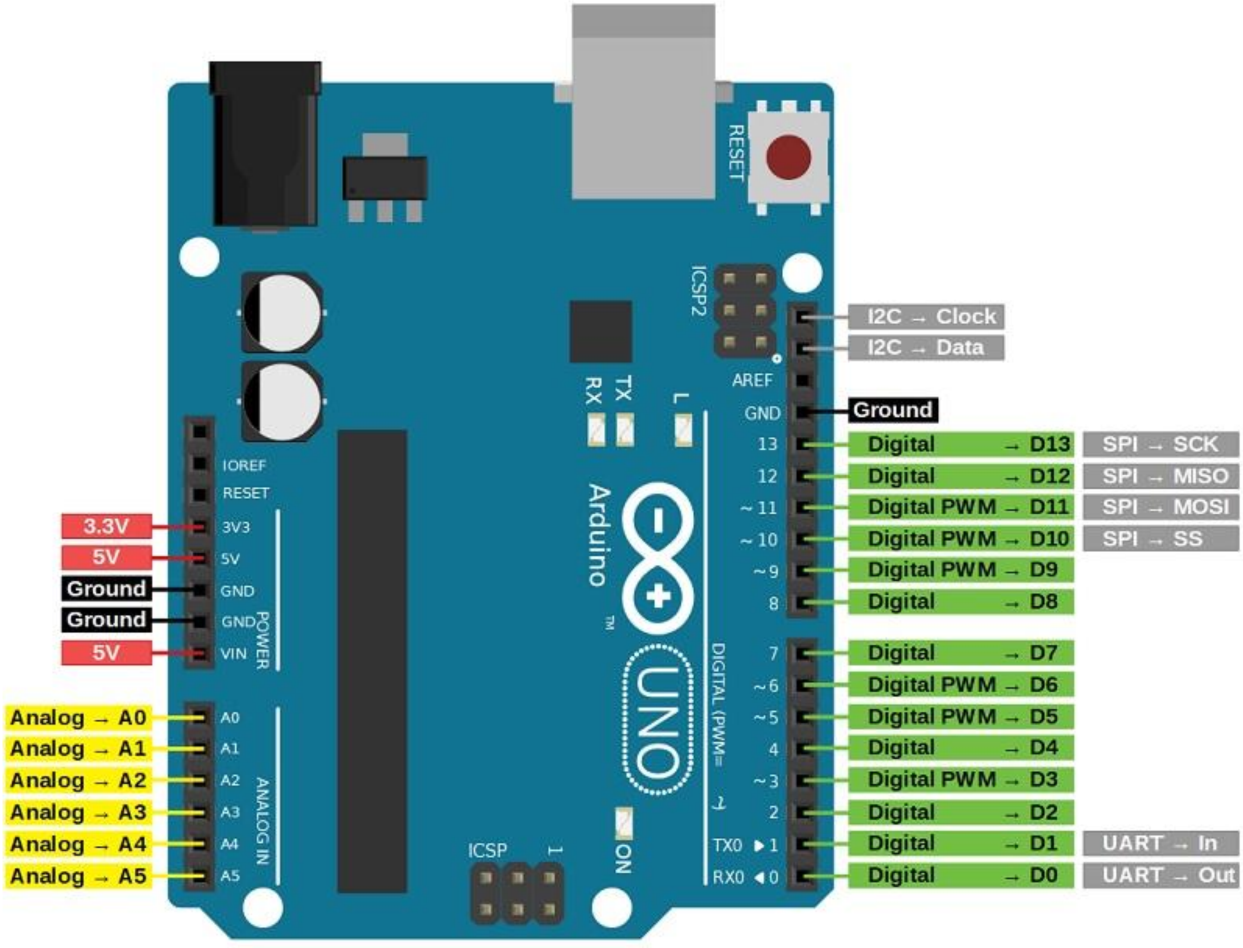
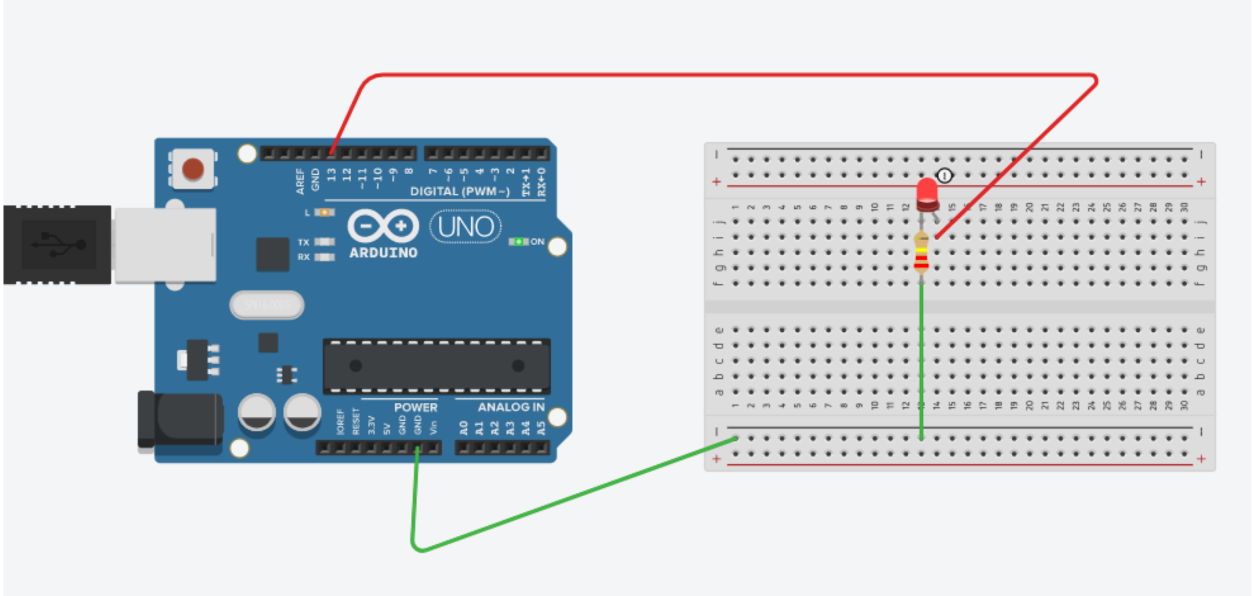
To light an LED using an Arduino (Tinkercad)





