Problem on Recursion

https://codingbat.com/prob/p194781

We have triangle made of blocks. The topmost row has 1 block, the next row down has 2 blocks, the next row has 3 blocks, and so on. Compute recursively (no loops or multiplication) the total number of blocks in such a triangle with the given number of rows.

triangle(0) → 0
triangle(1) → 1
triangle(2) → 3

Hint: Visually, how do the example input triangles look like?

Problem on Recursion

https://codingbat.com/prob/p173469

Given an array of ints, compute recursively if the array contains somewhere a value followed in the array by that value times 10. We'll use the convention of considering only the part of the array that begins at the given index. In this way, a recursive call can pass index+1 to move down the array. The initial call will pass in index as 0.

array220([1, 2, 20], 0) → true
array220([3, 30], 0) → true
array220([3], 0) → false

Try two versions:

```
boolean array220(int[] nums, int from)
boolean array220(int[] nums, int from, int to)
```

Ver. 1 assumes that the 'to' index denotes the end of the array.

Ver. 2 does not make such an assumption, though 'to' typically is the last index.

Hint: Max value of 'from' before an ArrayIndexOutOfBoundsException occurs?

Problem on Recursion

Return an array storing the first n numbers in an arithmetic sequence.

```
int[] arithmeticArray(int start, int diff, int n)
```

'start': the first term in an arithmetic sequence
'diff': the common difference between terms in an arithmetic sequence
'n': the first n numbers in an arithmetic sequence

e.g., arithmeticArray(2, 3, 5) returns an array {2, 5, 8, 11, 14}.

Hint: Let this method first create an array of the right size, then pass its **reference** to a recursive helper method, which modifies the array contents **recursively**.