Author

[Chaikitty](https://codefights.com/profile/Chaikitty)

https://codefights.com/img/coins_new.png2000

Create an n × x rectangular matrix resembling[Pascal's Triangle](https://en.wikipedia.org/wiki/Pascal's_triangle). To make it more triangle-like, put 0between each pair of numbers in one row.

**Example:**

pascalMatrix(9) = [[0,0,0,0,1,0,0,0,0],

[0,0,0,1,0,1,0,0,0],

[0,0,1,0,2,0,1,0,0],

[0,1,0,3,0,3,0,1,0],

[1,0,4,0,6,0,4,0,1]]

* **[input] integer n**
  + An odd number, the number of elements in the last row.  
    0 < n ≤ 65.
* **[output] array.array.integer**
  + Pascal's Triangle.

<https://codefights.com/challenge/mXCP8e7BgNNqZ3amQ>

#include <iostream>

#include <stdio.h>

#include <ctype.h>

#include <vector>

#include <algorithm>

using namespace std;

std::vector< std::vector< int > > pascalMatrix(int n) {

//int[,] matriz = new int[(n / 2) + 1, n];

std::vector<std::vector<int > > matriz;

for (int i = 0; i < (n / 2) + 1; i++)

{

std::vector<int> fila;

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

{

fila.push\_back(0);

}

matriz.push\_back(fila);

}

for (int i = 0; i < (n / 2) + 1; i++)

{

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

{

if (j == (n / 2) - i || j == i + n / 2)

{

//matriz[i, j] = 1;

matriz[i][j]=1;

}

}

}

for (int i = 0; i < (n / 2) + 1; i++)

{

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

{

if (i - 1 >= 0 && j - 1 >= 0 && j + 1 < n)

{

if (matriz[i - 1][j - 1] != 0 && matriz[i - 1][j + 1] != 0)

{

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

}

}

}

}

return matriz;

}

int main() {

system("pause");

return 0;

}