## 01) Find Pivot Index

Given an array of integers nums, calculate the **pivot index** of this array.

The **pivot index** is the index where the sum of all the numbers **strictly** to the left of the index is equal to the sum of all the numbers **strictly** to the index's right.

If the index is on the left edge of the array, then the left sum is 0 because there are no elements to the left. This also applies to the right edge of the array.

Return the **leftmost pivot index**. If no such index exists, return -1.

**Constraints:**

- 1 <= nums.length <= $10^4$
- -1000 <= nums[i] <= 1000

**Example 1:**

**Input: nums =** [1,7,3,6,5,6]

**Output:** 3

**Explanation:** The pivot index is 3.

Left sum = nums[0] + nums[1] + nums[2] = 1 + 7 + 3 = 11

Right sum = nums[4] + nums[5] = 5 + 6 = 11

**Example 2:**

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

**Output:** -1

**Explanation:** There is no index that satisfies the conditions in the problem statement

**Example 3:**

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

**Output:** 0

**Explanation:** The pivot index is 0.

Left sum = 0 (no elements to the left of index 0)

Right sum = nums[1] + nums[2] = 1 + -1 = 0

## 02) Group Anagrams

Given an array of strings strs, group the **anagrams** together. You can return the answer in any order.

**Constraints:**

- 1 <= strs.length <= $10^4$
- 0 <= strs[i].length <= 100
- strs[i] consists of lowercase English letters.

**Example 1:**

**Input: strs =** ["eat","tea","tan","ate","nat","bat"]

**Output:** [["bat"],["nat","tan"],["ate","eat","tea"]]

**Explanation:** There is no string in strs that can be rearranged to form "bat".

The strings "nat" and "tan" are anagrams as they can be rearranged to form each other.
The strings "ate", "eat", and "tea" are anagrams as they can be rearranged to form each other.

**Example 2:**

**Input: strs =** [""]

**Output:** [[""]]

**Example 3:**

**Input: strs =** ["a"]

**Output:** [["a"]]

## 03) Climbing Stairs

You are climbing a staircase. It takes n steps to reach the top.
Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

**Constraints:** 1 <= n <= 45

**Example 1:**

**Input: n =** 2

**Output:** 2

Explanation: There are two ways to climb to the top.

1. 1 step + 1 step
2. 2 steps

**Example 2:**

**Input: n =** 3

**Output:** 3

**Explanation:** There are three ways to climb to the top.

1. 1 step + 1 step + 1 step
2. 1 step + 2 steps
3. 2 steps + 1 step

## 04) Maximum Product Subarray

Given an integer array nums, find a subarray that has the largest product, and return the product.
The test cases are generated so that the answer will fit in a **32-bit integer**.

**Constraints:**

- 1 <= nums.length <= $2 * 10^4$
- -10 <= nums[i] <= 10
- The product of any subarray of nums is guaranteed to fit in a 32-bit integer.

**Example 1:**
**Input: nums =** [2,3,-2,4]
**Output:** 6
**Explanation:** [2,3] has the largest product 6.

**Example 2:**
**Input: nums =** [-2,0,-1]
**Output:** 0
**Explanation:** The result cannot be 2, because [-2,-1] is not a subarray.

## 05) Move All Zeroes to End

You are given an array arr[] of non-negative integers. Your task is to move all the zeros in the array to the right end while maintaining the **relative order of the non-zero elements**. The operation must be performed in place, meaning you should not use extra space for another array.

**Constraints:**

- $1 \leq arr.size() \leq 10^5$
- $0 \leq arr[i] \leq 10^5$

**Example 1:**
**Input: arr[] =** [1, 2, 0, 4, 3, 0, 5, 0]
**Output:** [1, 2, 4, 3, 5, 0, 0, 0]
**Explanation:** There are three 0s that are moved to the end.

**Example 2:**
**Input: arr[] =** [10, 20, 30]
**Output:** [10, 20, 30]
**Explanation:** No change in array as there are no 0s.