63.**Given an integer array nums and an integer val, remove all occurrences of val in nums in-place. The relative order of the elements may be changed.**

Since it is impossible to change the length of the array in some languages, you must instead have the result be placed in the first part of the array nums. More formally, if there are k elements after removing the duplicates, then the first k elements of nums should hold the final result. It does not matter what you leave beyond the first k elements.

Return k *after placing the final result in the first* k *slots of* nums.

Do not allocate extra space for another array. You must do this by modifying the input array in-place with O(1) extra memory.

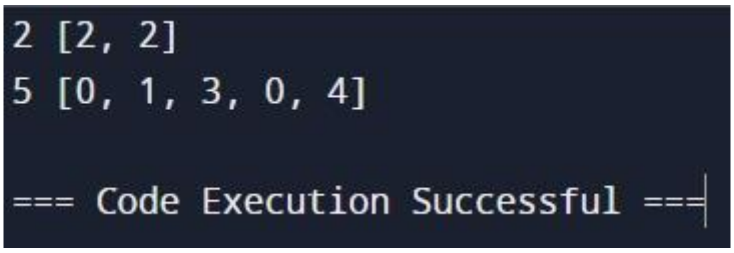
PROGRAM:-

```
def removeElement(nums, val):
    k = 0  # Initialize the index for placing non-val elements
    for i in range(len(nums)):
        if nums[i] != val:
            nums[k] = nums[i]
            k += 1
    return k

# Example usage:
nums1 = [3, 2, 2, 3]
val1 = 3
k1 = removeElement(nums1, val1)
print(k1, nums1[:k1])  # Output: 2, [2, 2]

nums2 = [0, 1, 2, 2, 3, 0, 4, 2]
val2 = 2
k2 = removeElement(nums2, val2)
print(k2, nums2[:k2])  # Output: 5, [0, 1, 3, 0, 4]
```

OUTPUT:-



```
2 [2, 2]
5 [0, 1, 3, 0, 4]

=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(n)