63. <mark>1. Remove Element</mark>

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.

**Custom Judge:**

**The judge will test your solution with the following code:**

**int[] nums = [...]; // Input array**

**int val = ...; // Value to remove**

**int[] expectedNums = [...]; // The expected answer with correct length.**

**// It is sorted with no values equaling val.**

**int k = removeElement(nums, val); // Calls your implementation**

**assert k == expectedNums.length;**

**sort(nums, 0, k); // Sort the first k elements of nums**

**for (int i = 0; i < actualLength; i++) {**

**assert nums[i] == expectedNums[i];**

**}**

If all assertions pass, then your solution will be accepted.

**Code:**

**def removeElement(nums,val):**

**k=0**

**for i in range(len(nums)):**

**if nums[i]!=val:**

```python
        nums[k]=nums[i]
        k += 1
    return k
nums=[3,2,2,3]
val=3
k=removeElement(nums,val)
print(f'Array after removing {val}:',nums[:k])
print('New length of the array:',k)
```

output:

```
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Desktop/daa.py
Array after removing 3: [2, 2]
New length of the array: 2
PS C:\Users\karth>
```

Time complexity:

F(n)=o(n)