1929. Concatenation of Array

Given an integer array nums of length n, you want to create an array ans of length 2n where ans[i] == nums[i] and ans[i + n] == nums[i] for 0 <= i < n (**0-indexed**).

Specifically, ans is the **concatenation** of two nums arrays.

Return the array ans.

1470. Shuffle the Array

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].

1431. Kids With the Greatest Number of Candies

There are n kids with candies. You are given an integer array candies, where each candies[i] represents the number of candies the ith kid has, and an integer extraCandies, denoting the number of extra candies that you have.

Return a boolean array result of length n, where result[i] is true if, after giving the ith kid all the extraCandies, they will have the ***greatest*** number of candies among all the kids, or false otherwise.

Note that **multiple** kids can have the **greatest** number of candies.

1512. Number of Good Pairs

Given an array of integers nums, return the number of ***good pairs***.

A pair (i, j) is called good if nums[i] == nums[j] and i < j.

287. Find the Duplicate Number

Given an array of integers nums containing n + 1 integers where each integer is in the range [1, n] inclusive.

There is only **one repeated number** in nums, return this repeated number.

You must solve the problem **without** modifying the array nums and uses only constant extra space.

## 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.

Add to Array-Form of Integer

The **array-form** of an integer num is an array representing its digits in left to right order.

* For example, for num = 1321, the array form is [1,3,2,1].

Given num, the **array-form** of an integer, and an integer k, return *the****array-form****of the integer* num + k

Build Array from Permutation

Given a **zero-based permutation** nums (**0-indexed**), build an array ans of the **same length** where ans[i] = nums[nums[i]] for each 0 <= i < nums.length and return it.

A **zero-based permutation** nums is an array of **distinct** integers from 0 to nums.length - 1 (**inclusive**).