DAY 2 ASSIGNMENT

1. Write a algorithm to calculate the Greatest Common Divisor (GCD) and Least Common Multiple (LCM) of corresponding elements from two arrays, ArrayX and ArrayY. The size of the arrays will be provided as an input.

Algorithm:

```
Step 1: Start
Step 2:Input the size of array
Step 3:Intialize ArrayX and ArrayY ,with given size of array
Step 4:Input elements into both arrays
Step 5:Define functions to calculate GCD and LCM
Step 6:Calculate GCD and LCM using the functions
Step 7:Print GCD and LCM
Step 8:End
```

2. Temperature Monitoring System

Objective: Design a temperature monitoring system that reads temperature data from a sensor and triggers an alarm if the temperature exceeds a predefined threshold.

Requirements:

- Read temperature data from a temperature sensor at regular intervals.
- Compare the read temperature with a predefined threshold.
- If the temperature exceeds the threshold, activate an alarm (e.g., LED or buzzer).
- Include functionality to reset the alarm.

Algorithm:

```
1. Start
2. Set a predefined_threshold and initialize alarm_active = False
3. Read the temperature from the temperature sensor
4. If temperature > predefined_threshold and alarm_activated == false:
       i. Activate the alarm and set alarm_activated=True
5. Else
       i. Reset the alarm and set alarm_activated= False
6. End
```

3. Motor Control System

Objective: Implement a motor control system that adjusts the speed of a DC motor based on user input.

Requirements:

- Use a potentiometer to read user input for desired motor speed.
- Control the motor speed using PWM (Pulse Width Modulation).
- Display the current speed on an LCD.

Algorithm:

1. Start
2. Read the input speed from the potentiometer
3. If the current speed is different from input
    i. Control the motor speed using PWM
4. Display the current speed on LCD
5. Repeat the steps continuosly
6. Stop

4. LED Blinking Pattern

Objective: Create an embedded system that controls an array of LEDs to blink in a specific pattern based on user-defined settings.

Requirements:

- Allow users to define blink patterns (e.g., fast, slow).
- Implement different patterns using timers and interrupts.
- Provide feedback through an LCD or serial monitor.

Algorithm:

1. Start
2. Setup the array of LED's
3. Set the timer
4. Take the desired pattern as input from user
5. According to the input adjust the Blinking interval timer based on user defined pattern
6. Display the Current pattern on LCD
7. Repeat

5. Data Logger

Objective: Develop a data logger that collects sensor data over time and stores it in non-volatile memory.

Requirements:

- Read data from sensors (e.g., temperature, humidity) at specified intervals.
- Store collected data in EEPROM or flash memory.
- Implement functionality to retrieve and display logged data

Algorithm:

1. Start
2. Define the time interval for data logging
3. Read the data into variables at the specified interval
4. Store the data in EEPROM or flash memory
5. Display the data when required
6. Repeat the steps

6 Calculator

Pseudocode:

```
1.      read num1 and num2
2.      read the operator
3.      compare the operator

if operator == '+'

        set result=num1 + num2

if operator == '-'

        set result=num1-num2

 if operator =='*'

        set result=num1*num2

 if operator =='/'

        if num2!=0:

        set result =num1/num2

        else:

        display error
```

7 Factorial

Pseudocode:

```
1. read input n
2. set factorial =1
3. for i = 1 to n

        set factorial = factorial *i

  end for

4. display factorial
```

OR

```
Function factorial(n):

    if n == 0 OR n == 1:

        return 1
```

```
  else:
      return n * factorial(n - 1)
end Function


set num = 5
print "Factorial of", num, "is", factorial(num)
END
```

8.Problem Statement: Smart Irrigation System

Objective: Design a smart irrigation system that automatically waters plants based on soil moisture levels and environmental conditions. The system should monitor soil moisture and activate the water pump when the moisture level falls below a predefined threshold.

Requirements:

Inputs:

Outputs:

Conditions:

The pump should only activate if the soil moisture is below the threshold and it is daytime (e.g., between 6 AM and 6 PM).

If the soil moisture is adequate, the system should display a message indicating that watering is not needed.

Activate the water pump when the soil moisture is below the threshold.

Display the current soil moisture level and whether the pump is activated or not.

Soil moisture sensor reading (percentage).

User-defined threshold for soil moisture (percentage).

Time of day (to prevent watering during rain or at night).

