1480. Running sum of 1d array:

Given an array `nums`. We define a running sum of an array as `runningSum[i] = sum(nums[0]…nums[i])`.

Return the running sum of `nums`.

Example 1:

```
Input: nums = [1,2,3,4]
Output: [1,3,6,10]
Explanation: Running sum is obtained as follows: [1, 1+2, 1
+2+3, 1+2+3+4].
```

Example 2:

```
Input: nums = [1,1,1,1,1]
Output: [1,2,3,4,5]
Explanation: Running sum is obtained as follows: [1, 1+1, 1
+1+1, 1+1+1+1, 1+1+1+1+1].
```

Example 3:

```
Input: nums = [3,1,2,10,1]
Output: [3,4,6,16,17]
```

Constraints:

- `1 <= nums.length <= 1000`
- `10^6 <= nums[i] <= 10^6`

Solution:

time = O(N) and space = O(N)

```
class Solution:
    def runningSum(self, nums: List[int]) -> List[int]:
        lst = []
        sum = 0
        for i in nums:
            sum = sum+i
            lst.append(sum)
        return lst
```

Optimal: time O(N) and space = O(1)

```
class Solution:
    def runningSum(self, nums: list[int]) -> list[int]:
        for i in range(1, len(nums)):
            nums[i] += nums[i - 1]
        return nums
```