Professional Coding Challenges

1. Write a function that returns all possible partitions of an array from left to right. With an *n* amount of elements in the input the returned array should have *n-1* subarrays. An empty array should return an empty array.

Example:
Input: [1, 5, 3, 2]
Output: [[[1], [5, 3, 2]], [[1, 5], [3, 2]], [[1, 5, 3], [2]]]

Input: [a, b, c]
Output: [[[a], [b, c]], [[a, b], [c]]]

2. Write a function to return a string in star shorthand. If a character is repeated *n* amount of times, convert the character *x* into *x*n*. Leave single characters alone. An empty string should return an empty string.

Example:
Input: "zmmxxxy"
Output: "zm*2x*3y"

Input: "aab"
Output: "a*2b"

Input: "qwer"
Output: "qwer"

3. Even if a number is not a palindrome, one of the number's descendants may be. A number's descendant can be found by adding each pair of adjacent digits together to make the digits of the next number. Write a function that returns *true* if the input number or any of its descendants down to 2 digits is a palindrome, return *false* otherwise. Your input will always have an even number of digits. If there is a single number trailing after addition leave it.

Example:
Input: 443244
Calcs: 858
Output: true

Input: 56
Calcs: 11
Output: true

Input: 12344321
Output: true

Input: 121113
Calcs: 324 -> 54
Output: false

4. An IPv4 formatted address contains 4 integers ranging from 0 to 255 separated by periods (.). Write a function that takes a string as input and returns *true* if the string is a valid IPv4 address. Return *false* otherwise.

Example:
Input: 123.123.123.123
Output: true

Input: 0.0.0.256
Output: false

5. Write a function that calculates the Golomb sequence to the *nth* term. The Golomb sequence is a non-decreasing sequence of integers where *a(n)* is the total amount of times that *n* appears in the sequence, beginning with *a(1) = 1*. The equation to find the next number in the sequence is as follows: *a(n + 1) = 1 + a(n + 1 - a(a(n)))*.

Example:
Input: 5
Output: [1, 2, 2, 3, 3]

Input: 15
Output: [1, 2, 2, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 6]