Q1. Two sum

def twoSum(self, nums: List[int], target: int) -> List[int]:

        num\_dict = {}

        for i, num in enumerate(nums):

            rem = target - num

            if rem in num\_dict:

                return [num\_dict[rem], i]

            num\_dict[num] = i

        return []

Q2. Remove Element

def removeElement(self, nums: List[int], val: int) -> int:

        count = 0

        for i in range(len(nums)):

            if nums[i] != val:

                nums[count] = nums[i]

                count += 1

        return count

Q3.  binary search

def searchInsert(self, nums: List[int], target: int) -> int:

        for i in range(len(nums)):

            if nums[i]==target:

                return i

            elif nums[i]>target:

                return i

            else:

                j=len(nums)

        return j

Q4 : Plus one

def plusOne(self, digits: List[int]) -> List[int]:

        string=''

        lst=[]

        for i in digits:

            string+=str(i)

        ans=int(string)

        ans=ans+1

        ans=str(ans)

        for i in ans:

            lst.append(int(i))

        return lst

q5 . Merge sorted array

def merge(self, nums1: List[int], m: int, nums2: List[int], n: int) -> None:

        last=m+n-1

        while(m>0 and n>0):

            if(nums1[m-1]>nums2[n-1]):

                nums1[last]=nums1[m-1]

                m-=1

            else:

                nums1[last]=nums2[n-1]

                n-=1

            last-=1

        while(n>0):

            nums1[last]=nums2[n-1]

            n-=1

            last-=1

q6. duplicate

def containsDuplicate(self, nums: List[int]) -> bool:

        return len(set(nums))!=len(nums)

q7. move zeros

def moveZeroes(self, nums: List[int]) -> None:

        count = 0

        for i in range(len(nums)):

            if nums[i] != 0:

                nums[count] = nums[i]

                count+=1

        while count<len(nums):

            nums[count] = 0

            count += 1

q8 . set mismatch

def findErrorNums(self, nums: List[int]) -> List[int]:

        n=len(nums)

        Total=(n\*(n+1))//2

        actual=sum(set(nums))

        lst=[]

        my\_dict={}

        rep=0

        miss=0

        for i ,num in enumerate(nums):

            if (num in my\_dict):

                rep=num

            else:

                my\_dict[num]=1

        ans=Total-actual

        print(Total,actual,ans)

        lst.append(rep)

        lst.append(ans)

        return lst