Q.1]

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:**a. 1 <= nums.length <= 10^4  
b. -2^31 <= nums[i] <= 2^31 - 1

**Solution :**

def moveZeroes(nums):

left = 0

right = 0

while right < len(nums):

if nums[right] != 0:

nums[left], nums[right] = nums[right], nums[left]

left += 1

right += 1

while left < len(nums):

nums[left] = 0

left += 1

return nums

**Solution :**

def firstUniqChar(s):

char\_count = {}

# Count the occurrences of each character

for char in s:

if char in char\_count:

char\_count[char] += 1

else:

char\_count[char] = 1

# Find the first character with count 1 and return its index

for i, char in enumerate(s):

if char\_count[char] == 1:

return i

# If no unique character found, return -1

return -1