Components Needed:

1. Arduino board (e.g., Arduino Uno)
2. LED
3. Resistor (typically 220 ohms or 330 ohms)
4. Breadboard (optional, but helpful for prototyping)
5. Jumper wires

Steps:

1. **Connect the LED to the Arduino:**
   - **Long leg (Anode, positive):** Connect this to a digital pin on the Arduino (e.g., pin 13).
   - **Short leg (Cathode, negative):** Connect this to one end of the resistor.
   - **Other end of the resistor:** Connect this to the Arduino's GND (ground) pin.
2. **Write the Arduino Code:**

   Here's a simple sketch to blink the LED:

```
// Pin connected to the LED
const int ledPin = 13; // Change this if you're using a different pin

void setup() {
  // Set the LED pin as an OUTPUT
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Turn the LED on
  digitalWrite(ledPin, HIGH);
  // Wait for 1000 milliseconds (1 second)
  delay(1000);
  // Turn the LED off
  digitalWrite(ledPin, LOW);
  // Wait for 1000 milliseconds (1 second)
  delay(1000);
}
```

3. **Upload the Code:**
   - Connect your Arduino to your computer using a USB cable.
   - Open the Arduino IDE on your computer.
   - Copy and paste the code above into the IDE.
   - Select the appropriate board and port from the "Tools" menu.
   - Click the upload button (right arrow) to upload the code to your Arduino.
4. **Observe the LED:**
   - The LED should start blinking on and off with a one-second interval.

