Microcontroller Lab Activities

Part A: Toolchain Setup (LED Blink)

Code

```
void setup() {
  pinMode(13, OUTPUT); // Set pin 13 as output
}
void loop() {
  digitalWrite(13, HIGH); // LED ON
  delay(1000);          // Wait 1 sec
  digitalWrite(13, LOW); // LED OFF
  delay(1000);          // Wait 1 sec
}
```

Explanation

Step-by-Step Explanation

- **pinMode(13, OUTPUT)**: Configures pin 13 as an output pin. On Arduino Uno, pin 13 is connected to the onboard LED.
- **digitalWrite(13, HIGH)**: Sends 5V (logic 1) to the pin, turning the LED ON.
- **delay(1000)**: Pauses execution for 1000 milliseconds (1 second).
- **digitalWrite(13, LOW)**: Sends 0V (logic 0), turning the LED OFF.
- The **loop()** runs infinitely, so the LED blinks ON and OFF every second.

Part B: Digital Input (Button Reading)

Code

```
int buttonPin = 2;

void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  int state = digitalRead(buttonPin);
  Serial.println(state);
  delay(200);
}
```

Explanation

Step-by-Step Explanation

- **pinMode(buttonPin, INPUT)**: Configures pin 2 as an input pin.
- **digitalRead(buttonPin)**: Reads the button state (HIGH = pressed, LOW = not pressed).
- **Serial.begin(9600)**: Starts serial communication to print values to Serial Monitor.
- **Serial.println(state)**: Prints the button state repeatedly.
- **delay(200)**: Prevents flooding the Serial Monitor, makes output readable.

Part C: Input → Output Control

Task 1: LED mirrors button state

```
int buttonPin = 2;
int ledPin = 8;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int state = digitalRead(buttonPin);
  if (state == HIGH) {
    digitalWrite(ledPin, HIGH); // LED ON
  } else {
    digitalWrite(ledPin, LOW); // LED OFF
  }
}
```

Task 2: Toggle LED with button press

```
int buttonPin = 2;
int ledPin = 8;
int ledState = LOW;
int lastButtonState = LOW;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int currentButtonState = digitalRead(buttonPin);

  if (currentButtonState == HIGH && lastButtonState == LOW) {
    ledState = !ledState;
    digitalWrite(ledPin, ledState);
    delay(200); // Debounce
  }
  lastButtonState = currentButtonState;
}
```

Explanation

Step-by-Step Explanation

- In **Task 1**, the LED turns ON only while button is pressed.
- In **Task 2**, the LED toggles state with each press (like a switch).
- A variable `ledState` remembers LED's state between presses.
- The `delay(200)` avoids bouncing (false triggers from one press).

Part D: Debugging Practice

Code with Debugging

```
int buttonPin = 2;
int ledPin = 8;

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

void loop() {
  int state = digitalRead(buttonPin);
  Serial.print("Button State: ");
  Serial.println(state);

  digitalWrite(ledPin, state); // LED mirrors button
  delay(500);
}
```

Debugging Steps

Debugging Checklist

- If LED does not light, check pin connections and LED polarity.
- If button always reads HIGH, a pull-down resistor may be missing.
- Use `Serial.print()` to display variable values for troubleshooting.
- Common errors: wrong pin number, loose wiring, missing resistor.