Author

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

https://codefights.com/img/coins_new.png1000

Print out a pyramid with a symbol '\*'. The arguments count and direction define how the pyramid should be drawn.

The pyramid should contain count rows, where the ith row (1-based) consists of i'\*' characters.

If direction is true, the rows should be counted in top-down order, otherwise they should be counted from bottom to top.

**Example**

For count = 5 and direction = true, the output should be

printPyramid(count, direction) = ["\*",

"\*\*",

"\*\*\*",

"\*\*\*\*",

"\*\*\*\*\*"]

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer count**

The hight of the pyramid.

*Constraints:*  
3 ≤ count ≤ 50.

* **[input] boolean direction**

Direction in which the rows should be counted, true for top-down and falsefor bottom-up.

* **[output] array.string**

A pyramid built as described above.

<https://codefights.com/challenge/sMP85aZRQeFfjHDR4/main>

static string[] printPyramid(int count, bool direction)

{

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

if (direction)

{

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

{

string fila = "";

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

{

fila += "\*";

}

ans.Add(fila);

}

}

else

{

for (int i = count; i > 0; i--)

{

string fila = "";

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

{

fila += "\*";

}

ans.Add(fila);

}

}

return ans.ToArray();

}

-------------------------------otra solucion-----------------------------------------------------------------------

int b;

object printPyramid(int c, bool d)

{

return new int[c].Select(a => new string('\*', d ? ++b : c--));

}

-------------Solution by xeper-----------------------------

string[] printPyramid(int c, bool d)

{

var l = new string[c];

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

l[d ? i : c - i - 1] = new string('\*', i + 1);

return l;

}

string[] printPyramid(int c, bool d)

{

int a = d ? 1 : -1, b = d ? 1 : c, i = 0;

var s = new string[c];

for (; i < c; b += a)

s[i++] = new String('\*', b);

return s;

}

string[] printPyramid(int c, bool d)

{

var o = new string[c];

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

o[i] = new string('\*', i + 1);

return d ? o : o.Reverse().ToArray();

}

------solution by albinutte-------------

def printPyramid(count, direction):

res = []

for i in range(count):

res.append('\*' \* (i + 1))

if not direction:

return res[::-1]

return res