

# PPT Assignment-1 Questions

Q1.

Input: nums = [2,7,11,15], target = 9

Output0 [0,1]

Ans.

```
1 # Input: nums = [2,7,11,15], target = 9
2 # Output0 [0,1]
3
4 nums = [2,7,11,15]
5 target = 9
6 for i in range (len(nums)):
7     for j in range (i+1, len(nums)):
8         if (nums[i]+nums[j] == target):
9             print(i,j)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
...
0 1
```

 **Q2.** 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:

- 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.

**Example :**

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)

```

8  def my_func(nums, val):
9      a=[]
10     k=0
11     count=0
12     for i in range (len(nums)):
13         if(nums[i]!=val):
14             a.append(nums[i])
15             k=k+1
16         else:
17             count+=1
18         for i in range (count):
19             a.append('_')
20     print("Output:", k, "\n", "nums=",a)
21
22
23 nums = [3,2,2,3]
24 val = 3
25
26 my_func(nums, val)

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Output: 2  
 nums= [2, 2, '\_', '\_']  
 >>> []

**💡 Q3.** Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.  
 You must write an algorithm with  $O(\log n)$  runtime complexity.

**Example 1:**

Input: nums = [1,3,5,6], target = 5

Output: 2

Sol: when time complexity is :  $O(n)$

```
1 # Input: nums = [1,3,5,6], target = 5
2 # Output: 2
3
4 nums = [1,3,5,6]
5 target = 5
6 j=0
7
8 for i in range (len(nums)):
9     if(nums[i]<=target):
10         j+=1
11         if(nums[i] == target):
12             print("Output: ",i)
13             break
14     else:
15         print("Target not found, it should be at the position:",j)
16         break
17
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
...
else:
    print("Target not found, it should be at the position:",j)
...
...
Output: 2
```

Sol: when time complexity is : O(log n)

```

3
4     nums = [1, 3, 5, 6]
5     target = 5
6     left = 0
7     right = len(nums) - 1
8     k=0
9
10    while left <= right:
11        mid = (left + right) // 2
12
13        if nums[mid] == target:
14            print("Output:", mid)
15            k=1
16            break
17        elif nums[mid] < target:
18            left = mid + 1
19        else:
20            right = mid - 1
21    if k==0:
22        print("Target not found, it should be at the position:", left)
23
24

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

...  
Output: 2



**Q4.** 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.

**Example 1:**

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].

```

28
29  def my_func(a):
30      print(a)
31      arr_new = int(''.join(map(str, a)))
32      arr_new += 1
33      a = [int(digit) for digit in str(arr_new)]
34      print(a)
35
36  a = list(map(int, input("Enter an array of integers (space-separated): ").split()))
37  my_func(a)
38

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Python - abhishekmathur + ▾

```

>>> a = list(map(int, input("Enter an array of integers (space-separated): ").split()))
Enter an array of integers (space-separated): 1 6 8 9
>>> my_func(a)
[1, 6, 8, 9]
[1, 6, 9, 0]
>>>

```



**Q5.** You are given two integer arrays `nums1` and `nums2`, sorted in non-decreasing order, and two integers `m` and `n`, representing the number of elements in `nums1` and `nums2` respectively.

Merge `nums1` and `nums2` into a single array sorted in non-decreasing order.

The final sorted array should not be returned by the function, but instead be stored inside the array `nums1`. To accommodate this, `nums1` has a length of  $m + n$ , where the first  $m$  elements denote the elements that should be merged, and the last  $n$  elements are set to 0 and should be ignored. `nums2` has a length of  $n$ .

#### Example 1:

Input: `nums1` = [1,2,3,0,0,0],  $m$  = 3, `nums2` = [2,5,6],  $n$  = 3

Output: [1,2,2,3,5,6]

**Explanation:** The arrays we are merging are [1,2,3] and [2,5,6].

The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from `nums1`.

```
1 # Input: nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3
2 # Output: [1,2,2,3,5,6]
3
4
5 nums1 = [1, 2, 3, 0, 0, 0]
6 m = 3
7 nums2 = [2, 5, 6]
8 n = 3
9
10 nums1[3:] = nums2
11 nums1.sort()
12 nums1
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

[>] Python - a

```
>>> n = 3
>>> nums1[3:] = nums2
>>> nums1.sort()
>>> nums1
[1, 2, 2, 3, 5, 6]
>>>
```

**Q6.** Given an integer array `nums`, return true if any value appears at least twice in the array, and return false if every element is distinct.

**Example 1:**

Input: `nums` = [1,2,3,1]

Output: true

```
1 #Input: nums = [1,2,3,1]
2 #Output: true
3
4 nums = [1,2,3,1]
5 k=False
6 for i in range (len(nums)):
7     for j in range (i+1, len(nums)):
8         if (nums[i] == nums[j]):
9             k=True
10            print(k)
11            break
12 if k is False:
13     print(k)
14
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
...
...
...
...
True
```

**Q7.** Given an integer array `nums`, move all 0's to the end of it while maintaining the relative order of the nonzero elements.

Note that you must do this in-place without making a copy of the array.

**Example 1:**

Input: `nums = [0,1,0,3,12]`

Output: `[1,3,12,0,0]`

```
1
2      #Input: nums = [0,1,0,3,12]
3      #Output: [1,3,12,0,0]
4
5      nums = [0,1,0,3,12]
6      k=0
7
8      for i in range (len(nums)):
9          if(nums[i]==0):
10              nums.remove(0)
11              k+=1
12      for j in range (0,k):
13          nums.append(0)
14
15      print(nums)
16
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

```
>>> for j in range (0,k):
...     nums.append(0)
...
>>> nums
[1, 13, 12, 0, 0]
```

**Q8.** You have a set of integers  $s$ , which originally contains all the numbers from 1 to  $n$ . Unfortunately, due to some error, one of the numbers in  $s$  got duplicated to another number in the set, which results in repetition of one number and loss of another number.

You are given an integer array  $\text{nums}$  representing the data status of this set after the error.

Find the number that occurs twice and the number that is missing and return them in the form of an array.

**Example 1:**

Input:  $\text{nums} = [1,2,2,4]$

Output:  $[2,3]$

```
1
2      # Find the number that occurs twice and t
3
4      # Input: nums = [1,2,2,4]
5      # Output: [2,3]
6
7      nums = [1,2,2,4]
8      a=[]
9      nums.sort()
10     for i in range (len(nums)):
11         if (nums[i] == nums[i+1]):
12             a.append(nums[i])
13             a.append(nums[i]+1)
14     print(a)
15
16
17
```

---

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

---

```
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
IndexError: list index out of range
>>> print(a)
[2, 3]
>>>
```

