DSA SERIES

- Learn Coding

Questions to be Covered today

1. **Two Sum**
2. **Remove Duplicates from Sorted Array**
3. **Remove Element**
4. **Plus One**

LETS START TODAY'S LECTURE

LEETCODE - 1

Two sum

Given an array of integers nums and an integer target, return *indices of the two numbers such that they add up to target*.
You may assume that each input would have *exactly* one solution, and you may not use the *same* element twice.
You can return the answer in any order.

Example 1:
Input: nums = [2,7,11,15],
target = 9 Output: [0,1]

Explanation:
Because nums[0] + nums[1] == 9, we return [0, 1].

Example 2:
Input: nums = [3,2,4],
target = 6
Output: [1,2]

Example 3:
Input: nums = [3,3],
target = 6 Output: [0,1]

Code :

```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        int n = nums.size();
        vector<int> ans;
        for (int i = 0; i < n - 1; i++) {
            int val = target - nums[i];
            for (int j = i + 1; j < n; j++) {

                if (val == nums[j]) {
                    ans.push_back(i);
                    ans.push_back(j);
                }
            }
        }
        return ans;
    }
};
```

LEETCODE - 26

<u>Remove duplicates from sorted array</u>

Given an integer array nums sorted in **non-decreasing order**,
remove the duplicates **in-place** such that each unique element appears only **once**.
The **relative order** of the elements should be kept the **same**.
Then return *the number of unique elements in* nums.

Consider the number of unique elements of nums to 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 unique elements in the order they were present in nums initially.
  The remaining elements of nums are not important as well as the size of nums.

- Return k.

Example 1:

Input: nums = [1,1,2]

Output: 2, nums = [1,2,_]

Explanation:

Your function should return k = 2,
with the first two elements of nums being 1 and 2 respectively
.
It does not matter what you leave beyond the returned k (hence they are underscores).

Code :

```
class Solution {

public:
    int removeDuplicates(vector<int>& nums) {
        int i = 0;
        int j = 1;

        int n = nums.size();
        while( j<n ){

            if(nums[i] != nums[j]){
                i++;
                nums[i] = nums[j];
            }
            j++;
        }
        return i+1;
    }
};
```

LEETCODE - 27

Remove Element

Given an integer array nums and an integer val, remove all occurrences of val in nums **in-place**.
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:

- Return k.

- 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.

Example 1:

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).

Code :

```
class Solution {
public:
    int removeElement(vector<int>& nums, int val) {

        int n = nums.size();

        int count = 0;

        for (int i = 0; i < n; i++) {
            if (nums[i] == val) {

                continue;
            }
            nums[count] = nums[i];
            count++;
        }

        return count;

    }
};
```

LEETCODE - 66

Plus One

You are given a **large integer** represented as an integer array digits,
where each digits[i] is the ith digit of the integer.

The digits are ordered from most significant to least significant in left-to-right order.
The large integer does not contain any leading 0’s.

Increment the large integer by one and return *the resulting array of digits*.

Input: digits = [1,2,3]

Output: [1,2,4]

Explanation:

The array represents the integer 123.

Incrementing by one gives 123 + 1 = 124.

Thus, the result should be [1,2,4].

Code :

```
class Solution {
public:
    vector<int> plusOne(vector<int>& digits) {
        int n = digits.size();

        for(int i = n-1;i>=0;i--){

            if(digits[i] < 9){
                digits[i]++;
                return digits;
            }
            digits[i]=0;
        }

        digits.insert(digits.begin(),1);
        return digits;
    }
};
```

Learn coding

Thank You