**Algorithm:**

PS4: 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.

step 1: get the size of arrayX and arrayY

step 2: get the elements of the arrays

step 3: using loop get the corresponding pair from both the arrays

step 4: find the GCD and LCM of the numbers

gcd(arrayX[i], arrayY[i])

while (arrayY[i] != 0)

temp = arrayY[i]

arrayY[i] = arrayX[i] % arrayY[i]

arrayX[i] = temp

return arrayX[i]

lcm(arrayX[i], arrayY[i])

return ((arrayX[i] \* arrayY[i]) / gcd(arrayX[i], arrayY[i])

step 5: display the answer

step 6: repeat step 3 to 5 until there are no more elements in the arrays

PS1: 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.

step 1: set a threshold value

step 2: read the temperature data using temperature sensor

step 3: compare with the threshold value

step 4: if exceeds

step 4.1: activate alarm

step 4.2: reset the alarm

step 5: repeat at regular intervals

PS2: 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.

step 1: read desired input speed using potentiometer

step 2: read the current DC motor speed

step 3: control the speed based on the user input using PWM

step 4: output the speed using LCD

step 5: repeat step 2 to 4

PS3: 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.

step 1: set the timers and interrupts for different patterns

step 2: user inputs the desired pattern

step 3: decide the pattern to be implemented

step 4: implement the pattern

Step 5: provide feedback through LCD or serial monitor

PS5: 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

step 1: initialize hash tables to store the sensor data

step 2: read the data from the sensors and store it in respective tables

step 3: repeat step 1 and 2 at regular intervals

step 4: using hash keys retrieve the logged data and display it

**Pseudo Code:**

1)Simple Calculator

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.

Enter the numbers num1 and num2

Enter the operation op

switch(op)

case1: if (op == +)

return ans = num1 + num2

case2: if (op == -)

return ans = num1 - num2

case3: if (op == \*)

return ans = num1 \* num2

case4: if (op == /)

if num2 == 0

not possible, exit

else

return ans = num1/num2

case5: default

wrong user input

exit

2)Factorial Calculation

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.

Enter non-negative integer num

initialize i = 1, fact = 1

if (num != 0)

while (i <= num)

do

fact = fact \* i

i++

print fact

3)Factorial Calculation using recursion

factorial (int num)

if (n != 0)

return(num\*factorial(num-1)

else

return 1

enter the non-negative integer num

fact = factorial(num)

print fact

**>>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:

threshold = n

read: moisture m, time t, rain r

if (m <= threshold)

if (6 <= t <= 18 && r == 0)

pump\_state = 1

else

pump\_state = 0

end if

else

write: no watering needed

pump\_state = 0

end if

return m, pump\_state

Flowchart:  
  
