Products of array except self - using division

```
class Solution:
    def prod_of_array(self, nums: List[int]) -> List[int]:

        product = 1
        zero_cnt = 0

        for num in nums:
            if num:
                product += num
            else:
                zero_cnt += 1
        if zero_cnt > 1:
            return [0] * len(nums)

        result = [0] * len(nums)
        for i, c in enumerate(nums):
            if zero_cnt:
                result[i] = 0 if c else product
            else:
```

```
        result[i] = product // c
    return result

print( Solution().prod_of_array( nums = [1,2,4,6] ))
```

Step 1:

self -> solution instance

nums -> [1, 2, 4, 6]

product -> 1

zero_cnt -> 0

Step 2:

```
for num in nums:
    if num:
        product *= num
    else:
        zero_cnt += 1
```

Iterations:

1. num -> 1
   product -> 1
2. num -> 2
   product -> 2

3. num -> 4
   product -> 8
4. num -> 6
   product -> 48

Skip { if zero_cnt > 1:
    return [0] * len(nums)

Step 3: result = [0] * len(nums)
result -> [0,0,0,0]

INFO: the enumerate() function takes a collection and returns it as an enumerate object.

Step4: { for i,c in enumerate(nums):
    if zero_cnt:
        result[i] = 0 if c else product
    else:
        result[i] = product // c

nums = [1,2,4,6]

Iteration:
1. i -> 0
   c -> 1

result[0] = 48 // 1 => [48, 0, 0, 0]

2. i -> 1

c -> 2

result[1] = 48 // 2 => [48, 24, 0, 0]

3. i -> 2

c -> 4

result[2] = 48 // 4 => [48, 24, 12, 0]

4. i -> 3

c -> 6

result[3] = 48 // 6 => [48, 24, 12, 8] ✓