Experiment 1

Student Name: Nitil Jakhar
UID: 22BCS17300
Branch: CSE
Section/Group: NTPP_IOT-602/A
Semester: 6th
Date of Performance: 3/02/25
Subject: AP 2

1. **Aim: Arrays and logic building**
2. **Objective:**
   1. Remove duplicates from a sorted array
   2. Set Matrix zeroes
3. **Code:**

1. **Remove duplicates from a sorted array:**

```
from typing import List

class Solution:
    def removeDuplicates(self, nums: List[int]) -> int:
        if not nums:
            return 0  # Edge case: empty list

        i = 0  # Pointer for the last unique element

        for j in range(1, len(nums)):  # Iterate through the list
            if nums[j] != nums[i]:  # Found a new unique element
                i += 1
                nums[i] = nums[j]  # Update position i with new unique element

        return i + 1  # Length of unique elements
```

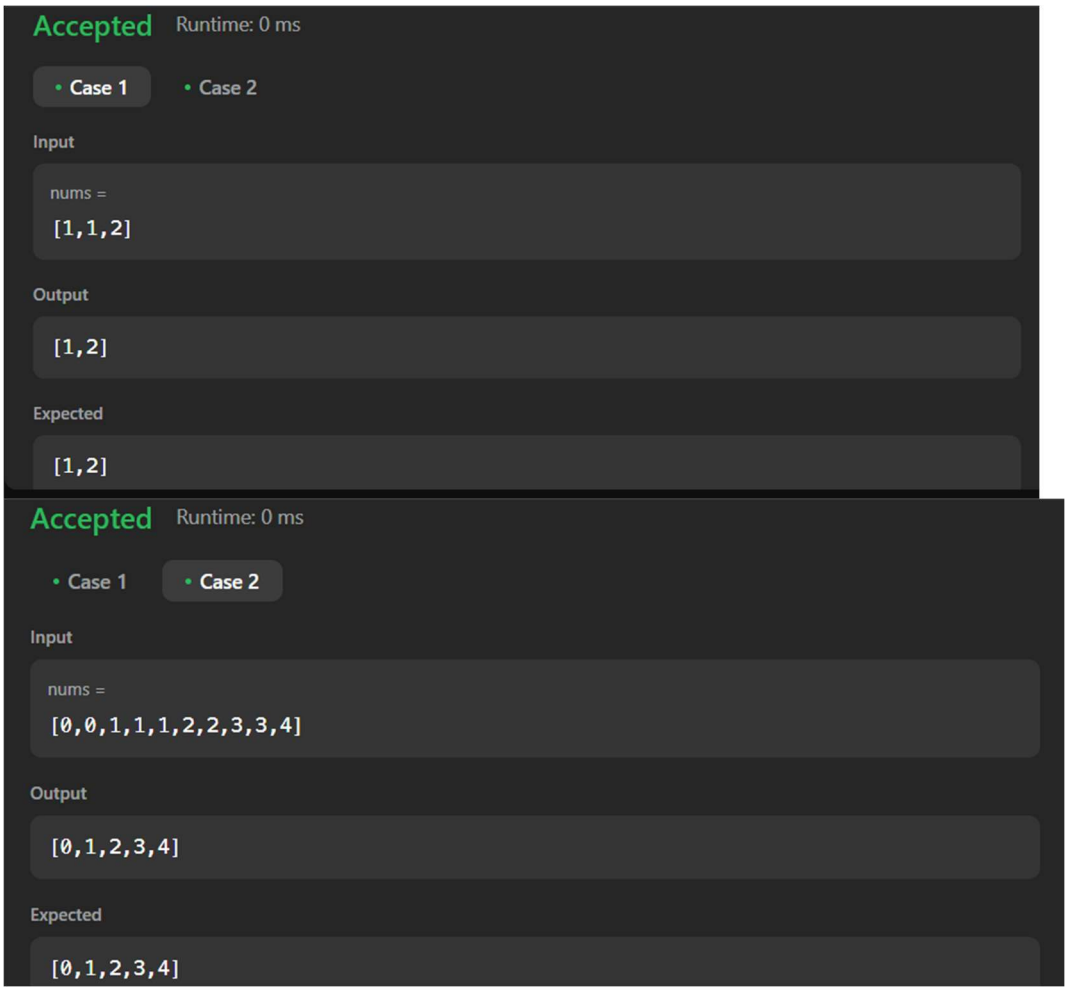
2. **Set Matrix zeroes:**

```
from typing import List

class Solution:
    def setZeroes(self, matrix: List[List[int]]) -> None:
        m, n = len(matrix), len(matrix[0])
        first_row_has_zero = any(matrix[0][j] == 0 for j in range(n))
        first_col_has_zero = any(matrix[i][0] == 0 for i in range(m))
```

