**Bài 1.**

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

**Bài 2.**

Given the array nums consisting of 2n elements in the form [x1,x2,...,xn,y1,y2,...,yn].

*Return the array in the form* [x1,y1,x2,y2,...,xn,yn].

**Example 1:**

**Input:** nums = [2,5,1,3,4,7], n = 3

**Output:** [2,3,5,4,1,7]

**Explanation:** Since x1=2, x2=5, x3=1, y1=3, y2=4, y3=7 then the answer is [2,3,5,4,1,7].

**Example 2:**

**Input:** nums = [1,2,3,4,4,3,2,1], n = 4

**Output:** [1,4,2,3,3,2,4,1]

**Example 3:**

**Input:** nums = [1,1,2,2], n = 2

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

**Constraints:**

* 1 <= n <= 500
* nums.length == 2n
* 1 <= nums[i] <= 10^3