Functions and Fractals: Sierpinski triangles ★ **Points: 553.1800000000001 Rank: 1150**

Problem Submissions Leaderboard

RATE THIS CHALLENGE

★ ★ ★ ★ ★

```
_______________________________1_______________________________
______________________________111______________________________
_____________________________1___1_____________________________
____________________________111_111____________________________
___________________________1_______1___________________________
__________________________111_____111__________________________
_________________________1___1___1___1_________________________
________________________111_111_111_111________________________
_______________________1_______________1_______________________
______________________111_____________111______________________
_____________________1___1___________1___1_____________________
____________________111_111_________111_111____________________
___________________1_______1_______1_______1___________________
__________________111_____111_____111_____111__________________
_________________1___1___1___1___1___1___1___1_________________
________________111_111_111_111_111_111_111_111________________
_______________1_______________________________1_______________
______________111_____________________________111______________
_____________1___1___________________________1___1_____________
____________111_111_________________________111_111____________
___________1_______1_______________________1_______1___________
__________111_____111_____________________111_____111__________
_________1___1___1___1___________________1___1___1___1_________
________111_111_111_111_________________111_111_111_111________
_______1_______________1_______________1_______________1_______
______111_____________111_____________111_____________111______
_____1___1___________1___1___________1___1___________1___1_____
____111_111_________111_111_________111_111_________111_111____
___1_______1_______1_______1_______1_______1_______1_______1___
__111_____111_____111_____111_____111_____111_____111_____111__
_1___1___1___1___1___1___1___1___1___1___1___1___1___1___1___1_
111_111_111_111_111_111_111_111_111_111_111_111_111_111_111_111
```

Sierpinski Triangle

The Sierpinski Triangle is a pretty fractal which consistes of layers of self-similar triangles, nested inside each other. This challenge involves the construction of such triangles, in the form of ASCII Art. The restriction is, that you need to accomplish this with functional programming, and you cannot declare even local variables!

We have to deal with real world constraints, so we cannot keep repeating the pattern infinitely. So, we will provide you a number of iterations, and you need to generate the ASCII version of the Sierpinski Triangle for those many iterations (or, levels of recursion). A few samples are provided below.

Iteration #0

In the beginning, we simply print a triangle which points upwards. There are 32 rows and 63 columns in this matrix. The triangle is composed of underscores and ones as shown below.

```
_______________________________1_______________________________
______________________________111______________________________
_____________________________11111_____________________________
____________________________1111111____________________________
___________________________111111111___________________________
```

```
__________________________11111111111__________________________
_________________________1111111111111_________________________
________________________111111111111111________________________
_______________________11111111111111111_______________________
______________________1111111111111111111______________________
_____________________111111111111111111111_____________________
____________________11111111111111111111111____________________
___________________1111111111111111111111111___________________
__________________111111111111111111111111111__________________
_________________11111111111111111111111111111_________________
________________1111111111111111111111111111111________________
_______________111111111111111111111111111111111_______________
______________11111111111111111111111111111111111______________
_____________1111111111111111111111111111111111111_____________
____________111111111111111111111111111111111111111____________
___________11111111111111111111111111111111111111111___________
__________1111111111111111111111111111111111111111111__________
_________111111111111111111111111111111111111111111111_________
________11111111111111111111111111111111111111111111111________
_______1111111111111111111111111111111111111111111111111_______
______111111111111111111111111111111111111111111111111111______
_____11111111111111111111111111111111111111111111111111111_____
____1111111111111111111111111111111111111111111111111111111____
___111111111111111111111111111111111111111111111111111111111___
__11111111111111111111111111111111111111111111111111111111111__
_1111111111111111111111111111111111111111111111111111111111111_
111111111111111111111111111111111111111111111111111111111111111
```

