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. Read temperature from sensor after every 30 mins.
2. Initialize and set a threshold value and also the alarm state as 0.
3. Check whether the read temperature greater than the set threshold
    3.1. If *yes* then activate the alarm by setting its state to 1.
    3.2.Else return alarm state as 0.

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. Read desired motor speed from user(mspeed).
2. set currentspeed = mspeed
3. Print CurrentSpeed

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 Timer and interrupts.
2. Read the desired blink pattern from user(fast,slow).
3. Implement patterns according to given input using timer and interrupts.
4. Print feedback.

Problem Statement 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 has context menu

ALGORITHM :

1.Set interval = 5
2.Read the values for temperature(temp) and humidity(humid)from sensor at specified interval(i.e.interval).
3.Write the values of *temp* and *humid* in EEPROM or flash memory.
4.Display stored *temp* and *humid* value when requested with the help of a context menu.

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.

Pseudocode :

```
print (“Enter the first number”)

Read num1

Print (“Enter the second number “ )

Read num2

 Result = 0

Print (“Select operator (+,-,*,/)”)

if (op==’+’)

        Result = num1+num2

else if (op==’-’)

                Result = num1-num2
```

```
else if (op==’*’)
        Result = num1*num2
else
    Result = num1/num2
Print (“Result = %d”,Result)
```

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.

Factorial of a number Pseudocode:

```
num = print (“Enter the a number”)
res=1
for i in 1 to num
    res = res*i
print(“The factorial = %d”,res)
```

Using Recursion :

```
main()
    num = print (“Enter the a non-negative number”)
    print(“The factorial = %d”,fact(num))
fact(num)
    res = 1
    for i in 1 to num
        res = res*i
    return res
```

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:

1. Inputs:
2. Outputs:
3. 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 = print("Enter the minimum threshold value : ")

Moist = readfromsensor()

ctime= currentTime

Pump = OFF

If(time(ctime)<6.00 || time(ctime)>18.00), do

        Pump = OFF
```

else do :

if(Moist<Threshold), do

Pump = ON

else

Pump =OFF

Print(Moist,Pump)

FLOWCHART: