Reward: https://codefights.com/img/coins_new.png **3000**

For a given non negative integer n find the n-th row of Pascal's triangle. <http://en.wikipedia.org/wiki/Pascal%27s_triangle>. For example for the input n = 0 output should be [1], for input n = 3 output should be [1,3,3,1]

[Show Less](javascript:void(0);)

**Input (n)** → integer :

Non negative integer

**Output** → array.integer :

The n-th row of the Pascal's Triangle

<https://codefights.com/feed/CN6rCuF9zbDZCmZHG?fb_ref=Default>

**----------MI SOLUCION - ACEPTADO----------**

#include <iostream>

#include <vector>

#include <stdio.h>

std::vector<int> PascalRow(int n)

{

int tabla[n+1][n+1];

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

{

tabla[fila] [0] = 1;

}

for (int fila = 1; fila <= n; fila++)

{

tabla[fila][1] = fila;

}

for (int i = 2; i <= n; i++)

{

tabla[i][i] = 1;

}

for (int i = 3; i <= n; i++)

{

for (int j = 2; j < i ; j++)

{

tabla[i] [j] = tabla[i - 1] [j - 1] + tabla[i - 1][j];

}

}

std::vector<int> R;

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

{

R.push\_back(tabla[n][i]);

}

return R;

}

int main() {

std::vector<int> r = PascalRow(6);

for(int i = 0; i < r.size(); i++) {

printf("%d ", r[i]);

}

return 0;

}

static int[] Pascal\_triangle(int n)

{

int[,] tablero = new int[n+1, n+1];

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

{

tablero[i, 0] = 1;

tablero[i, i] = 1;

}

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

{

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

{

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

}

}

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

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

{

fila.Add( tablero[n-1, i]);

}

return fila.ToArray();

}