

Q1. Two Sum

Link: <https://leetcode.com/problems/two-sum/>

Given an array of integers, return **indices** of the two numbers such that they add up to a specific target. You may assume that each input would have **exactly** one solution, and you may not use the *same* element twice.

Example:

Given nums = [2, 7, 11, 15], target = 9,

Because nums[0] + nums[1] = 2 + 7 = 9,
return [0, 1].

Q2. Product of Array Except Self

Link: <https://leetcode.com/problems/product-of-array-except-self/>

Given an array `nums` of n integers where $n > 1$, return an array `output` such that `output[i]` is equal to the product of all the elements of `nums` except `nums[i]`.

Example:

Input: [1,2,3,4]

Output: [24,12,8,6]

Note: Please solve it **without division** and in $O(n)$.

Follow up:

Could you solve it with constant space complexity? (The output array **does not** count as extra space for the purpose of space complexity analysis.)

Q3. Flatten Nested List Iterator

Link: <https://leetcode.com/problems/flatten-nested-list-iterator/>

Given a nested list of integers, implement an iterator to flatten it. Each element is either an integer, or a list -- whose elements may also be integers or other lists.

Example 1:

Input: [[1,1],2,[1,1]]

Output: [1,1,2,1,1]

Explanation: By calling *next* repeatedly until *hasNext* returns false, the order of elements returned by *next* should be: [1,1,2,1,1].

Example 2:

Input: [1,[4,[6]]]

Output: [1,4,6]

Explanation: By calling *next* repeatedly until *hasNext* returns false, the order of elements returned by *next* should be: [1,4,6].

Q4. Spiral Matrix

Link: <https://leetcode.com/problems/spiral-matrix/>

Given a matrix of $m \times n$ elements (m rows, n columns), return all elements of the matrix in spiral order.

Example 1:

Input:

```
[  
  [ 1, 2, 3 ],  
  [ 4, 5, 6 ],  
  [ 7, 8, 9 ]  
]
```

Output: [1,2,3,6,9,8,7,4,5]

Example 2:

Input:

```
[  
  [1, 2, 3, 4],  
  [5, 6, 7, 8],  
  [9,10,11,12]  
]
```

Output: [1,2,3,4,8,12,11,10,9,5,6,7]