**15th March**

**Problem Statement: 1**

**Given a range [m, n] (both inclusive) where 0 <= m, n <= 10000, find the sum of all integers between m and n**

**Example:**

**Input:**

**0 3**

**Output:**

**6**

**Explanation:**

**0+1+2+3 = 6**

**Problem Statement: 2**

**3. Find a Unique Element in an Array**

**You are given an array containing N integers where only one element is unique (appears exactly once), while all other elements appear twice. Find and return the unique element.**

**Example:**

**Input:**

**arr = [5, 3, 2, 3, 2]**

**Output:**

**5**

**2. Minimum Team Selection to Cover Required Skills**

**Problem Statement: 3**

**You are given a list of required skills and a list of candidates, where each candidate has a subset of skills. Your task is to find the smallest possible team such that all required skills are covered.**

**You will be given:**

**1. Required skills list**

**2. Number of candidates (N)**

**3. Skillsets of N candidates**

**Return the indices of selected candidates forming the smallest team.**

**Example:**

**Input:**

**A b c d**

**4**

**a b**

**b c**

**c d**

**d**

**Output:**

**0 2**

**20 March**

Q Shortest Path in a Directed Acyclic Graph

Given a Directed Acyclic Graph (DAG) with N vertices and E edges,

find the shortest path from the initial state to the destination state.

