**NAME :** FELIX OUMA

**REG NO :** SCT212-0169/2022

**UNIT :** ICS 2105 DATA STRUCTURES AND ALGORITHMS

Assignment Arrays

Question one: **Remove duplicates from sorted arrays**

1. Initialize a variable i to 0. This variable will be used to keep track of the position of the last non-duplicate element in the array.
2. Iterate over the array from the second element (index 1) to the end. For each element, check if it is equal to the element at index i. If it is not, increment i and replace the element at index i with the current element.
3. After the loop, the length of the array without duplicates is i + 1.

Here is the Python code for the above steps:

def removeDuplicates(nums):

if not nums:

return 0

i = 0

for j in range(1, len(nums)):

if nums[j] != nums[i]:

i += 1

nums[i] = nums[j]

return i + 1

Question two : **Rotate arrays**

Given an array, rotate the array to the right by k steps, where k is non-negative.

1. Normalize k by taking its modulus with the length of the array. This is because rotating the array by its length or multiples of its length results in the same array.
2. Reverse the entire array.
3. Reverse the first k elements.
4. Reverse the remaining elements.

Here is the Python code for the above steps:

def rotate(nums, k):

k %= len(nums)

nums.reverse()

nums[:k] = reversed(nums[:k])

nums[k:] = reversed(nums[k:])

Question three : **Contains duplicates**

Given an array of integers, find if the array contains any duplicates. Your function should return true if any value appears at least twice in the array, and it should return false if every element is distinct.

1. Initialize an empty set.
2. Iterate over the array. For each element, check if it is in the set. If it is, return True. Otherwise, add it to the set.
3. If the loop completes without finding a duplicate, return False.

Here is the Python code for the above steps:

def containsDuplicate(nums):

seen = set()

for num in nums:

if num in seen:

return True

seen.add(num)

return False

Question four : **Single number**

Given a non-empty array of integers nums, every element appears twice except for one. Find that single one.

1. Initialize a variable single to 0.
2. Iterate over the array. For each element, take the bitwise XOR of single and the element, and assign the result back to single.
3. After the loop, single is the element that appears only once.

Here is the Python code for the above steps:

def singleNumber(nums):

single = 0

for num in nums:

single ^= num

return single

The bitwise XOR operation has the property that a XOR a = 0 and 0 XOR a = a. Therefore, all elements that appear twice will be cancelled out, leaving only the element that appears once.