QUE 1:
Remove Duplicates from Sorted Array

Input: nums = [1,1,2]
Output: 2, nums = [1,2,_]
Explanation: Your function should return k = 2, with the first two elements of nums being 1 and 2 respectively.
It does not matter what you leave beyond the returned k (hence they are underscores).

```
def remove_duplicate(arr):
    ht={}
    for num in arr:
        if num not in ht:
            ht[num]=arr[num]
        else:
            num+=1
    return ht[num]
print(remove_duplicate([1,1,2]))
```

```
def remove_duplicate(arr):
    unique_nums=[]
    seen=set()

    for num in arr:
        if num not in seen:
            unique_nums.append(num)
        seen.add(num)
    return unique_nums
print(remove_duplicate([1,2,2]))
```

what we have do in the remove duplicate
unique_nums=[] #store the array element
seen=set() # set is the store the unique element
for num in arr: # the given num in the arr search
if num not in seen # if not the in the seen
unique_nums.append(num) #the unique number is append in array

#remove the element second code
Input: nums = [3,2,2,3], val = 3
Output: 2, nums = [2,2,_,_]
Explanation: Your function should return k = 2, with the first two elements of nums being 2.
It does not matter what you leave beyond the returned k (hence they are underscores).

QUE 2:

#code 2 done:

```
def remove_element(arr , value):
    counter=0
    for i in range(len(arr)):
        if arr[i]==value:
            continue
        counter+=1
    return counter
print(remove_element([2,1] , 2))
```

QUE 3:

#quetion 3 is done:

```
def valid_palimdrome(str1):
    if str1==" ":return True
    left=0
    right=len(str1)-1
    while(left<right):
        if str1[left]!=str1[right]:
            return False
        left+=1
        right-=1
    return True
print(valid_palimdrome("ollo"))
```

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.

QUE 4:

#forth one is done

```
def first_index(str1 ,str2):
        if str2 in str1:
            return str1.index(str2)
        return -1
print(first_index("butsad" ,"sad"))
```

what is done in this

index fuction will return the fisrt index of that str2 in the str1
means index fuction uska first index return karega

QUE 5:

que 5 merge two sorted array:
Example 1:

Input: nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3
Output: [1,2,2,3,5,6]
Explanation: The arrays we are merging are [1,2,3] and [2,5,6].
The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from nums1.

```
def merge(nums1, m, nums2, n):
    i = m - 1
    j = n - 1
    k = m + n - 1

    while j >= 0:
        if i >= 0 and nums1[i] > nums2[j]:
            nums1[k] = nums1[i]
            i -= 1
        else:
            nums1[k] = nums2[j]
            j -= 1
        k -= 1

nums1 = [1, 3, 9, 0, 0, 0]
m = 3
nums2 = [2, 5, 6]
n = 3

merge(nums1, m, nums2, n)
print(nums1)


#que 6: DSA :
#MOVE ZEROS

def move_zeros(arr):
    ptr1=0
    for ptr2 in range(len(arr)):
        if arr[ptr2]!=0 and arr[ptr1]==0:
```

```
            arr[ptr1] ,arr[ptr2]=arr[ptr2] ,arr[ptr1]

        if arr[ptr1]!=0:
            ptr1+=1
    return arr
print(move_zeros([0,3,4,0,5,3,2,1]))
```

soln approch : simple two pointer tequenic simple used loop for the second pointer and first pointer at fisrt index pointing and second also intally at 0 th condition arr[ptr2]!=0 and arr[ptr1]==0 swaping the ptr1 and ptr2 and also if arr[ptr1]!=0 increment by one

"""

QUE 7:

```
#dsa quetion 7
def reverse_string(arr):
    left=0
    right=len(arr)-1
    while(left<right):
        arr[left] ,arr[right]=arr[right],arr[left]
        left+=1
        right-=1
    return arr
print(reverse_string(["h","e","l","l","o"]))
```

QUE 8:

```
#dsa quetion 8
class Solution:
    def getCommon(self, nums1, nums2):
        i, j = 0, 0

        while i < len(nums1) and j < len(nums2):
            if nums1[i] == nums2[j]:
                return nums1[i]
            elif nums1[i] < nums2[j]:
                i += 1
            else:
                j += 1

        return -1
```