**7 kyu**

**Scaling Squared Strings**

631192% of 26193 of1,440[g964](https://www.codewars.com/users/g964)

C#

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

Details

[Solutions](https://www.codewars.com/kata/scaling-squared-strings/solutions/csharp)

[Forks (1)](https://www.codewars.com/kata/scaling-squared-strings/forks/csharp)

[Discourse (35)](https://www.codewars.com/kata/scaling-squared-strings/discuss/csharp)

* Add to Collection
* |
* Share this kata:

You are given a string of n lines, each substring being n characters long. For example:

s = "abcd\nefgh\nijkl\nmnop"

We will study the "horizontal" and the "vertical" **scaling** of this square of strings.

A k-horizontal scaling of a string consists of replicating k times each character of the string (except '\n').

* Example: 2-horizontal scaling of s: => "aabbccdd\neeffgghh\niijjkkll\nmmnnoopp"

A v-vertical scaling of a string consists of replicating v times each part of the squared string.

* Example: 2-vertical scaling of s: => "abcd\nabcd\nefgh\nefgh\nijkl\nijkl\nmnop\nmnop"

Function scale(strng, k, v) will perform a k-horizontal scaling and a v-vertical scaling.

Example: a = "abcd\nefgh\nijkl\nmnop"

scale(a, 2, 3) --> "aabbccdd\naabbccdd\naabbccdd\neeffgghh\neeffgghh\neeffgghh\niijjkkll\niijjkkll\niijjkkll\nmmnnoopp\nmmnnoopp\nmmnnoopp"

Printed:

abcd -----> aabbccdd

efgh aabbccdd

ijkl aabbccdd

mnop eeffgghh

eeffgghh

eeffgghh

iijjkkll

iijjkkll

iijjkkll

mmnnoopp

mmnnoopp

mmnnoopp

#Task:

* Write function scale(strng, k, v) k and v will be positive integers. If strng == "" return "".

<https://www.codewars.com/kata/scaling-squared-strings/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static string Scale(string strng, int k, int n)

{

// your code

string fila = "";

string concat = "";

string res = "";

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

{

concat += strng[i];

if (strng[i] == '\n')

{

//fila = "";

fila = "";

for (int j = 0; j < concat.Length; j++)

{

string letra = concat[j].ToString();

if (letra != "\n")

{

for (int l = 0; l < k; l++)

{

fila += letra;

}

}

}

string filas\_concatenadas = "";

for (int j = 0; j < n; j++)

{

filas\_concatenadas += fila + "\n";

}

res += filas\_concatenadas ;

//res += "\n";

fila = "";

filas\_concatenadas = "";

concat = "";

}

}

if(concat.Length > 0)

{

for (int j = 0; j < concat.Length; j++)

{

string letra = concat[j].ToString();

if (letra != "\n")

{

for (int l = 0; l < k; l++)

{

fila += letra;

}

}

}

string filas\_concatenadas = "";

for (int j = 0; j < n; j++)

{

filas\_concatenadas += fila + "\n";

}

res += filas\_concatenadas;

}

return res.TrimEnd('\n');

}

static void Main(string[] args)

{

string a = "abcd\nefgh\nijkl\nmnop";

Console.WriteLine(Scale(a, 2, 3));

Console.ReadLine();

}

}

}