Problem Statement 1: Temperature Monitoring System
Objective: Design a system to monitor temperature and trigger an alarm if a threshold is exceeded.

Steps:

Connect the temperature sensor to the microcontroller's analog or digital input.
Read the temperature sensor data at set intervals
Compare the sensor value to the predefined threshold.
If the temperature exceeds the threshold, trigger the alarm
Include a button or reset mechanism to turn off the alarm once acknowledged.
Read data using analogRead() or the appropriate method for your sensor.
Set up a loop to continuously monitor and compare values.
Use conditional statements to trigger and reset the alarm.

Problem Statement 2: Motor Control System

Objective:
Implement a system to control the speed of a DC motor based on user input.

Steps:

Use an analog input pin to read the potentiometer value and map it to a motor speed range
Use the analogWrite() function to control the motor speed
Display the current motor speed on an LCD or monitor.

Problem Statement 3: LED Blinking Pattern

Objective:
Control LEDs to blink in specific patterns based on user-defined settings.

Step:
Allows the user to choose the flashing pattern.
Use a timer or delay to control the blinking pattern.
Displays the current flicker pattern on the LCD or monitor.

Problem Statement 4: Data Logger

Objective:
Develop a data logger that collects and stores sensor data over time.

step:

Collects data from sensors at predefined intervals.
Store sensor readings in non-volatile memory
Use the functionality to retrieve and display stored data on the LCD

Problem Statement 5:

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

```
#include <stdio.h>

int main() {
   int num, factorial = 1;

     printf("Enter a non-negative integer: ");
   scanf("%d", &num);

      if (num < 0) {
      printf("Please enter a non-negative integer.\n");
   } else {
      // Calculate factorial using a loop
      for (int i = 1; i <= num; i++) {
         factorial *= i;         }

         printf("The factorial of %d is: %d\n", num, factorial);
   }

   return 0;
```

}

Problem Statement 6:

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.

```
#include <stdio.h>

int main() {
    float num1, num2, result;
    char operator;

       printf("Enter first number: ");
    scanf("%f", &num1);
    printf("Enter second number: ");
    scanf("%f", &num2);

     printf("Enter an operator (+, -, *, /): ");
    scanf(" %c", &operator);

    switch (operator) {
        case '+':
            result = num1 + num2;
            printf("%.2f + %.2f = %.2f\n", num1, num2, result);
            break;
        case '-':
            result = num1 - num2;
            printf("%.2f - %.2f = %.2f\n", num1, num2, result);
            break;
        case '*':
            result = num1 * num2;
```

```
            printf("%.2f * %.2f = %.2f\n", num1, num2, result);
            break;
        case '/':
            if (num2 != 0) {
                result = num1 / num2;
                printf("%.2f / %.2f = %.2f\n", num1, num2, result);
            } else {
                printf("Error!");
            }
            break;
        default:
            printf("Invalid operator");
    }

    return 0;
}
```

Problem Statement 7:

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 alogrithm for the above problem statement

1. Begin
2. Input the size of arrays n
3. Input ArrayX of size n
4. Input ArrayY of size n

5. For i = 0 to n-1:
   a. Set num1 = ArrayX[i]
   b. Set num2 = ArrayY[i]

   c. Calculate GCD(num1, num2):
      - While num2 != 0:
        - Set temp = num2
        - Set num2 = num1 % num2
        - Set num1 = temp

- num1 is the GCD

d. Calculate LCM(num1, num2):
- Set lcm = (num1 * num2) / GCD(num1, num2)

e. Output GCD(num1, num2) and LCM(num1, num2)

6. End

Problem Statement 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.

1. Begin
2. Input soil moisture reading
3. Input user-defined threshold
4. Input current time
5. If current_time is between 6 AM and 6 PM:
   a. If current_moisture < threshold:
      i. Activate water pump
      ii. Display "Pump Activated. Watering plants."
   b. Else:
      i. Display "Watering not needed. Soil moisture is adequate."

6. Else (if current_time is outside 6 AM to 6 PM):
   a. Display "Watering not allowed at night."

7. End

FLOWCHART

1.Start
2.Input current_moisture, threshold, current_time
3.Check if current_time is between 6 AM and 6 PM

- If Yes, proceed to step 4.
- If No, go to step 7 (Watering not allowed at night).

4.Check if current_moisture is less than threshold

- If Yes, activate the water pump, display "Pump Activated. Watering plants." and go to step 5.
- If No, display "Watering not needed. Soil moisture is adequate." and go to step 6.

5.End