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]

class Solution {

    public void moveZeroes(int[] nums) {

       int size = nums.length;

       if(size==0 || size==1)

       {

           return;

       }

       int a =0,b=0;

       while(a<size)

       {

           if(nums[a]!=0)

           {

               int temp = nums[a];

               nums[a]=nums[b];

               nums[b]=temp;

               a++;

               b++;

           }

           else

           {

               a++;

           }

       }

    }

}

Given a string s, *find the first non-repeating character in it and return its index*. If it does not exist, return -1.

**Example 1:**

**Input:** s = "leetcode"

**Output:** 0

class Solution {

    public int firstUniqChar(String s) {

        int len = s.length();

        int i=0;

        while(i<len)

        {

            char c = s.charAt(i);

            int fi = s.indexOf(c);

            int li = s.lastIndexOf(c);

            if(fi==li)

            {

                return fi;

            }

            i++;

        }

        return -1;

    }

}