**Task 4 a:**

**Program**

def smallestCommonElement(mat):

rows = len(mat)

cols = len(mat[0])

# Check each number in first row

for num in mat[0]:

found = True

for r in range(1, rows):

# Binary search in each row

left, right = 0, cols - 1

while left <= right:

mid = (left + right) // 2

if mat[r][mid] == num:

break

elif mat[r][mid] < num:

left = mid + 1

else:

right = mid - 1

else: # executed if while loop doesn't break

found = False

break

if found:

return num

return -1

# --- Get matrix input from user ---

rows = int(input("Enter number of rows: "))

cols = int(input("Enter number of columns: "))

mat = []

print("Enter matrix values row by row (sorted in increasing order):")

for \_ in range(rows):

row = list(map(int, input().split()))

mat.append(row)

# Find and display smallest common element

result = smallestCommonElement(mat)

print("Smallest common element in all rows:", result)

Example 1:

Input:

4 5

1 2 3 4 5

2 4 5 8 10

3 5 7 9 11

1 3 5 7 9

 Output:

5

**task 4b:**

**Program**

def countBits(n):

ans = []

for i in range(n + 1):

ans.append(bin(i).count('1'))

return ans

# Get input from user

n = int(input("Enter a non-negative integer: "))

result = countBits(n)

print("Count of 1's for numbers from 0 to", n, ":", result)

**Example**

Enter a non-negative integer: 5

Count of 1's for numbers from 0 to 5 : [0, 1, 1, 2, 1, 2]

**Task 4 c:**

**Program**

def count\_operations(nums):

nums = list(nums) # Convert tuple to list for easier modification

operations = 0

while nums:

if nums[0] == min(nums):

nums.pop(0) # Remove first element

else:

nums.append(nums.pop(0)) # Move first element to end

operations += 1

return operations

# Get input from user

nums = tuple(map(int, input("Enter distinct integers separated by space: ").split()))

print("Number of operations:", count\_operations(nums))

Example

Enter distinct integers separated by space: 3 1 2

Number of operations: 5

**Task 4 d:**

**Program**

def find\_duplicate(nums):

seen = set()

for num in nums:

if num in seen:

return num

seen.add(num)

# Get input from user

nums = list(map(int, input("Enter numbers separated by space: ").split()))

duplicate = find\_duplicate(nums)

print("The repeated number is:", duplicate)

**Example**

Enter numbers separated by space: 3 1 3 4 2

The repeated number is: 3

**Task 4 e:**

students = {}

n = int(input("Enter number of students: "))

for \_ in range(n):

name = input("\nEnter student name: ")

test = float(input("Enter Test mark: "))

assignment = float(input("Enter Assignment mark: "))

lab = float(input("Enter Lab mark: "))

students[name] = [test, assignment, lab]

# Function to calculate average

def average(marks):

return sum(marks) / len(marks)

# 1. Highest average score

max\_avg = max(average(marks) for marks in students.values())

highest\_avg\_students = [name for name, marks in students.items() if average(marks) == max\_avg]

# 2. Highest assignment marks

max\_assignment = max(marks[1] for marks in students.values())

highest\_assignment\_students = [name for name, marks in students.items() if marks[1] == max\_assignment]

# 3. Lowest lab marks

min\_lab = min(marks[2] for marks in students.values())

lowest\_lab\_students = [name for name, marks in students.items() if marks[2] == min\_lab]

# 4. Lowest average score

min\_avg = min(average(marks) for marks in students.values())

lowest\_avg\_students = [name for name, marks in students.items() if average(marks) == min\_avg]

# Display results

print("\n--- Results ---")

print("Highest average score:", highest\_avg\_students, "with average =", max\_avg)

print("Highest assignment marks:", highest\_assignment\_students, "with marks =", max\_assignment)

print("Lowest lab marks:", lowest\_lab\_students, "with marks =", min\_lab)

print("Lowest average score:", lowest\_avg\_students, "with average =", min\_avg)

Example

Enter number of students: 3

Enter student name: Alice

Enter Test mark: 85

Enter Assignment mark: 90

Enter Lab mark: 80

Enter student name: Bob

Enter Test mark: 78

Enter Assignment mark: 92

Enter Lab mark: 70

Enter student name: Charlie

Enter Test mark: 85

Enter Assignment mark: 85

Enter Lab mark: 85

**Output**-

Highest average score: ['Charlie'] with average = 85.0

Highest assignment marks: ['Bob'] with marks = 92.0

Lowest lab marks: ['Bob'] with marks = 70.0

Lowest average score: ['Bob'] with average = 80.0