IARE

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING
(An Autonomous institute affiliated to JNTUH, Hyderabad)
Dundigal, Hyderabad - 500 043

LABORATORY WORK BOOK

Name of the Student: Muhammad fauzan zohaib

Class: CSE - C Semester: II

Roll Number									
2	3	9	3	1	A	0	5	5	H

Course Code: ACSD06 Course Name: PPS Laboratory

Name of the Course Faculty: Dr M. Madhusudhan Reddy Faculty ID: IARE 10881

Exercise Number: 02 Week Number: 02 Date:

S. No.	Exercise Number	EXERCISE NAME	MARKS AWARDED						
			Aim/ Preparation	Algorithm / Procedure / Performance in the Lab		Source Code / Calculations and Graphs	Program Execution / Results and Error Analysis	Viva - Voce	Total
			4	4		4	4	4	20
1	2.1	Contains Duplicate	4		4	4	4	4	20
2	2.2	Roman to Integer							
3	2.3	Two sum							
4	2.4	plus one							
5	2.5	Majority element							
6	2.6	Richest customer wealth							
7	2.7	Fizz Buzz							
8	2.8	Step 2 N							
9	2.9	Running the sum of 1D Array.							
10	2.10	Remove element.							
11									
12									

Signature of the Student

Signature of the Faculty 3/4/24

START WRITING FROM HERE

1. Contains Duplicate

Q- 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

code:-

```
def contains_duplicate(nums):
    unique_nums = set()
    for num in nums:
        if nums in unique_nums:
            return True
        unique_nums.add(num)
        return False
nums = [1,2,3,4,5]
print(contains_duplicate(nums))
```

output:-

False.

---

2. Roman to Integer

Q- Roman numerals are represented by seven different symbols I, V, X, L, C, D and M

code:-

```
def romanToInteger(self):
    m = {
        'I' : 1,
        'V' : 5,
        'X' : 10,
        'L' : 50,
        'C' : 100,
        'D' : 500,
        'M' : 1000,
```

```
    }
    ans = 0
    for i in range (len(self)):
        if i < len(self) - 1 and m[self[i]] < m[s[i+1]]:
            ans -= m[self[i]]
        else:
            ans += m[self[i]]
    return ans
s = int("Enter your roman number : ")
print (romanToInt(s))
```

output:-

Enter your roman number : 100

C.

3. Two Sum

Q:- Given an array of integers nums and an integer target, return indices of the two numbers such they add up to target

code:-

```
def twoSum (nums, target):
    m = {}
    for i,x in enumerate (nums):
        y = target - x
        if y in m
            return [m[y], i]
        m[x] = i
nums = [12, 5, -3, 4
target = 9
print (twoSums (nums, target)).
```

output:- [0 : 2] -

ROLL NUMBER :

4. plus one

Q:- you are given an array, where each digits[i] is the ith digit of the integer. Large integer does not contain any leading 0's. increement the large integer by one and return the resulting of array.

code:-

```
def plusOne(digits):
    if digits[-1] < 9:
        digits[-1] += 1
        return digits
    elif len(digits) == 1 and digits[0] == 9:
        return [1, 0]
    else:
        digits[-1] = 0
        digits[0:-1] = plusOne(digits[0:-1])
digits = [8]
print(plusOne(digits))
```

output:-

9.

5. Majority elements

Q:- Given of array of size n, return the majority element. The majority element is the element that appears more than [n/2] times

code:-

```
def majority-element(nums):
    count = 0
    candidate = None

    for num in nums:
        if count == 0:
```

```
            candidate = num
        count += (1 if num == candidate else -1)
    return candidate
nums = [3,2,3]
print(majority_element(nums))
```

output :-

3

---

6. Richest customer wealth

Q:- you are given an m x n integer grid account where accounts[i][j] is the amount of money the ith customer has in the jth bank. The richest customer is the customer that has the maximum wealth.

code:-

```
def maximumWealth(accounts):
    max = 0
    for i in range(len(accounts)):
        count = 0
        for j in range(len(accounts[i])):
            count += accounts[i][j]
        if count > max:
            max = count
    return max
accounts = [[2,8,7],[7,1,3],[1,9,5]]
print(maximumWealth(accounts))
```

output:-

17

ROLL NUMBER :

7. Fizz Buzz

Given an integer n, return a string array answer (1-indexed) where:

answer[i] == "fizzBuzz" if i is divisible by 3 and 5.

answer[i] == "Fizz" if i is divisible by 3.

answer[i] == "Buzz" if i is divisible by 5.

answer[i] == i (as a string) if none of the conditions are true.

Code:-

```
def fizzBuzz(n):
    return_list = []
    for i in range(1, n+1):
        if i%3 == 0 and i%5 == 0:
            return_list.append("fizzBuzz")
        elif i%3 == 0:
            return_list.append("fizz")
        elif i%5 == 0:
            return_list.append("Buzz")
        else:
            return_list.append(str(i))
    return return_list return_list
n = int(input("Enter the value of n: "))
print(fizzBuzz(n)).
```

output:-

Enter the value of n: 15

['1', '2', 'fizz', '4', 'Buzz', 'fizz', '7', '8', 'fizz', 'Buzz', '11', 'fizz', '13', '14', 'fizzBuzz'].

ROLL NUMBER:

8. Number of steps to reduce a number to zero.

Given a integer num, return the number of steps to reduce it to zero.

Code:-

```
def step2N (n):
    count = 0
    while (n):
        if n%2 = = 0;
            count + = 1
            n = n/2
        else:
            count + = 1
            n = n-1
    return count

n = 14
print (step2N(n))
n = 8
print (step2N(n))
n = 123
print (step2N(n))
```

output:-

6

4

12

ROLL NUMBER :

9. Running Sum of 1-D Array.

Given an array nums. we define a running sum of an array as runningSum[i] = sum(nums[0]...nums[i]). Return the running sum of nums.

code:-

```
def runningSum(nums):
    for i in range(1, len(nums)):
        nums[i] = nums[i]+ num[i-1]

    return nums
nams = [1,2,3,4]
point (runningsum(nums))
```

output:-

[1, 3, 6, 10].

10. Remove element.

Given an integer array nums and integer val, remove all occurence of val in nums in-place.

code:-

```
def removeelement (nums, val):
    i = 0
    while (i< len (nums):
        if (nums[i] == val):
            nums.pop (i)

        else :
```

```
            i+ = 1
        return len(nums)
nums = [3,2,2,3]
val  = 2
k = (removeelement(nums,val))
print(k)
print(nums[:k])
```

output:-

2

[3,3].

Pradeep
3/4/24