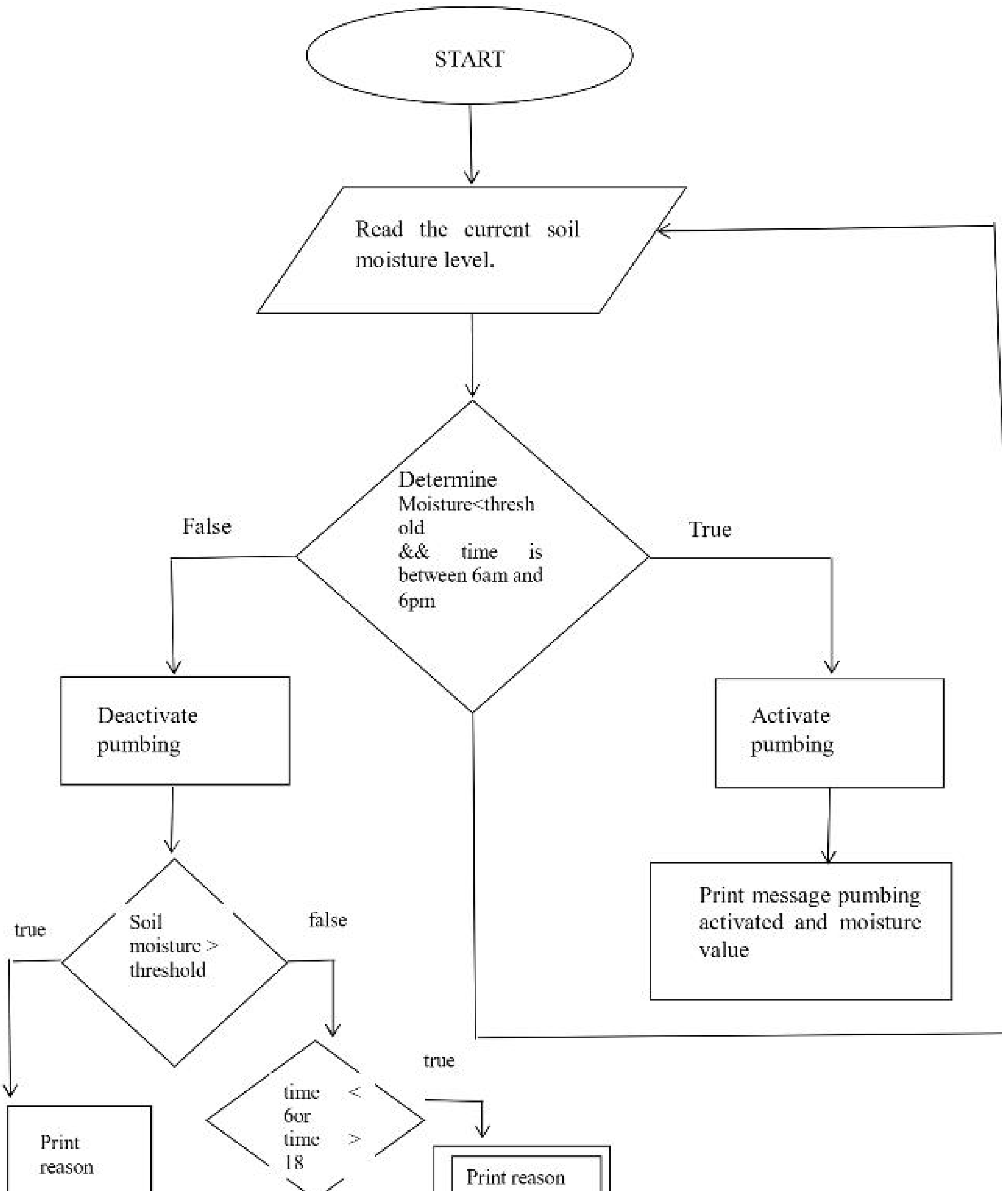
Deliverables: Write pseudocode that outline..

Write pseudocode that outlines the algorithm for the smart irrigation system.

Create a flowchart that visually represents the logic of your pseudocode.

Pseudocode:

```
Step 1: Start

Step 2: Set predefined_threshold

Step 3: Read Soil Moisture sensor data and current_time from clock

Step 4: if soil_moisture_data < predefined_threshold and current_time>=6am AND
current_time<=6pm

                    4.1 If alarm_activated=FALSE

                              4.1.1 activate pump and set alarm_activated=TRUE

                              4.1.2 display "PUMP ACTIVATED"

          4.2 Endif

          4.3 Else

                    if alarm_activated==TRUE then

                    4.3.1 deactivate pump and set alarm_activated=false


          4.4 endif

          display "soil moisture is adequate"

Step 6:Display the current soil moisture level

Step 7:Display the pump is activated or not.

Step 5:Stop
```


START
Read the current soil moisture level.
Determine Moisture<threshold && time is between 6am and 6pm
False
True
Deactivate pumbing
Activate pumbing
Print message pumbing activated and moisture value
true
Soil moisture > threshold
false
time < 6or time > 18
true
Print reason
Print reason