Input : 5

7

0 1 5

0 2 4

1 2 5

1 3 7

2 3 6

2 4 3

3 4 7

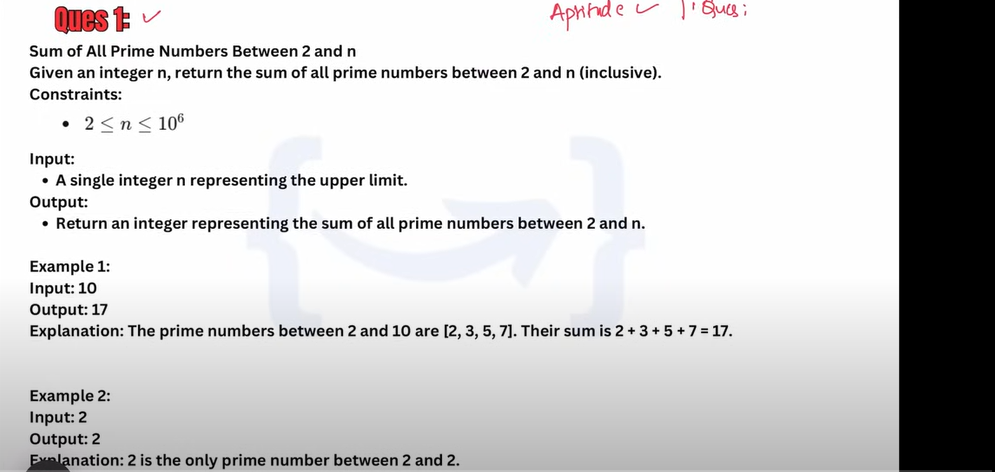
0

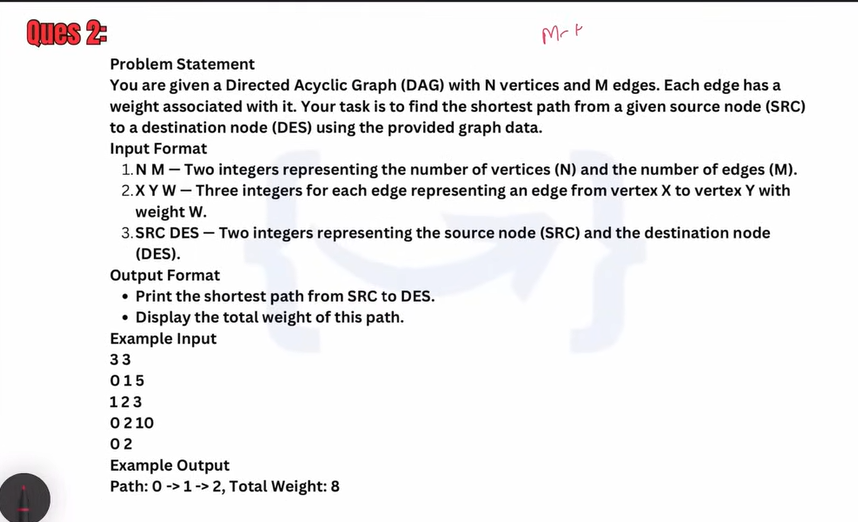
4

Output path

0 -> 2 -> 4

Total weight: 7

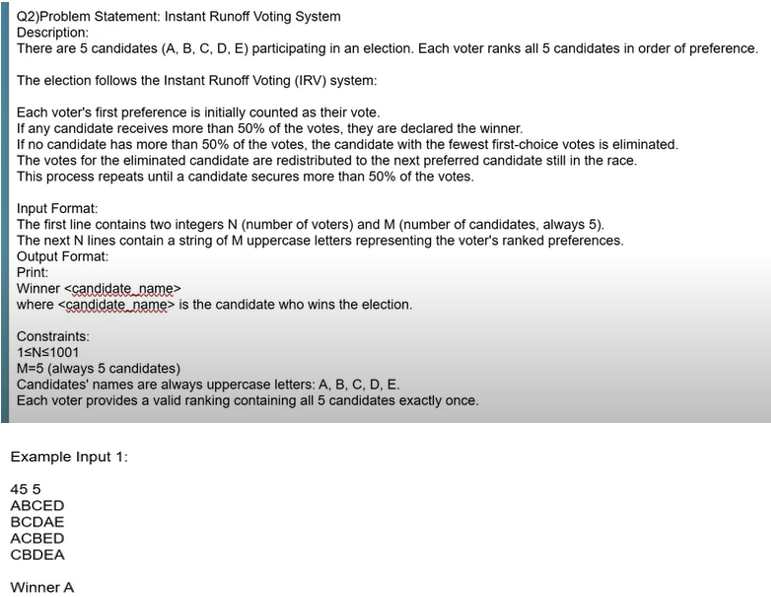




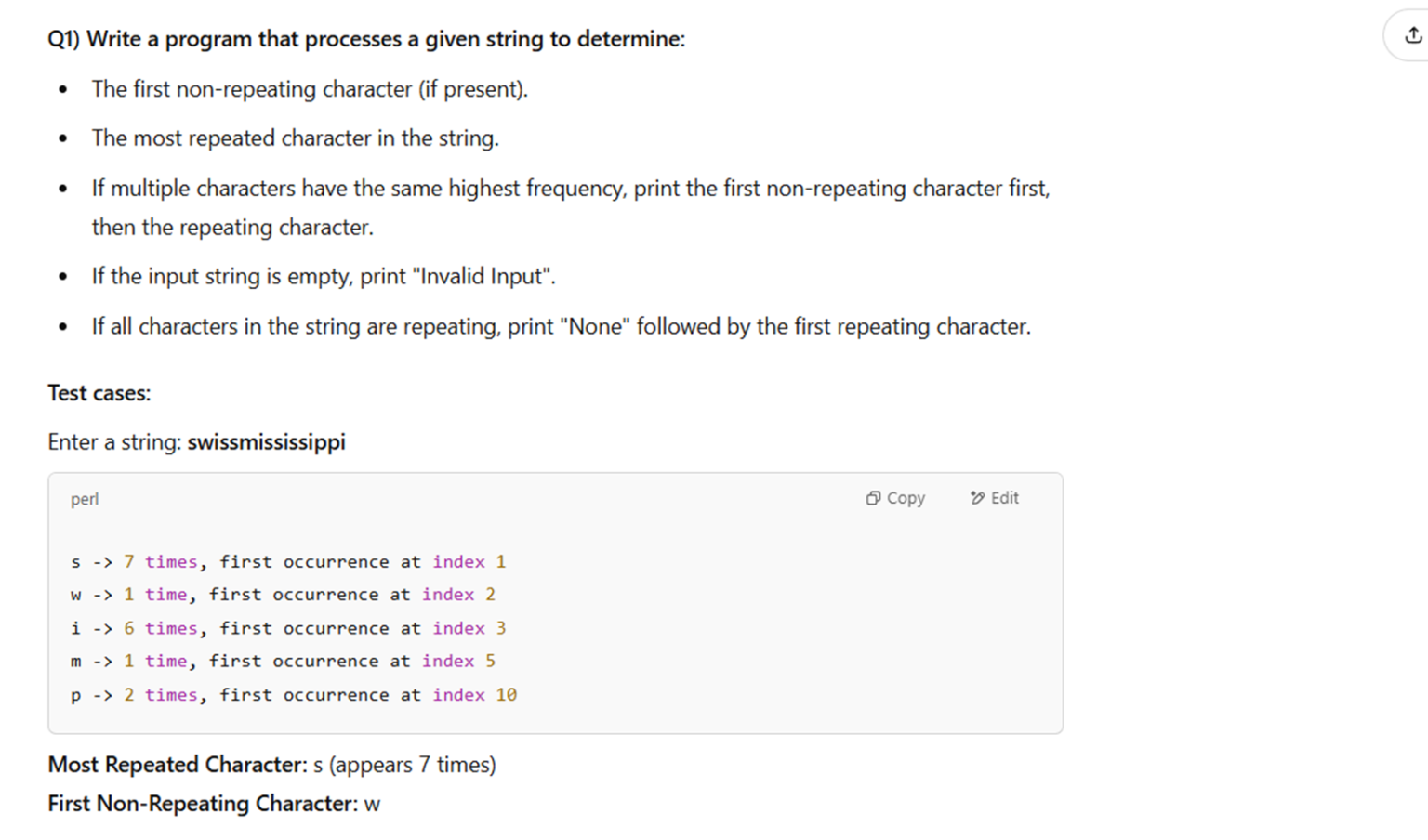
**26 March**

26 coding

Q1. Find LCM and GCD







**from collections import Counter**

**s = input("Enter a string: ")**

**if not s:**

**print("Invalid Input")**

**else:**

**l = []**

**char\_count\_dict = Counter(s)**

**print(char\_count\_dict)**

**first\_non\_repeating\_char =  0**

**Most\_repeated\_char\_count = 0**

**for char in s:**

**if char\_count\_dict[char] == 1:**

**first\_non\_repeating\_char = char**

**break**

**Most\_repeated\_char = max(char\_count\_dict, key = char\_count\_dict.get)**

**Most\_repeated\_char\_count = char\_count\_dict[Most\_repeated\_char]**

**if first\_non\_repeating\_char:**

**print("First Non reapteating char", first\_non\_repeating\_char)**

**else:**

**for char in s:**

**if char\_count\_dict[char]> 1:**

**print("None")**

**break**

**print("Most Repeated Char", Most\_repeated\_char, "(", Most\_repeated\_char\_count,")")**

Q2.Write a program that continuously takes user input for the following details:

* Income (amount earned).
* Type of Material (category of expenditure).
* Expenditure on that Material (amount spent).

The input process continues until the user enters “done”.

After input completion, the program should:

* Display the total income.
* Calculate and display the total savings (Income – Total Expenditure).
* List where the money was spent along with respective expenditure.

