Problem statement 1: Temperature monitoring system

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 from a temperature sensor at regular intervals.
- Compare the read temperature with predefined threshold.
- If the temperature exceeds the threshold, activate an alarm(LED/Buzzer)
- Include functionality to reset the alarm.

Algorithm:

1. Define threshold value for temperature.
2. Read temperature data from temperature sensor.
3. If the read temperature > threshold temperature:
    3.1 Activate alarm
    3.2 Reset alarm
4. Repeat the steps 2 and 3 at regular intervals

Problem statement 2: Motor control system

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 LCD.

Algorithm:

1. Read desired motor speed from user using potentiometer.
2. Set the motor speed using pwm based on the input.
3. Display the current speed on LCD.
4. Repeat these steps at regular intervals.

Problem statement 3: LED blinking pattern

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(fast, slow)
- Implement different patterns using timers and interrupts.
- Provide feedbacks through an LED or serial monitor.

Algorithm:

1. Define an array of LEDs.
2. Define timer to control blink patterns.
3. Define blink patterns for each LED in the array from user.
4. Implement these blink patterns for each LED in the array using timers and interrupts.
5. Provide feedbacks.

Problem statement 4: Data Logger

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

Requirements:

- Read data from sensors at specified intervals.
- Store collected data in EEPROM or flash memory.
- Implement functionality to retrieve and display logged data.

Algorithm:

1.Read data from sensors.

2.Store it in EEPROM.

3.Increment the address

4.Repeat the above steps at regular intervals

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.

Psuedocode:

1. Enter num1

2. Enter num2

3.Choose an operation:

    1:Addition

    2:Subtraction

    3:Multiplication

    4:Division

4.Input operation

5.If operation is 1:

    result=num1+num2

    return result

6.If operation is 2:

    result=num1-num2

    return result

7.If operation is 3:

    result=num1*num2

    return result

8.If operation is 4:

    if num1==0:

        print "Error"

    else:

        result=num1/num2

        return result

Psuedocode to find factorial of a number

enter non negative integer

fact=1

for(i=1 to num),do

    fact=fact*i

    return fact

Using recursion:

Read num

Factorial(num)

    if(num==1 or num==0)

        return 1

    else:

        return Factorial(num-1)

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).

Psuedocode:

Read soil moisture from sensor

Print it

Set threshold value

Print it

Read time of day

if(soil moisture < threshold & (time > 6 AM & time < 6 PM):

    Activate pump

    print "Pump is activated"

else:

    print"Watering is not needed"

Flowchart: