**8 kyu**

**Generate range of integers**

42594% of 358281 of2,201[XxxX88](https://www.codewars.com/users/XxxX88" \o "This kata's Sensei)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/generate-range-of-integers/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/generate-range-of-integers/solutions/csharp)

[Discourse (19)](https://www.codewars.com/kata/generate-range-of-integers/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Implement a function named generateRange(min, max, step), which takes three arguments and generates a range of integers from min to max, with the step. The first integer is the minimum value, the second is the maximum of the range and the third is the step. (min < max)

**Task**

Implement a function named

GenerateRange(2, 10, 2) == new int[]{ 2, 4, 6, 8, 10 }

GenerateRange(1, 10, 3) == new int[]{ 1, 4, 7, 10 }

, which takes three arguments and generates a range of integers from min to max, with given step. The first is minimum value, second is maximum of range and the third is step.

**Note**

* min < max
* step > 0

<https://www.codewars.com/kata/generate-range-of-integers/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static int[] GenerateRange(int min, int max, int step)

{

List<int> res = new List<int>();

for (int i = min; i <= max; i += step)

{

res.Add(i);

}

return res.ToArray();

}

static void Main(string[] args)

{

foreach(int item in GenerateRange(1,10,4))

{

Console.Write(item + " ");

}

Console.ReadLine();

}

}

}