

63) Remove Element

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.

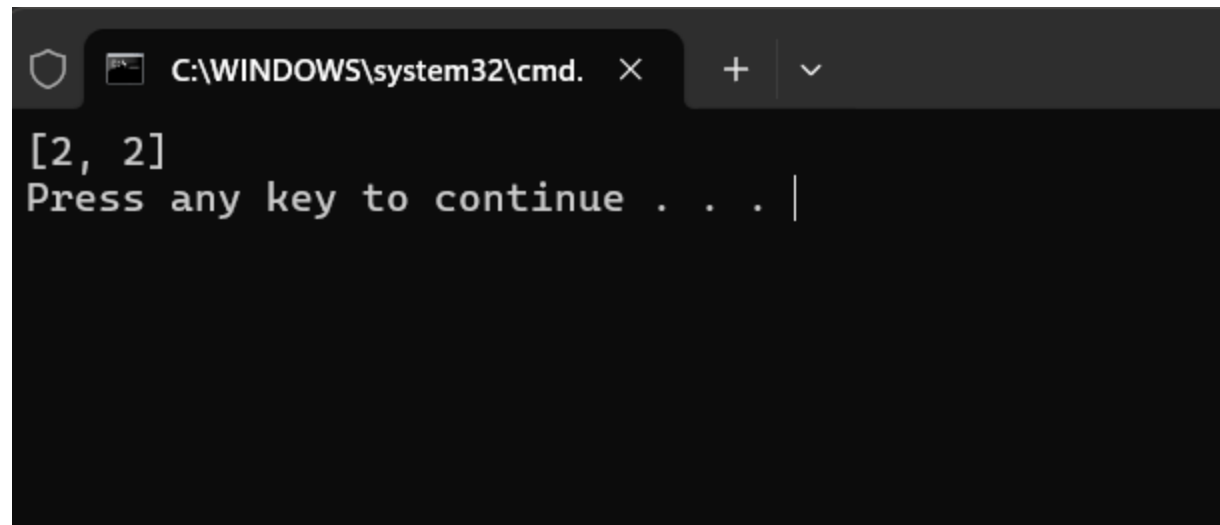
Return `k` after placing the final result in the first `k` slots of `nums`.

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

CODE:

```
def removes(a,t):
    b=[]
    for i in range(len(a)):
        if t!=a[i]:
            b.append(a[i])
    return b
a=[3,2,2,3]
b=3
print(removes(a,b))
```

OUTPUT:

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.' and standard window controls. The command prompt displays the output '[2, 2]' on the first line and 'Press any key to continue . . . |' on the second line, with a cursor at the end of the second line.

TIME COMPLEXITY : $O(1)$