1.Write a program to calculate the Greatest Common Divisor (GCD) and Least Common Multiple (LCM) of corresponding elements from two arrays, Array X and Array Y. The size of the arrays will be provided as an input. Write an algorithm for the above problem statement .

1. Start
2. Input the size of the array n.
3. Initialize Array X and Array Y of size n.
4. Input elements of Array X and Array Y.
5. Calculate GCD and LCM of array X and array Y
6. Print GCD and LCM
7. End.

2.Problem Statement 1: 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.

```
1. Start

2. Initialize the Components(temperature sensor, alarm)

3. Set a Threshold Value for Temperature

4. Continuous Monitor the Temperature

   4.1 Read Temperature
   4.2 If temperature is greater than threshold, activate the alarm.
   4.3 Else deactivate the alarm.

5. End
```

Problem Statement 2: 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.

Step1. Start

Step2. Initialize the components (potentiometer, motor, LCD).

Step3. Read user input from the potentiometer.

Step4. Convert the potentiometer value to a PWM signal.

Step5. Control the motor speed using PWM.

Step6. Display the current motor speed on the LCD.

Step7. Repeat steps 3 to 6.

Step8. End

Problem Statement 3: 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.

Step1. Start

Step2. Initialize Components (LED array, timer, interrupt, LCD or serial monitor)

Step3. Allow User to Define Blink Patterns (e.g., fast, slow)

Step4. Set Blink Pattern Based on User Input

Step5. Implement the Blink Pattern Using Timers and Interrupts

Step6. Provide Feedback on LCD or Serial Monitor

Step7. End

Problem Statement 4: 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
has context menu

Step1. Start

Step2. Initialize Components (sensors, EEPROM/flash memory)

Step3. Set Data Collection Interval

Step4.Read data from sensors (e.g., temperature, humidity).

Step5. Store data in EEPROM

Step6. Retrieve and Display Logged Data

Step7. End

3.Pseudocode for simple calculator

```
Start
  read num1
  read num2
  read operation

  if operation = "+" then
    result = num1 + num2
  elseif operation = "-" then
    result = num1 - num2
  elseif operation = "*" then
    result = num1 * num2
  elseif operation = "/" then
    result = num1 / num2
  else
    display "Error:"
  end if

  display result
end
```

4.Pseudocode for Factorial of number

```
 Start
Input number
Set factorial = 1
for i = 1 to number
   factorial = factorial * i
End for
Output factorial
End
```

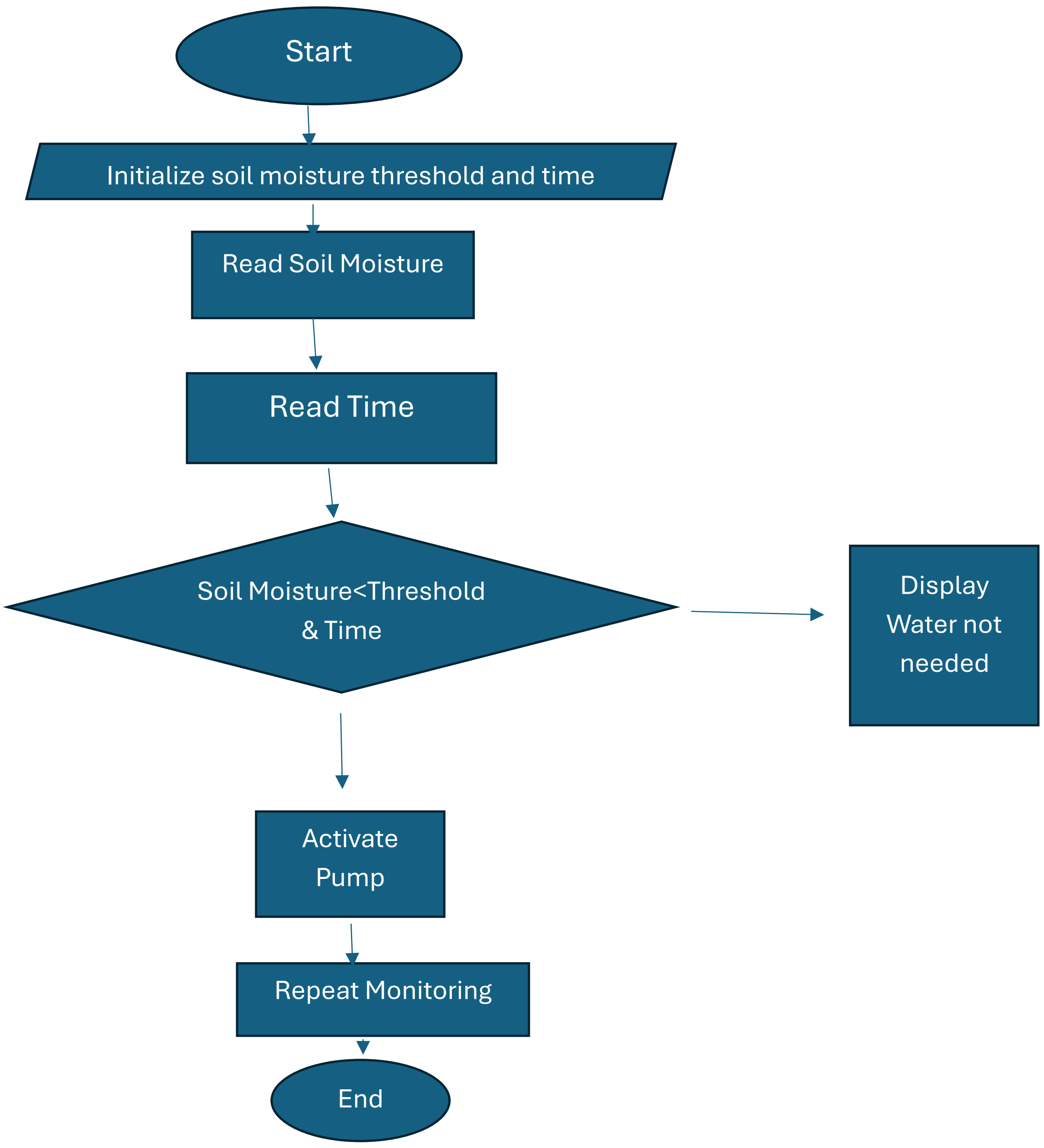
5.Pseudocode for Factorial of number using recursion

```
FUNCTION factorial(n)
 IF n is equal to 0 OR 1 THEN
RETURN 1
 ELSE RETURN n * factorial(n - 1)
 END IF
 END FUNCTION
```

7. 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).
Write pseudocode that outlines the algorithm for the smart irrigation system.
Create a flowchart that visually represents the logic of your pseudocode.
has context menu

```
1. Start System
2. Initialize soil moisture threshold and time range (6 AM to 6 PM).
3. Begin Monitoring Loop:
      3.1Read soil moisture level.
      3.2Read current time.
       3.3Display soil moisture level.
        3.4 If soil moisture < threshold and time :
                    3.4.1Activate water pump.
                    3.4.2Display "Pump Activated".
        3.5 Else:
                3.5.1Display "Watering Not Needed".

4. Repeat Monitoring Loop.
5. End
```


Start
Initialize soil moisture threshold and time
Read Soil Moisture
Read Time
Soil Moisture<Threshold & Time
Display Water not needed
Activate Pump
Repeat Monitoring
End