ejercicio #01

const int leds = 8;

int ledPins[] = {13, 12, 11, 10, 9, 8, 7, 6};

void setup() {

  for (int i = 0; i < leds; i++) {

    pinMode(ledPins[i], OUTPUT);

  }

   }

void loop() {

  secuenciaNormal();

  delay(1000);  // Esperar 1 segundo entre secuencias

  secuenciaReversa();

  delay(1000);

 }

void secuenciaNormal() {

  for (int i = 0; i < leds; i++) {

    digitalWrite(ledPins[i], HIGH);

    delay(200);

    digitalWrite(ledPins[i], LOW);

  }

}

void secuenciaReversa() {

  for (int i = leds - 1; i >= 0; i--) {

    digitalWrite(ledPins[i], HIGH);

    delay(200);

    digitalWrite(ledPins[i], LOW);

  }

}

ejercicio #2

const int buttonPin = 11;    // Pin del pulsador

const int ledPins[] = {5, 6, 7, 8, 9};  // Pines de los LEDs

const int buzzerPin = 10;    // Pin del zumbador

int currentLed = 0;          // Índice del LED actual

int delayTime = 200;         // Tiempo inicial entre encendidos en milisegundos

const int minDelayTime = 10; // Tiempo mínimo entre encendidos en milisegundos

const int delayDecrement = 20; // Decremento del tiempo en milisegundos

void setup() {

  for (int i = 0; i < 5; i++) {

    pinMode(ledPins[i], OUTPUT);

  }

  pinMode(buzzerPin, OUTPUT);

  pinMode(buttonPin, INPUT);

  randomSeed(analogRead(0)); // Inicializar la semilla aleatoria

}

void loop() {

  if (digitalRead(buttonPin) == HIGH) {

    if (currentLed == 2) {

      // El jugador ha acertado, toca el zumbador

      tone(buzzerPin, 1000, 100); // Frecuencia y duración del zumbido

      delayTime -= delayDecrement; // Aumentar velocidad

      if (delayTime < minDelayTime) {

        delayTime = 200; // Restaurar tiempo inicial si es demasiado rápido

      }

    }

    currentLed = 0; // Reiniciar la secuencia

    delay(500);    // Espero medio segundo antes de iniciar la siguiente secuencia

  }

  digitalWrite(ledPins[currentLed], HIGH); // Enciende el LED actual

  delay(delayTime);

  digitalWrite(ledPins[currentLed], LOW); // Apagar el LED actual

  currentLed = (currentLed + 1) % 5; // Avanzar al siguiente LED

}

ejercicio #3

int rojo1 = 3;

int amarillo1 = 4;

int verde1 = 5;

int rojo 2 = 6;

int amarillo2 = 7;

int verde2 = 8;

void setup() {

  pinMode(rojo1, OUTPUT);

  pinMode(amarillo1, OUTPUT);

  pinMode(verde1, OUTPUT);

  pinMode(rojo2, OUTPUT);

  pinMode(amarillo2, OUTPUT);

  pinMode(verde2, OUTPUT);

}

void loop() {

  // Fase 1: rojo 1 - verde 2 durante 3 segundos

  digitalWrite(rojo1, HIGH);

  digitalWrite(verde2, HIGH);

  delay(3000);

  // Fase 2: rojo 1 - ambar 2 (parpadea) durante 500 ms

  digitalWrite(verde2, LOW);

  for (int i = 0; i < 5; i++) {

    digitalWrite(rojo1, HIGH);

    digitalWrite(amarillo2, HIGH);

    delay(250);

    digitalWrite(rojo1, LOW);

    digitalWrite(amarillo2, LOW);

    delay(250);

  }

  // Fase 3: verde 1 - rojo 2 durante 3 segundos

  digitalWrite(amarillo2, LOW);

  digitalWrite(verde1, HIGH);

  digitalWrite(rojo2, HIGH);

  delay(3000);

  // Fase 4: ambar 1 (parpadea) - rojo 2 durante 500 ms

  digitalWrite(verde1, LOW);

  for (int i = 0; i < 5; i++) {

    digitalWrite(amarillo1, HIGH);

    digitalWrite(rojo2, HIGH);

    delay(250);

    digitalWrite(amarillo1, LOW);

    digitalWrite(rojo2, LOW);

    delay(250);

  }

}

ejercicio #4

const int leds[] = {5, 6, 7, 8, 9, 10, 11};

const int leds = sizeof(leds) / sizeof(leds[0]);

const int tiempoEncendido = 50;

const int tiempoApagado = 50;

void setup() {

  for (int i = 0; i < leds; i++) {

    pinMode(leds[i], OUTPUT);

  }

}

void loop() {

  for (int i = 0; i < leds; i++) {

    digitalWrite(leds[i], HIGH);

    delay(tiempoEncendido);

    digitalWrite(leds[i], LOW);

    delay(tiempoApagado);

  }

  for (int i = leds - 1; i >= 0; i--) {

    digitalWrite(leds[i], HIGH);

    delay(tiempoEncendido);

    digitalWrite(leds[i], LOW);

    delay(tiempoApagado);

  }

}