Iteration #1

The "Fractalization" now begins. We create a new triangle, which points downwards, and its vertices co-incide with the midpoints of the outer, upward-pointing triangle. The ones are flipped into underscores. Note, that the original upward-pointing triangle has now been split into four segments: one downward-pointing triangle, filled with underscores - and three triangles which point upwards and are filled with ones.

```
_______________________________1_______________________________
______________________________111______________________________
_____________________________11111_____________________________
____________________________1111111____________________________
___________________________111111111___________________________
__________________________11111111111__________________________
_________________________1111111111111_________________________
________________________111111111111111________________________
_______________________11111111111111111_______________________
______________________1111111111111111111______________________
_____________________111111111111111111111_____________________
____________________11111111111111111111111____________________
___________________1111111111111111111111111___________________
__________________111111111111111111111111111__________________
_________________11111111111111111111111111111_________________
________________1111111111111111111111111111111________________
_______________1_______________________________1_______________
______________111_____________________________111______________
_____________11111___________________________11111_____________
____________1111111_________________________1111111____________
___________111111111_______________________111111111___________
__________11111111111_____________________11111111111__________
_________1111111111111___________________1111111111111_________
________111111111111111_________________111111111111111________
_______11111111111111111_______________11111111111111111_______
______1111111111111111111_____________1111111111111111111______
_____111111111111111111111___________111111111111111111111_____
____11111111111111111111111_________11111111111111111111111____
___1111111111111111111111111_______1111111111111111111111111___
__111111111111111111111111111_____111111111111111111111111111__
_11111111111111111111111111111___11111111111111111111111111111_
1111111111111111111111111111111_1111111111111111111111111111111
```

Iteration #2

We repeat the process on the three smaller upward-pointing triangles created at the end of Iteration #1. We create a downward pointing triangle inside each of those.

```
_______________________________1_______________________________
______________________________111______________________________
_____________________________11111_____________________________
____________________________1111111____________________________
___________________________111111111___________________________
__________________________11111111111__________________________
_________________________1111111111111_________________________
________________________111111111111111________________________
_______________________1_______________1_______________________
______________________111_____________111______________________
_____________________11111___________11111_____________________
____________________1111111_________1111111____________________
___________________111111111_______111111111___________________
__________________11111111111_____11111111111__________________
_________________1111111111111___1111111111111_________________
________________111111111111111_111111111111111________________
_______________1_______________________________1_______________
______________111_____________________________111______________
_____________11111___________________________11111_____________
____________1111111_________________________1111111____________
___________111111111_______________________111111111___________
__________11111111111_____________________11111111111__________
_________1111111111111___________________1111111111111_________
________111111111111111_________________111111111111111________
_______1_______________1_______________1_______________1_______
______111_____________111_____________111_____________111______
_____11111___________11111___________11111___________11111_____
____1111111_________1111111_________1111111_________1111111____
___111111111_______111111111_______111111111_______111111111___
__11111111111_____11111111111_____11111111111_____11111111111__
_1111111111111___1111111111111___1111111111111___1111111111111_
111111111111111_111111111111111_111111111111111_111111111111111
```

Input Format

One Integer N which is the Iteration Number for which you need to generate the Sierpinski triangle, in accordance with the triangles displayed above.

Generate the Nth triangle in the series shown above.

Input Constraint

N <= 5

Notes about the Triangle

As in the figures above, the canvas has a total of 32 rows and 63 columns. The outermost, upward-pointing triangle has a perpendicular height of 32 characters. The height of each of the downwards-pointing triangle, drawn in each iteration, is half of the upward-pointing one in which it is drawn.

Output Format

The Nth triangle of the series shown above. The output will consist of 32 rows and 63 columns, and will be composed of ones (1) and underscores as in the triangles above.

Change Theme Language Haskell

```
import Control.Monad
import Data.List (intersperse)

```

```
makeup :: (Int, Int) -> [[Char]]
makeup (h,w) = build (w `div` 2)
    where build 0 = [replicate w '1']
          build n = (replicate n '_' ++ replicate (w-2*n) '1' ++ replicate n '_') : build (n-1)

makeup' :: [[Char]] -> [[Char]]
makeup' xs = let w = length (head xs)
                 h = length xs
                 n1 = (w+1) `div` 2
                 up = (\x -> replicate n1 '_' ++ x ++ replicate n1 '_') <$> xs
                 down = (\x -> x ++ ('_':x)) <$> xs
              in up ++ down



breakup :: (Int, Int) -> (Int, Int)
breakup (h,w) = (h `div` 2, w `div` 2)

applyNTimes :: Int -> (a -> a) -> a -> a
applyNTimes 0 _ a = a
applyNTimes 1 f a = f a
applyNTimes n f a = f $ applyNTimes (n-1) f a

iterateN :: Int -> [[Char]]
```

Line: 35 Col: 1

Upload Code as File   □ Test against custom input   Run Code   Submit Code

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature