

# Functions and Fractals - Recursive Trees ★

Points: 553.180000000001 Rank: 1150

Problem	Submissions	Leaderboard
TE THIS CHALI	ENGE	
***		
	1 1 1 1 1 1 1 1	
		_1111111111111_
	11111	
	1_11_1 1 1	1_11_11_11_11_1
	111	1111111
	111	11111111
	<sup>1</sup> <sup>1</sup>	<sup>1</sup> <sup>1</sup> <sup>1</sup> <sup>1</sup> <sup>1</sup>
	1_1	1_11_11
	11	111111
	1	1
	11	1111
	1	111
	11	111
	11	11111
	1	1 1
	1	_11
	1_	11
	1 1	ll l
	1	l1
	1	l11111
	ـــــــــــــــــــــــــــــــــــــ	l 1
	1	l1
	1	11
	1	11
		_111
		111
		11111111
		1111
		111
		1111
		111
		11
		111
		111
		1
		1111

 $1\_\_\_\_$
 1
 1
1
 1
 1
 1
 1
 1
1
 <sup>1</sup>

# **Creating a Fractal Tree from Y-shaped branches**

This challenge involves the construction of trees, 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 Fractal Tree for only those many iterations (or, levels of recursion). A few samples are provided below.

## Iteration #1

In the beginning, we simply create a Y. There are 63 rows and 100 columns in the grid below. The triangle is composed of underscores and ones as shown below. The vertical segment and the slanting segments are both 16 characters in length.

1 1
1 1
111111
111
1 1
1111
11



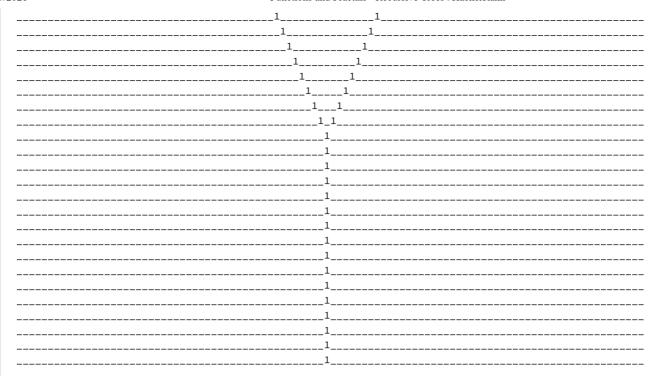
111	11
	11
	1
1_	1
1	11
	_1
	1
	1
	1
	 1
	1
	1
	1
	 1
	1
	1
	1
	1
	1

# Iteration #2

At the top of the left and right branches of the first Y, we now add a pair of Y-shapes, which are half the size of the original Y.

	11
	11
	_11
	11
	11
	11
	11
	1 1
	11
	11
	11
 1	11
 11	11
 11	11
	1
 11	1
 11	1
 11	_1





## **Input Format**

A single integer, N.

#### Constraints

N <= 5

And, you need to accomplish this without directly defining any local variables. For example, var and val have been blocked in Scala; def and defn are blocked in Clojure.

### **Output Format**

The N<sup>th</sup> iteration of the Fractal Tree, as shown above. It should be a matrix of 63 rows and 100 columns. (i.e. 6300 printable characters) It should be composed entirely of underscores and ones, in a manner similar to the examples provided. Do not include any extra leading or trailing spaces.

```
Change Theme Language Haskell
                                                                                      1
    import Control.Monad
 2
    treeN :: Int -> [[Char]]
 4
    treeN n = let branch = foldl (\xs i -> (replicate (n-i) '_' ++ ('1':replicate (2*i-1)
     '_') ++ "1" ++ replicate (n-i) '_'):xs) [] [1..n]
                   trunk = replicate n (replicate n '_' ++ "1" ++ replicate n '_')
 6
 7
                in branch ++ trunk
 8
 9
    treeN' :: Int -> Int -> ([[Char]],Int)
     treeN' n 0 = (treeN n, 0)
10
     treeN' n i = let n' = n `div` 2
11
                      (up,j) = treeN' n' (i-1)
12
                      up' = (\xs -> xs ++ replicate (2*n'-2*j-1) '_' ++ xs) <$> up
13
14
                      down = treeN n
                      m = (length (head up') - length (head down)) `div` 2
15
```

```
down' = (\xs -> replicate m '_' ++ xs ++ replicate m '_') <$> down
in (up' ++ down', m)
16
18
19
     main = do
20
        i <- read <$> getLine
         let res = (\x -> let n = (100 - length x) `div` 2 in replicate n '_' ++ x ++
21
     replicate n '_') <$> (fst $ treeN' 16 (i-1))
             res' = (\xspace x ->  let n = 100 - length x in x ++ replicate n '_') <\xspace > res
22
23
             len = 63 - length res'
             up = replicate len $ replicate 100 '_'
24
25
         forM (up ++ res') putStrLn
26
                                                                                           Line: 26 Col: 1
                                                                              Run Code
                                                                                            Submit Code
Test against custom input
```

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

