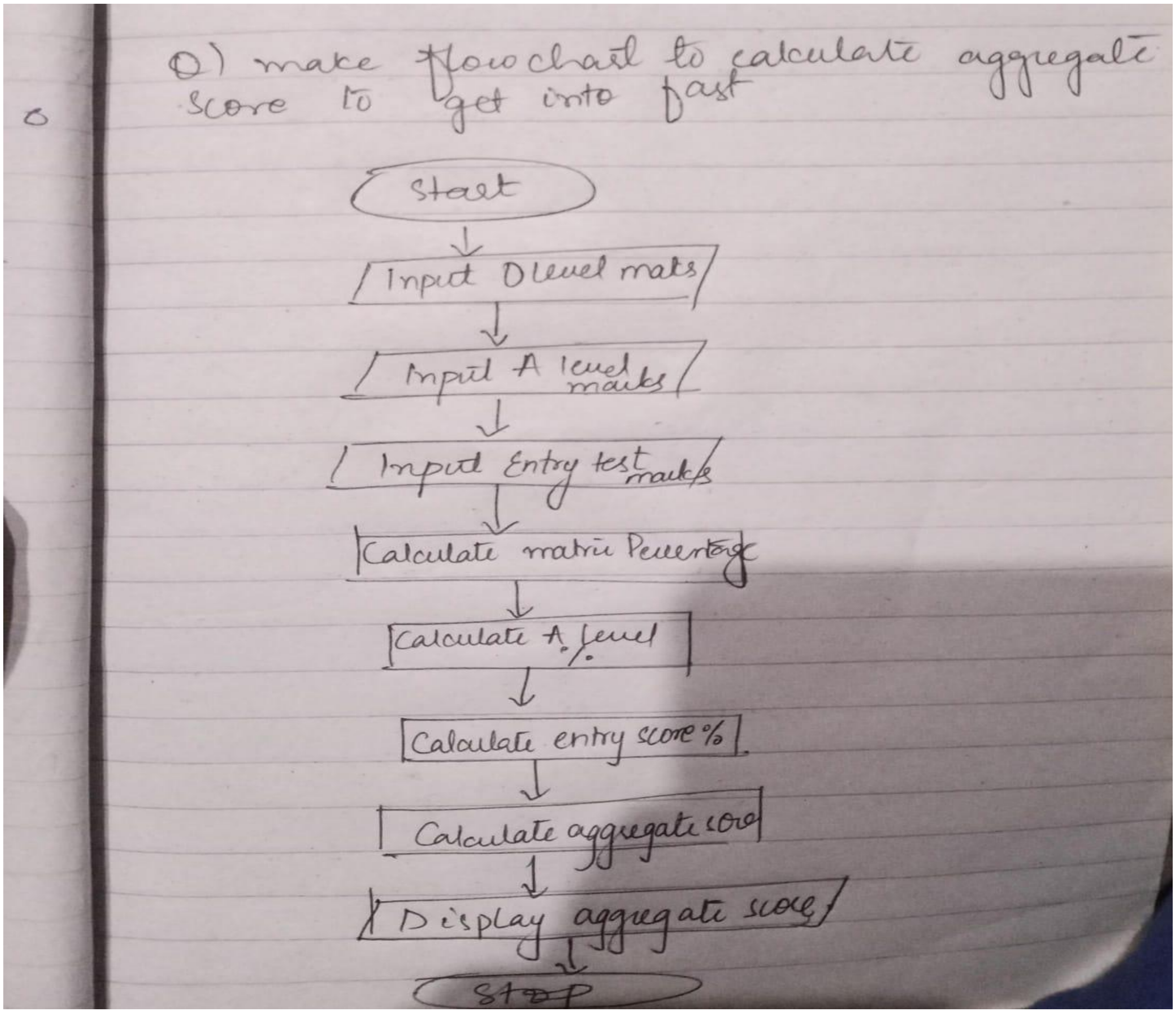
MOIZ RAZA KHAN

24K-1011

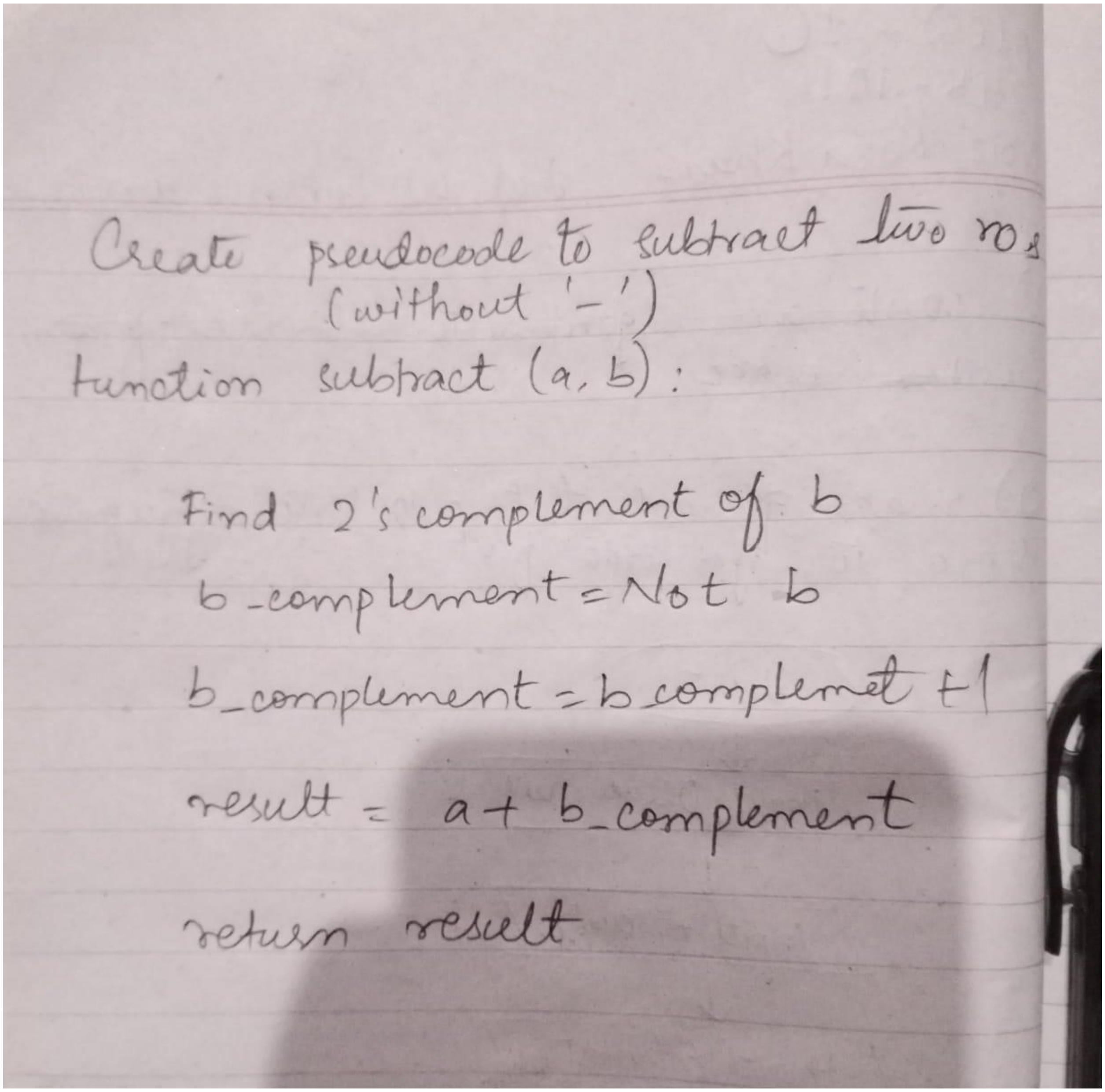
SECTION C

LAB 2

Make a flowchart to calculate aggregate score of university named FAST



Q) make flowchart to calculate aggregate score to get into fast



Create pseudocode to subtract two nos
(without '-')

```
function subtract (a, b):

    Find 2's complement of b
    b-complement = Not b
    b_complement = b complemet +1
    result = a+ b_complement
    return result
```

1. Write an algorithm to determine whether a number is a prime number. The algorithm should iterate through possible divisors and determine if the number has any divisors other than 1 and itself.

