ANSU MARIUM SHIBU

DAY-2 ASSIGNMENTS

1. 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.

Algorithm:

1. Initialize a predefined threshold

2. At regular intervals read temperature data from sensor

3. IF the temperature is greater than the threshold:

   - Turn On alarm

   Else:

   - Alarm off

4.Reset the alarm

5. Repeat the step

2. 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.

Algorithm:

1. Set up PWM for control motor speed

2. set up a Potentiometer to get user input

3. Set up LCD to display speed

4. Each time it will continuously read the potentiometer value

5. Potentiometer value will set the motor speed using PWM

6. Display current speed using LCD.

7.Repeat the process

3. 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.

Algorithm:

1. set up an array of LED

2. set up the timer for controlling blink intervals

3. set up LCD to provide feedback

4. Allow users to choose a blink pattern

5. Based on user input use the timer to adjust the intervals

6. Blink LED according to the pattern

7. Display the current blink on LCD

8. repeat the process based on input

4. 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

Algorithm:

1. set up sensor

2. Initialize EEPROM or flash memory

3. Set up a timer to read data at specified intervals.

4. At specified intervals read data from sensor

5. store the data in EEPROM or flash memory

6. when the stored data is needed display the data

7. repeat the process collect and stored the data at specified intervals

5. Simple Calculator (Pseudocode)

Problem Statement: Write a program that functions as a simple calculator. It should be able to perform addition, subtraction, multiplication, and division based on user input.

Requirements:

1. Prompt the user to enter two numbers.

2. Ask the user to select an operation (addition, subtraction, multiplication, division).

3. Perform the selected operation and display the result.

4. Handle division by zero appropriately.

Pseudocode:

```
num1 = GetUserInput("Enter the first number: ")
num2 = GetUserInput("Enter the second number: ")
 op = GetUserInput("Enter operation (+, -, *, /): ")
If op is "+"
      res = num1 + num2
  Else If op is "-"
```

```
    res = num1 - num2
  Else If op is "*"
    res = num1 * num2
  Else If op is "/"
    If num2 is not 0
      res = num1 / num2
    Else
      result = "Error"
    End If
  Else
    res = "Invalid operation"
  End
Print result
End Function
```

6. Factorial Calculation(Pseudocode)

Problem Statement: Write a program to calculate the factorial of a given non-negative integer.

Requirements:

1. Prompt the user to enter a non-negative integer.
2. Calculate the factorial using a loop.
3. Display the factorial of the number

Pseudocode:

```
 um = GetUserInput("enter non negative integer”)
facto = 1
For i = 1 to num
    facto = facto * i
  End For
Print ("The factorial of ", num, " is: ", facto)
```

7. Factorial Calculation(Pseudocode)

(Recursive)

```
Function Factorial(n)
   If n == 1
  Return 1
   Else
      Return n * Factorial(n - 1)
   End If
End Function
```

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 outlines the algorithm for the smart irrigation system.

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

Pseudocode:

1. SET threshold = 30

2.SET current_soil_moisture = GetUserInput("Enter current soil moisture:")

3.SET current_time = GetUserInput("Enter current time :")

4 WHILE (true)

a. READ current_soil_moisture

b. READ current_time

c. IF (current_soil_moisture < threshold) AND (current_time between 6 AM and 6 PM) THEN

i. ACTIVATE water_pump

ii. DISPLAY "Watering plants. Current soil moisture: " + current_soil_moisture

ELSE

i. DEACTIVATE water_pump

ii. DISPLAY "Watering not needed. Current soil moisture: " + current_soil_moisture

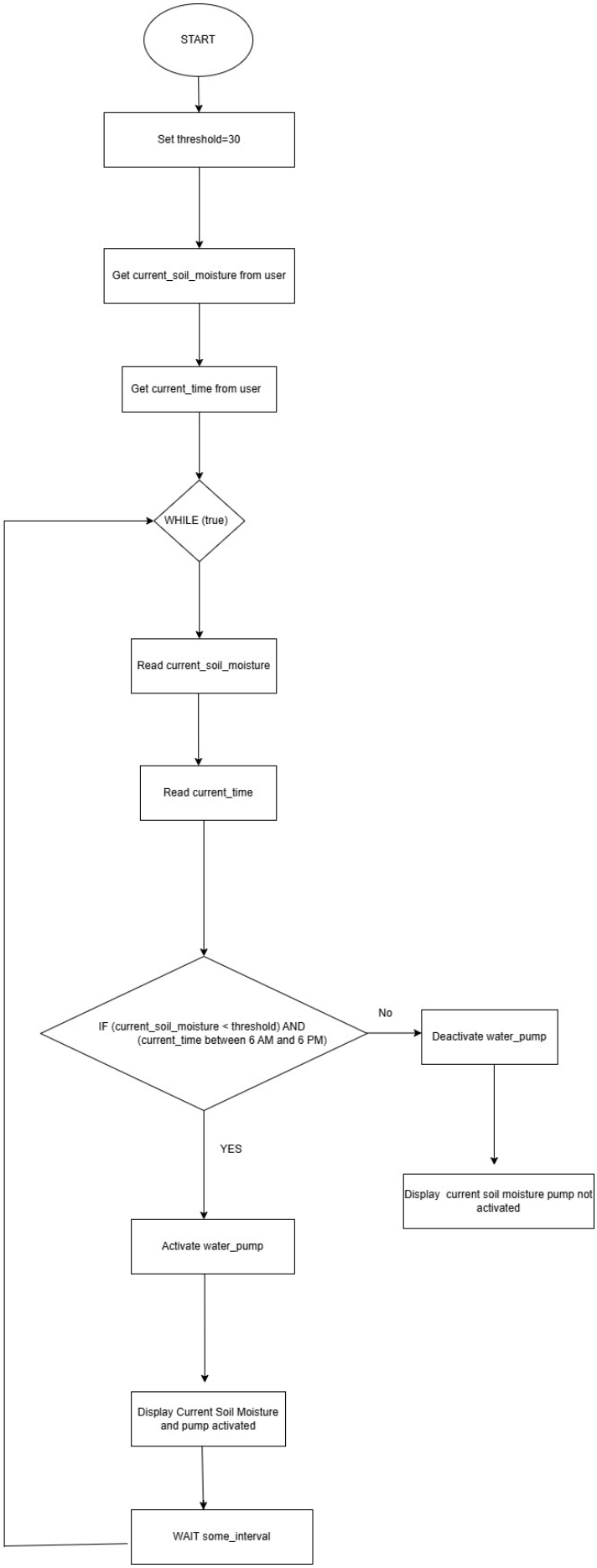
ENDIF

3. WAIT some_interval .

END WHILE

END

Flowchart:


START
Set threshold=30
Get current_soil_moisture from user
Get current_time from user
WHILE (true)
Read current_soil_moisture
Read current_time
IF (current_soil_moisture < threshold) AND
(current_time between 6 AM and 6 PM)
No
Deactivate water_pump
Display current soil moisture pump not activated
YES
Activate water_pump
Display Current Soil Moisture and pump activated
WAIT some_interval

9. Write a program 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.

Write an algorithm for the above problem statement

1. input the size n of the array

2. Input the elements of array x and array y each size n.

3. take each element pair

- compute GCD until the remainder becomes 0
- compute lcm

6. output gcd and lcm for each pair