoglyphics"), Equals(true));

Assert::That(is\_isogram("moose"), Equals(false));

Assert::That(is\_isogram("isIsogram"), Equals(false));

\*/

int main() {

cout << is\_isogram("Dermatoglyphics") << endl;

return 0;

}