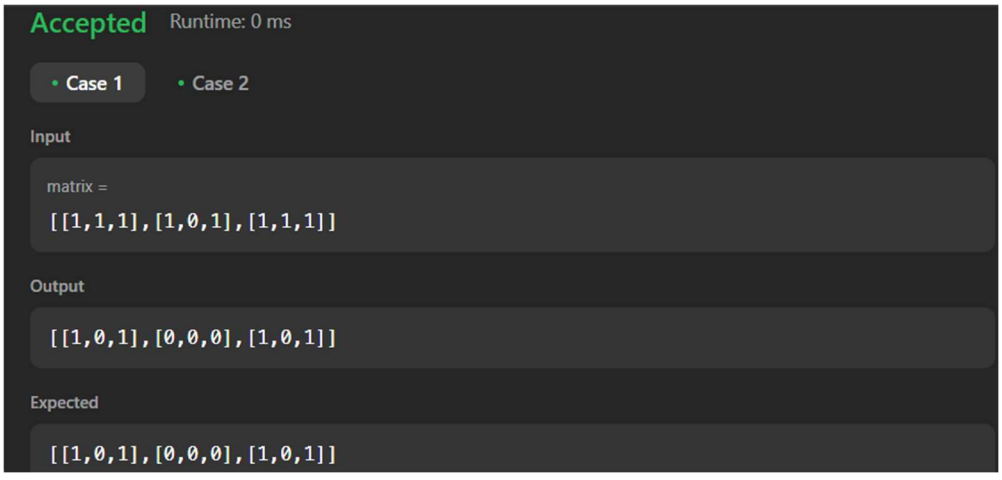
```
# Use first row and column as markers
for i in range(1, m):
for j in range(1, n):
if matrix[i][j] == 0:
matrix[i][0] = 0  # Mark row
matrix[0][j] = 0  # Mark column

# Set matrix cells to zero based on markers
for i in range(1, m):
for j in range(1, n):
if matrix[i][0] == 0 or matrix[0][j] == 0:
matrix[i][j] = 0

# Handle first row
if first_row_has_zero:
for j in range(n):
matrix[0][j] = 0

# Handle first column
if first_col_has_zero:
for i in range(m):
matrix[i][0] = 0
```

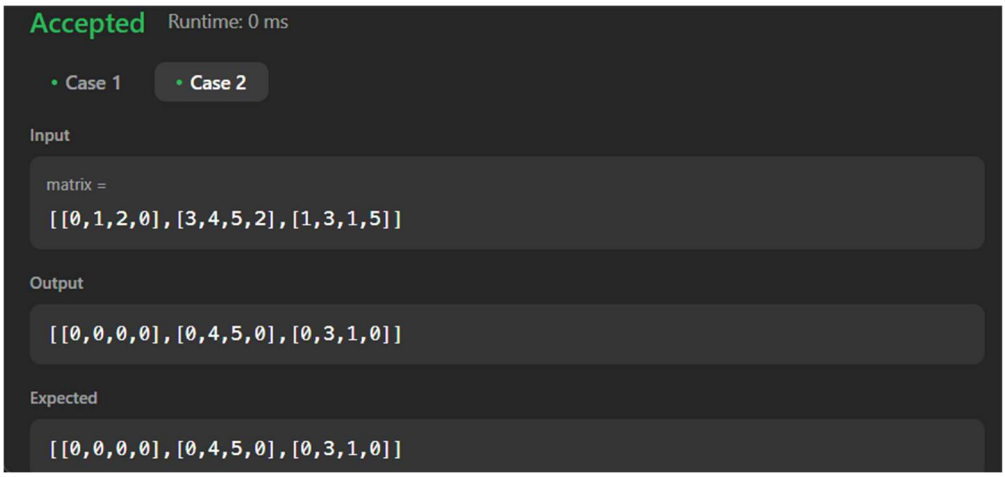
4. Output:

1) Longest Nice Substring



2) Search 2d matrix 2





5. Learning Outcome

1) Learned how to modify a matrix without using extra space.
2) Used the first row and column as markers to track which rows/columns should be zeroed.
3) Applied a two-pass approach to avoid premature overwrites.
4) Addressed cases where the first row/column contains zeroes.
5) Achieved O(m×n)O(m \times n)O(m×n) time complexity with O(1)O(1)O(1) extra space.