Author

[peltorator](https://codefights.com/profile/peltorator)

https://codefights.com/img/coins_new.png3000

A *Palindrome* is a string which reads the same backward or forward. Find the length of the longest *palindrome* you can get from the given string s by removing some (possibly zero) characters.

**Example**

For s = "foundpalindrome", the output should be  
Palindrome(s) = 5.

One of the possible *palindromes* that can be constructed is "onino".  
There are other *palindromes* of length 5, but none of greater length.

* **[input] string s**

A string consisting of lowercase and uppercase English letters, digits and symbols.  
3 ≤ s.length ≤ 20.

* **[output] integer**

A positive integer, the longest *palindrome*that can be obtained.

<https://codefights.com/challenge/EGbN25HuH2PvaEYTS>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

static List<List<char>> Powerset(char[] palabra)

{

List<List<char>> ps = new List<List<char>>();

ps.Add(new List<char>()); // add the empty set

// for every item in the original list

foreach (char item in palabra)

{

List<List<char>> newPs = new List<List<char>>();

foreach (List<char> subset in ps)

{

// copy all of the current powerset's subsets

newPs.Add(subset);

// plus the subsets appended with the current item

List<char> newSubset = new List<char>(subset);

newSubset.Add(item);

newPs.Add(newSubset);

}

// powerset is now powerset of list.subList(0, list.indexOf(item)+1)

ps = newPs;

}

return ps;

}

static int Palindrome(string s)

{

List<List<char>> lista = Powerset(s.ToCharArray());

int maxPalin = 0;

foreach (List<char> l in lista)

{

string concat = "";

foreach (char ch in l)

{

//Console.Write(ch);

concat += ch;

}

//Console.WriteLine();

bool espalin = true;

int i = 0, j = concat.Length - 1;

while (i < j)

{

if (concat[i] != concat[j])

{

espalin = false;

break;

}

i++;

j--;

}

if (espalin)

{

maxPalin = Math.Max(maxPalin, concat.Length);

}

}

return maxPalin;

}

static void Main(string[] args)

{

string s = "foundpalindrome";

//string s = "found";

Console.WriteLine(Palindrome(s));

Console.ReadLine();

}

}

}