**5 kyu**

**Pascal's Diagonals**

20690% of 5923 of228[stevehopkinson](https://www.codewars.com/users/stevehopkinson)

C#

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

Details

[Solutions](https://www.codewars.com/kata/pascals-diagonals/solutions/csharp)

[Discourse (36)](https://www.codewars.com/kata/pascals-diagonals/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Create a function that returns an array containing the first l digits from the nth diagonal of [Pascal's triangle](https://en.wikipedia.org/wiki/Pascal's_triangle" \t "_blank).

n = 0 should generate the first diagonal of the triangle (the 'ones'). The first number in each diagonal should be 1.

If l = 0, return an empty array. Assume that both n and l will be non-negative integers in all test cases.

[**https://www.codewars.com/kata/576b072359b1161a7b000a17/solutions/csharp**](https://www.codewars.com/kata/576b072359b1161a7b000a17/solutions/csharp)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Numerics;

namespace ConsoleApp1

{

class Program

{

public static BigInteger[] GenerateDiagonal(int n, int l)

{

if (l == 0) return new BigInteger[0];

int max = Math.Max(l, n);

BigInteger[,] matrix = new BigInteger[max,max];

for (int i = 0; i < max; i++)

{

matrix[i, 0] = 1;

matrix[0, i] = 1;

}

for (int i = 1; i < max; i++)

for (int j = 1; j < max; j++)

matrix[i, j] = matrix[i - 1, j] + matrix[i, j - 1];

//BigInteger[] ans = new BigInteger[l];

List<BigInteger> ans = new List<BigInteger>();

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

{

ans.Add( matrix[n,i]);

}

return ans.ToArray();

}

static void Main(string[] args)

{

BigInteger[] res = GenerateDiagonal(10,15);

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

{

Console.Write(res[i] + " ");

}

Console.ReadLine();

}

}

}

**----------------**

**using System.Numerics;**

**using System.Linq;**

**using System.Collections.Generic;**

**public class Kata**

**{**

**public static BigInteger[] GenerateDiagonal(int n, int l)**

**{**

**if (l == 0)**

**{**

**return new BigInteger[0];**

**}**

**var diagonal = new BigInteger[l];**

**diagonal[0] = BigInteger.One;**

**for (int i = 1; i <= l - 1; i++)**

**{**

**diagonal[i] = (diagonal[i - 1]\*(n + i)/i);**

**}**

**return diagonal;**

**}**

**}**