# 001-099

## 001. Valid Anagram (LC242)

Given two strings **s** and **t**, return true if **t** is an anagram of **s**, and false otherwise.

**Example 1:**
**Input:** s = "anagram", t = "nagaram"
Output: true

**Example 2:**
**Input:** s = "rat", t = "car"
Output: false

**Constraints:**
- $1 <= len(s), len(t) <= 5 * 10^4$
- s and t consist of lowercase English letters.

## 002. Group Anagrams (LC49)

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

**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"]]

**Constraints:**
- $1 <= len(strs) <= 10^4$
- $0 <= len(strs[i]) <= 100$
- strs[i] consists of lowercase English letters.

## 003. Contains Duplicate (LC217)

Given an integer array **nums**, return true if any value appears **at least twice** in the array, and return false if every element is distinct.

**Example 1:**
**Input:** nums = [1,2,3,1]
**Output:** true

**Explanation:**
The element 1 occurs at the indices 0 and 3.

**Example 2:**
**Input:** nums = [1,2,3,4]
**Output:** false

**Explanation:**
All elements are distinct.

**Example 3:**
**Input:** nums = [1,1,1,3,3,4,3,2,4,2]
**Output:** true

**Constraints:**
- $1 <= \text{len(nums)} <= 10^5$
- $-10^9 <= \text{nums[i]} <= 10^9$

## 004. Longest Substring Without Repeating Characters (LC3)

Given a string **s**, find the length of the longest substring without repeating characters.

**Example 1:**
**Input:** s = "abcabcbb"
**Output:** 3

**Explanation:** The answer is "abc", with the length of 3.

**Example 2:**
**Input:** s = "bbbbb"
**Output:** 1

**Explanation:** The answer is "b", with the length of 1.

**Example 3:**
**Input:** s = "pwwkew"
**Output:** 3

**Explanation:** The answer is "wke", with the length of 3.
Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

**Constraints:**
- $0 <= \text{len(s)} <= 5 * 10^4$
- s consists of English letters, digits, symbols and spaces.

## 005. Find the Index of the First Occurrence in a String (LC28)

Given two strings **needle** and **haystack**, return the index of the first occurrence of **needle** in **haystack**, or -1 if **needle** is not part of **haystack**.

**Example 1:**
**Input:** haystack = "sadbutsad", needle = "sad"
**Output:** 0

**Explanation:** "sad" occurs at index 0 and 6.
The first occurrence is at index 0, so we return 0.

**Example 2:**
**Input:** haystack = "leetcode", needle = "leeto"
**Output:** -1

**Explanation:** "leeto" did not occur in "leetcode", so we return -1.

**Constraints:**
- $1 <= len(haystack), len(needle) <= 10^4$
- haystack and needle consist of only lowercase English characters.

## 006. Move Zeroes (LC283)

Given an integer array **nums**, move all 0's to the end of it while maintaining the relative order of the non-zero elements.

**Note** that you must do this in-place without making a copy of the array.

**Example 1:**
**Input:** nums = [0,1,0,3,12]
**Output:** [1,3,12,0,0]

**Example 2:**
**Input:** nums = [0]
**Output:** [0]

**Constraints:**
- $1 <= len(nums) <= 10^4$
- $-2^{31} <= nums[i] <= 2^{31} - 1$

## 007. Missing Number (LC268)

Given an array **nums** containing **n** distinct numbers in the range [0, n], return the only number in the range that is missing from the array.

**Example 1:**
**Input:** nums = [3, 0, 1]
**Output:** 2

**Explanation:** n = 3 since there are 3 numbers, so all numbers are in the range [0,3]. 2 is the missing number in the range since it does not appear in **nums**.

**Example 2:**
**Input:** nums = [0, 1]
**Output:** 2
**Explanation:** n = 2 since there are 2 numbers, so all numbers are in the range [0,2]. 2 is the missing number in the range since it does not appear in **nums**.

**Example 3:**
**Input:** nums = [9, 6, 4, 2, 3, 5, 7, 0, 1]
**Output:** 8
**Explanation:** n = 9 since there are 9 numbers, so all numbers are in the range [0,9]. 8 is the missing number in the range since it does not appear in **nums**.

**Constraints:**
- n == len(nums)
- $1 <= n <= 10^4$
- $0 <= nums[i] <= n$
- All the numbers of nums are **unique**.

## 008. Two Maximum Numbers

Given an integer array **nums**, return 2(two) maximum numbers in this array.

**Example 1:**
**Input:** nums = [3, 0, 1]
**Output:** The first maximum: 3
        The second maximum: 1

**Example 2:**
**Input:** nums = [0, 1, 2, 3, 4, 5, 6]
**Output:** The first maximum: 6
         The second maximum: 5

**Example 3:**
**Input:** nums = [9, 6, 4, 2, 3, 5, 7, 0, 1]
**Output:** The first maximum: 9
        The second maximum: 7

**Constraints:**
- $2 <= len(nums) <= 10^4$