

1. Given a list of numbers and a number k , return whether any two numbers from the list add up to k .

For example, given `[10, 15, 3, 7]` and k of 17, return true since $10 + 7$ is 17.

2. Given an array of integers, return a new array such that each element at index i of the new array is the product of all the numbers in the original array except the one at i .

For example, if our input was `[1, 2, 3, 4, 5]`, the expected output would be `[120, 60, 40, 30, 24]`. If our input was `[3, 2, 1]`, the expected output would be `[2, 3, 6]`.

3. Given an array of integers, find the first missing positive integer in linear time and constant space. In other words, find the lowest positive integer that does not exist in the array. The array can contain duplicates and negative numbers as well.

For example, the input `[3, 4, -1, 1]` should give 2. The input `[1, 2, 0]` should give 3.

4. Given the mapping $a = 1, b = 2, \dots, z = 26$, and an encoded message, count the number of ways it can be decoded.

For example, the message '111' would give 3, since it could be decoded as 'aaa', 'ka', and 'ak'.

You can assume that the messages are decodable. For example, '001' is not allowed.

5. This problem was asked by Airbnb.

Given a list of integers, write a function that returns the largest sum of non-adjacent numbers. Numbers can be 0 or negative.

For example, `[2, 4, 6, 2, 5]` should return 13, since we pick 2, 6, and 5. `[5, 1, 1, 5]` should return 10, since we pick 5 and 5.

Follow-up: Can you do this in $O(N)$ time and constant space?

