The fight on whether to store numbers starting with their most significant digit or their least significant digit has been going on for years. It even got a name and is called the [Endian War](https://en.wikipedia.org/wiki/Endianness) by some specialists.

Joe Stoy in his (excellent, by the way) book "Denotational Semantics", tells the following story about Alan Turing: "...One early British computer had numbers running from right to left (because the spot on an oscilloscope tube runs from left to right, but in serial logic the least significant digits are dealt with first). Turing used to mystify audiences at public lectures when, quite by accident, he would slip into this mode even for decimal arithmetic, and write things like 73+42=16...".

You are given an expression that was presumably written by Alan Turing. Return trueif it is a correct expression written in the little-endian decimal format, or return falseotherwise.

**Example**

* For expression = 73+42=16, the output should be  
  endianWar(expression) = true.

In the little-endian decimal format, the expression becomes 37 + 24 = 61, which is correct.

* For expression = "5+8=13", the output should be  
  endianWar(expression) = false.

In the little-endian decimal format, the result of the expression should be 31.

**Input/Output**

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

An expression in the format a+b=c, where a, b and c are decimal numbers (possibly with leading zeros) that are guaranteed to be smaller than 106.

The expression is guaranteed to be valid.

* **[output] boolean**

Return true if expression is valid in the little-endian decimal notation, otherwise return false.

<https://codefights.com/challenge/ChcFLSa3rfJsKNgkC>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static bool endianWar(string expression)

{

string[] s = expression.Split('+', '=');

char[] a = s[0].ToCharArray();

char[] b = s[1].ToCharArray();

char[] c = s[2].ToCharArray();

Array.Reverse(a);

Array.Reverse(b);

Array.Reverse(c);

if (int.Parse(new string(a)) + int.Parse(new string(b)) == int.Parse(new string(c)))

{

return true;

}

return false;

}

static void Main(string[] args)

{

string exp = "73+42=16";

Console.WriteLine( endianWar(exp));

Console.ReadLine();

}

}

}