**REMOVE ELEMENT**

Given an integer array nums and an integer val, remove all occurrences of val in nums [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm). The order of the elements may be changed. Then return *the number of elements in*nums*which are not equal to*val.

Consider the number of elements in nums which are not equal to val be k, to get accepted, you need to do the following things:

* Change the array nums such that the first k elements of nums contain the elements which are not equal to val. The remaining elements of nums are not important as well as the size of nums.
* Return k.

**CODE**

class Solution {

public:

    int removeElement(vector<int>& nums, int val)

    {

        int k = 0; // Pointer to track the position of the next non-val element

        for (int i = 0; i < nums.size(); i++)

        {

            if (nums[i] != val)

            {

                nums[k] = nums[i]; // Move non-val element to the front

                k++;

            }

        }

        return k;

    }

};

**SAMPLE OUTPUT**

**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).