

Debugger Steps

Add a 'debugger' statement in your function
Call the function manually
At the terminal, run 'node inspect index.js'
To continue execution of the file, press 'c' then 'enter'
To launch a 'repl' session, type 'repl' then 'enter'
To exit the 'repl', press Control + C

Make an empty array 'words'

Split the input string by spaces to get an array

For each word in the array

Uppercase the first letter of the word

Join first letter with rest of the string

Push result into 'words' array

Join 'words' into a string and return it

Create 'result' which is the first character of the input string capitalized

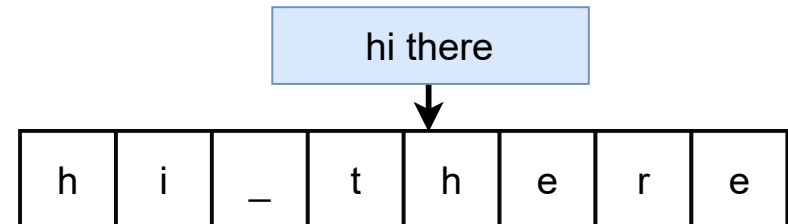
For each character in the string

IF the character to the left a space

Capitalize it and add it to 'result'

ELSE

Add it to 'result'



n = 3

column →

row ↓

	# c0	c1	
r0		r0 -	-
	# c0	#	-
r1			
	#	#	c2
			r2 #

From 0 to n (iterate through rows)

Create an empty string, 'stair'

From 0 to n (iterate through columns)

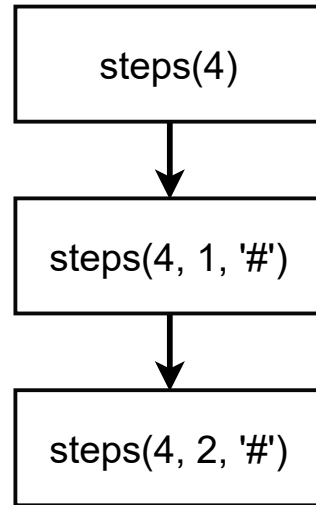
IF the current column is equal to or less than the current row

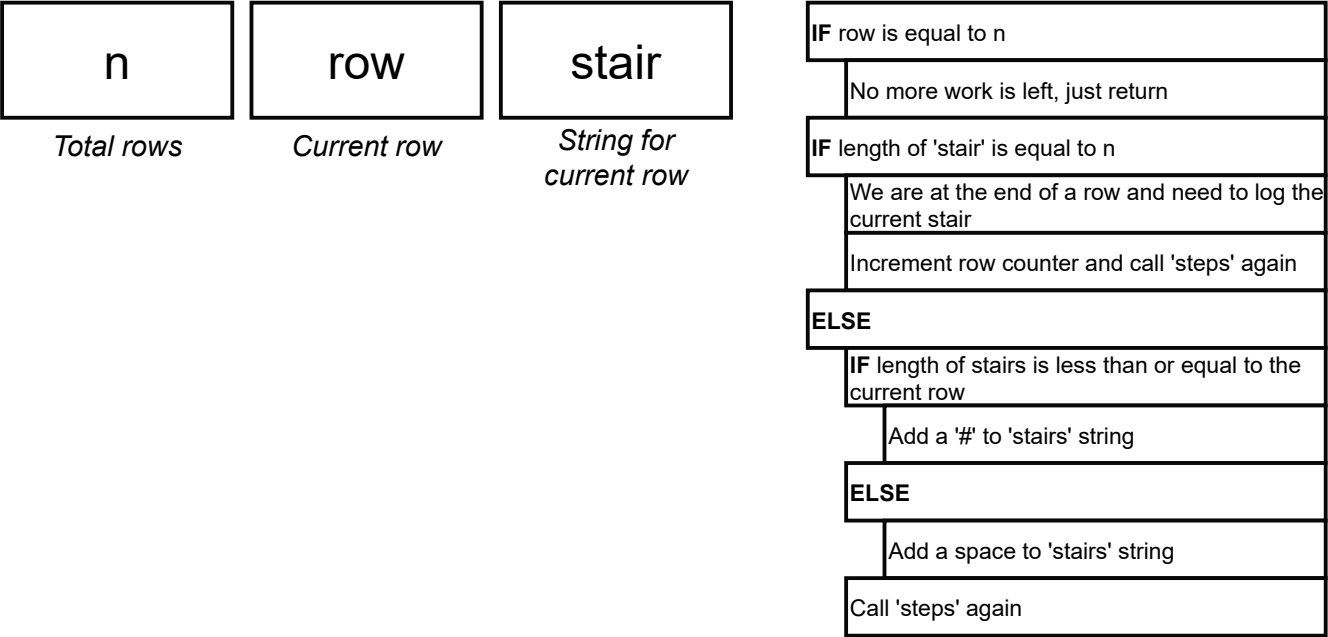
Add a '#' to 'stair'

