

Job Interview Questions: Array

Question1: Assume that you are given an integer array sorted in non-decreasing order, write a function to remove the duplicates in this array such that each unique element appears only once (The relative order of the elements should be kept the same). The function should return k (the number of elements after removing the duplicate) as its return. Please Mention the computational Complexity of your solution

Note: This should be an **in-place** solution, meaning that you can not use any extra memory and you should write to the same input array, so the space complexity must be $O(1)$.

```
def remove_duplicates_from__sorted_array(nums):
```

```
    ...
```

Example:

```
nums = [0,0,1,1,1,2,2,3,3,4,5,5]
remove_duplicates_from__sorted_array(nums)
```

Output: 6, nums = [0,1,2,3,4,5,_,_,_,_,_,_]

The values after the first k element do not matter, in fact the nums array will be as:

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

Question2:

A palindrome is a word or phrase that reads the same backwards as forwards, for example, radar, level, rotor, kayak, reviver, racecar, madam, and refer. Implement a function `is_palindrome(x)` using a stack that returns true if a string is palindrome and false otherwise (assume you have access to a stack implementation).

```
# Driver Code
```

```
if __name__ == "__main__":  
  
    string = "madam"  
  
    if isPalindrome(string):  
        print("It is a palindrome!")  
    else:  
        print("No, It is not a palindrome!")
```

Will print: It is a palindrome!

Question 3:

Implement a queue (enqueue/dequeue function) using (only) two stacks.

Solution:

```
void enqueue(x) {  
    push (S1,x);  
}  
  
item dequeue(){  
  
    if(stack-empty(S2)) then  
        if(stack-empty(S1)) then {  
            print("Q is empty");  
            return;  
        }  
        else while (!(stack-empty(S1))){  
            x=pop(S1);  
            push(S2,x);  
        }  
    return pop(S2);  
}
```