Input: A number n.

Step 1: If n is less than or equal to 1, return False. (Since numbers less than 2 are not prime.)

Step 2: If n is 2 or 3, return True. (2 and 3 are prime numbers.)

Step 3: If n is divisible by 2 or 3, return False. (If a number is divisible by 2 or 3, it is not prime.)

Step 4: Initialize a variable i to 5. This variable will be used to check divisors.

Step 5: While i * i is less than or equal to n, do the following:

- If n is divisible by i, return False. (If n is divisible by any i, it is not prime.)
- If n is divisible by i + 2, return False. (This checks divisibility by numbers like 5, 7, 11, 13, etc.)
- Increment i by 6. (This step skips even numbers and multiples of 3.)

Step 6: If no divisors are found, return True.

2. Create an algorithm that asks the user for a day number (1-365) and outputs the corresponding day of the week, assuming that January 1st is a Monday.

Input: A day number n (from 1 to 365).

Steps:

Step 1: Define an array daysOfWeek containing the names of the days in order: ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"].

Step 2: Calculate the index in the daysOfWeek array that corresponds to the day number n by using the formula (n - 1) % 7.

  - (n - 1) is used because the array index starts at 0.

Step 3: Use the calculated index to find the day of the week in the daysOfWeek array.

Step 4: Output the day of the week.

3. Develop an algorithm for a program that takes two numbers as input and finds the Greatest Common Divisor (GCD) of the two numbers using the Euclidean algorithm.

Input: Two positive integers a and b.

Output: The GCD of a and b.

Steps:

1. **Step 1**: If b is 0, return a as the GCD. (This is the base case of the Euclidean algorithm.)
2. **Step 2**: Otherwise, replace a with b and b with the remainder of a divided by b (a % b).
3. **Step 3**: Repeat Step 1 and Step 2 until b becomes 0.
4. **Step 4**: The value of a at this point is the GCD.

1.Write pseudocode to find the smallest number among three given variables. Implement a decision-making structure to compare the variables.

```
function findSmallestNumber(a, b, c):
  # Initialize a variable to hold the smallest number, starting with the first variable
  smallest = a

  # Compare smallest with the second variable
  if b < smallest:
    smallest = b

  # Compare smallest with the third variable
  if c < smallest:
    smallest = c

  # At this point, smallest contains the smallest of the three variables
  return smallest
```

2. Create pseudocode to subtract two numbers without using the - operator. (Hint: Use addition and complement techniques.)

```
function subtractWithoutMinusOperator(a, b):
  # Step 1: Find the two's complement of b
  # The two's complement is obtained by inverting the bits of b and adding 1

  # Invert the bits of b
  b_complement = NOT b

  # Add 1 to the complement
  b_complement = b_complement + 1

  # Step 2: Add a and the two's complement of b
  result = a + b_complement
```

```
    # Step 3: Return the result
    return result
```

3. Develop pseudocode for a basic calculator that performs multiplication and division. The pseudocode should prompt the user for two numbers and an operator, then display the result of the operation.

```
function basicCalculator():
    # Step 1: Prompt the user to enter the first number
    Display "Enter the first number:"
    num1 = Input()

    # Step 2: Prompt the user to enter the second number
    Display "Enter the second number:"
    num2 = Input()

    # Step 3: Prompt the user to enter an operator (* for multiplication, / for division)
    Display "Enter an operator (* for multiplication, / for division):"
    operator = Input()

    # Step 4: Perform the operation based on the entered operator
    if operator == "*":
        # Multiply num1 and num2
        result = num1 * num2
        Display "The result of multiplication is:", result

    else if operator == "/":
        # Check if num2 is not zero to avoid division by zero error
        if num2 == 0:
            Display "Error: Division by zero is not allowed."
        else:
```

```
        # Divide num1 by num2
        result = num1 / num2
        Display "The result of division is:", result

else:
    # If the user enters an invalid operator, display an error message
    Display "Invalid operator. Please enter * for multiplication or / for division."
```