ELSE

Add a space to 'stair'

Console log 'stair'





Recursion Tips

Figure out the bare minimum pieces of information to represent your problem

Give reasonable defaults to the bare minimum pieces of info

Check the base case. Is there any work left to do? If not, return

Do some work. Call your function again, making sure the arguments have changed in some fashion

n = 3

column

row

#	—	—
#	#	—
#	#	#

If (row === n) then we have hit the end of our problem

If the 'stair' string has a length === n then we are at the end of a row

If the *length* of the stair string is less than or equal to the row number we're working on, we add a '#', otherwise add a space

n = 3

column

row

—	—	#	—	—
—	#	#	#	—
#	#	#	#	#

From 0 to n (iterate through rows)

Create an empty string, 'level'

From 0 to ??? (columns)

IF the column should have a #

Add a '#' to 'level'

ELSE

Add a space to 'level'

Console log 'stair'

N = 2

—	#	—
#	#	#

3

N = 3

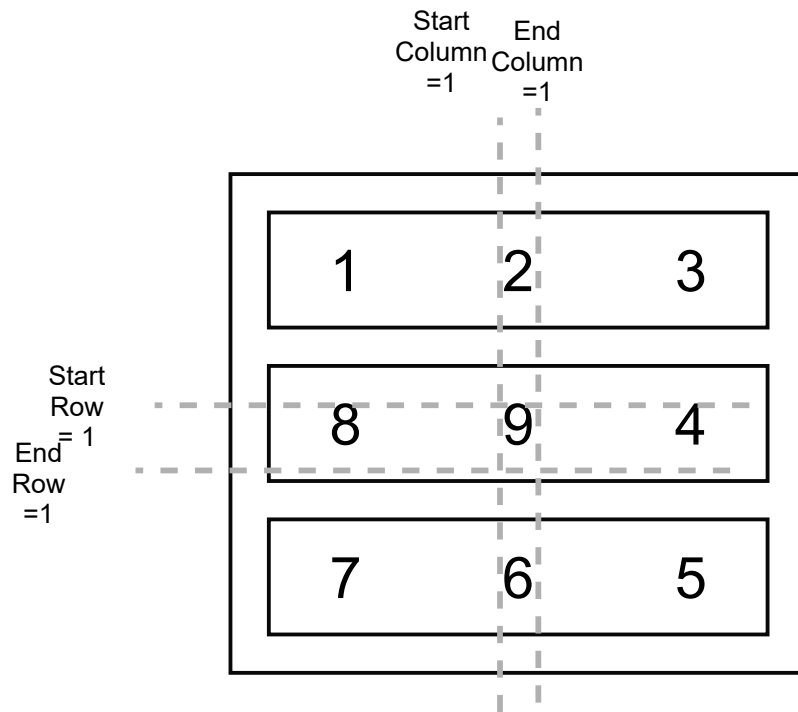
—	—	#	—	—
—	#	#	#	—
#	#	#	#	#

5

N = 4

—	—	—	#	—	—	—
—	—	#	#	#	—	—
—	#	#	#	#	#	—
#	#	#	#	#	#	#

1	2	3
8	9	4
7	6	5



Create empty array of arrays called 'results'
Create a counter variable, starting at 1
As long as (start column <= end column) AND (start row <= end row)
Loop from start column to end column
At results[start_row][i] assign counter variable
Increment counter
Increment start row
Loop from start row to end row
At results[i][end_column] assign counter variable
Increment counter
Decrement end column
...repeat for other two sides