IOT LAB PROGRAMES 1 TO 10

Program 1:

```
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  dht.begin();
}
void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("Sensor error");
    return;
  }
  Serial.print("Humidity: "); Serial.print(h);
  Serial.print(" %  Temp: "); Serial.print(t); Serial.println(" °C");
}
```

(https://github.com/adafruit/DHT-sensor-library.git) adafruit library

Program 2:

```
void setup() {
  pinMode(8, OUTPUT);   // Red
  pinMode(10, OUTPUT);  // Yellow
  pinMode(12, OUTPUT);  // Green
}

void loop() {
  digitalWrite(8, HIGH);   // Turn on Red for 3 sec
  delay(3000);

  digitalWrite(10, HIGH);  // Turn on Yellow for 1 sec
  delay(1000);

  digitalWrite(8, LOW);    // Turn off Red
  digitalWrite(10, LOW);   // Turn off Yellow

  digitalWrite(12, HIGH);  // Turn on Green 3 sec
  delay(3000);
  digitalWrite(12, LOW);   // Turn off Green 3 sec
  delay(500);

  // Blinking of LEDs
  digitalWrite(12, HIGH);  // Turn on Green for 0.5 sec ……(1)
  delay(500);
  digitalWrite(12, LOW);   // Turn off Green for 0.5 sec
  delay(500);

  digitalWrite(12, HIGH);  // Turn on Green for 0.5 sec ……(2)
  delay(500);
  digitalWrite(12, LOW);   // Turn off Green for 0.5 sec
  delay(500);

  digitalWrite(12, HIGH);  // Turn on Green for 0.5 sec ……(3)
  delay(500);
  digitalWrite(12, LOW);   // Turn off Green for 0.5 sec
  delay(1000);
}
```

Program3:

```
const int ledPin = 2;      // LED connected to pin 2
const int buttonPin = 4;   // Push button connected to pin 4

// Variable to store the button state
int buttonState = 0;

void setup() {
  Serial.begin(9600);           // Initialize serial communication
  pinMode(ledPin, OUTPUT);      // Set LED pin as OUTPUT
  pinMode(buttonPin, INPUT);    // Set button pin as INPUT
}

void loop() {
  // Read the state of the button
  buttonState = digitalRead(buttonPin);

  // Print button state to Serial Monitor (for debugging)
  Serial.println(buttonState);

  // Turn LED ON when button is pressed
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

Program4:

```
const int pirPin = 5; // PIR sensor connected to digital pin 5
void setup() {
  pinMode(pirPin, INPUT);
  Serial.begin(9600);
  delay(2000); // Small startup delay
  Serial.println("Motion Detector Initialized");
}

void loop() {
  if (digitalRead(pirPin) == HIGH) {
    Serial.println("Motion detected!");
    delay(1000); // Debounce delay
  } else {
    Serial.println("No motion.");
    delay(1000);
  }
}
```

Program5:

```
const int ledPin1 = 2;
const int ledPin2 = 7;
const int ledPin3 = 8;
const int buttonPin = 4;
int buttonState = 0;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, HIGH);
    digitalWrite(ledPin3, HIGH);
  } else {
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, LOW);
    digitalWrite(ledPin3, LOW);
  }
}
```

Program6:

```
#define ledPin 6
#define sensorPin A0

void setup() {
  Serial.begin(9600);
  Serial.println("Demonstration of Smoke Experiment:");
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
}

void loop() {
  int analogValue = readSensor();  // Call sensor reading function
  Serial.println("Smoke has been Detected:");
  Serial.print("Analog output: ");
  Serial.println(analogValue);
  delay(500);
}

// This function returns the analog data to the calling function
int readSensor() {
  unsigned int sensorValue = analogRead(sensorPin);  // Read analog value
  unsigned int outputValue = map(sensorValue, 0, 1023, 0, 255);  // Map 10-bit to 8-bit

  if (outputValue > 30) {
    analogWrite(ledPin, outputValue);  // Generate PWM signal
  } else {
    digitalWrite(ledPin, LOW);  // Turn off LED
  }

  return sensorValue;  // Return original analog value
}
```

Program7:

```
#define ctsPin 2  // Pin for capacitive touch sensor
#define ledPin 13 // Pin for the LED

void setup() {
  Serial.begin(9600);
  Serial.println("Demonstration of Touch Sensor Experiment!!!");
  Serial.println("Welcome to Touch Sensor");

  pinMode(ledPin, OUTPUT);
  pinMode(ctsPin, INPUT);
}

void loop() {
  int ctsValue = digitalRead(ctsPin);

  if (ctsValue == HIGH) {
    digitalWrite(ledPin, HIGH);
    Serial.println("TOUCHED");
  } else {
    digitalWrite(ledPin, LOW);
    Serial.println("Not touched");
  }

  delay(500);
}
```

Program8:

```
const int trigPin = 9;
const int echoPin = 10;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
  delay(2000);
  Serial.println("Ultrasonic Sensor Initialized");
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);
  float distance = (duration * 0.0343) / 2;
```

```
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  delay(500);
}
```

Program9:

```
const int sensorPin = A0;  // Soil moisture sensor connected to analog pin A0
const int ledPin = 6;      // LED connected to digital pin 6

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  delay(2000);
  Serial.println("Soil Moisture Sensor with LED Ready");
}

void loop() {
  int moisture = analogRead(sensorPin);
  Serial.print("Moisture Level: ");
  Serial.println(moisture);

  if (moisture < 500) {  // Adjust threshold as needed (lower = wetter)
    digitalWrite(ledPin, HIGH);  // Turn on LED (soil is dry)
  } else {
    digitalWrite(ledPin, LOW);   // Turn off LED (soil is wet)
  }

  delay(1000);
}
```

Program10:

```
const int sensorPower = 7;
const int sensorPin = A0;
const int ledPin = 6; // LED connected to pin 6

void setup() {
  pinMode(sensorPower, OUTPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(sensorPower, LOW);
  Serial.begin(9600);
  delay(2000);
  Serial.println("Water Level Sensor with LED Ready");
}

void loop() {
  digitalWrite(sensorPower, HIGH);
  delay(10);  // Allow sensor to stabilize
  int level = analogRead(sensorPin);
  digitalWrite(sensorPower, LOW);

  Serial.print("Water Level: ");
  Serial.println(level);

  if (level < 400) {  // Threshold for low water level
    digitalWrite(ledPin, HIGH); // LED ON: Low water
  } else {
    digitalWrite(ledPin, LOW);  // LED OFF: Water OK
  }

  delay(1000);
}
```