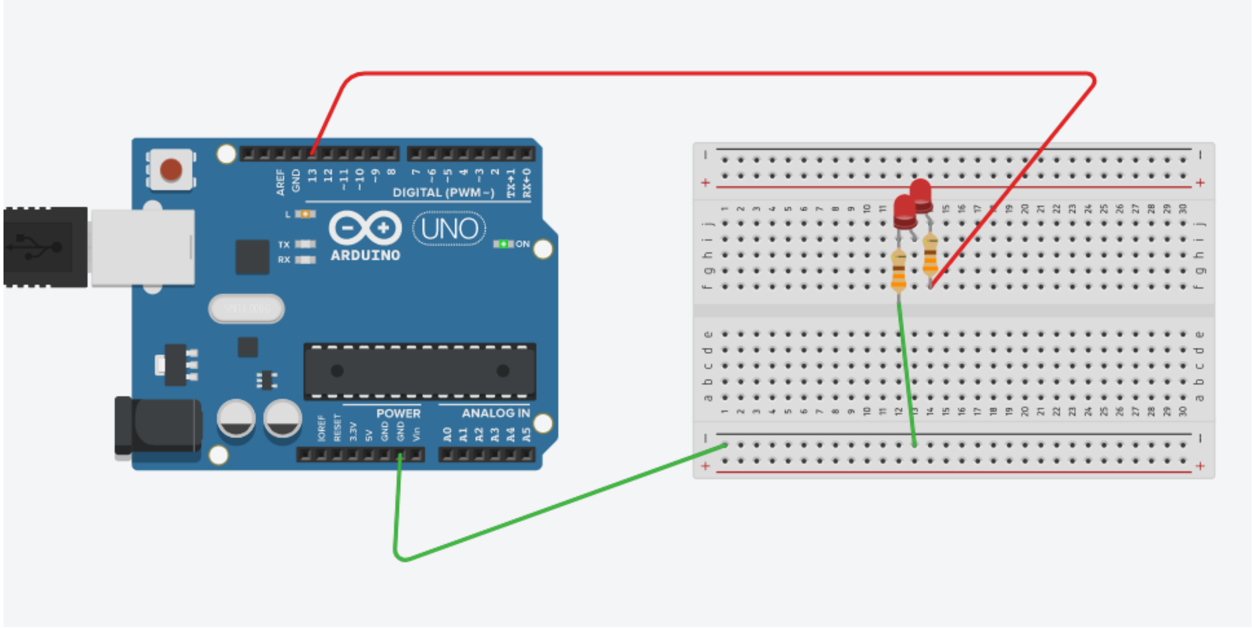
Additional Notes:

- **Resistor Value:** The resistor limits the current through the LED to prevent it from burning out. For most standard LEDs, a 220-ohm or 330-ohm resistor works well.
- **Breadboard Use:** If you're using a breadboard, you can place the LED and resistor on it and use jumper wires to connect them to the Arduino.

Python program for raspberry pi – refer python programs pdf

Lighting two LEDs connected in series using an Arduino

Lighting two LEDs connected in series using an Arduino is a bit unconventional (little different from what is usual or traditional), as LEDs in series require a different approach than individual LEDs. Here’s how you can do it:



Components Needed:

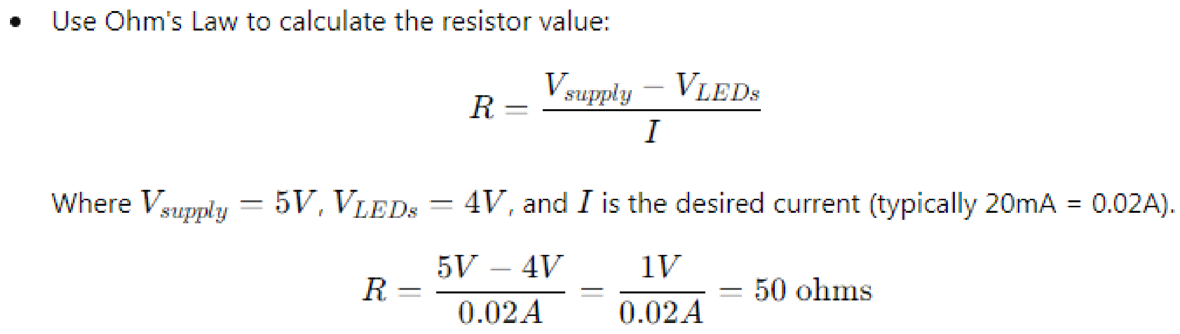
1. Arduino board (e.g., Arduino Uno)
2. Two LEDs
3. Two resistors (typically 220 ohms or 330 ohms each)
4. Breadboard
5. Jumper wires

Circuit Design:

When LEDs are connected in series, they share the same current, so you need to account for the total voltage drop across both LEDs and ensure the resistors are appropriately sized.

Steps:

1. **Connect the LEDs and Resistors:**
   - **Connect the LEDs in Series:**
     - Connect the anode (long leg) of the first LED to the digital pin on the Arduino (e.g., pin 13).
     - Connect the cathode (short leg) of the first LED to the anode (long leg) of the second LED.
     - Connect the cathode (short leg) of the second LED to one end of a resistor.
     - Connect the other end of the resistor to the Arduino's GND (ground) pin.
   - **Add Resistor:**
     - Place a resistor between the anode of the first LED and the Arduino pin. This resistor will limit the current through both LEDs.
2. **Determine the Resistor Value:**
   - The total forward voltage drop of the two LEDs needs to be considered. For standard LEDs, the forward voltage is about 2V each (total of 4V). Assume the Arduino output voltage is 5V.
   - Use Ohm's Law to calculate the resistor value:



- Use Ohm's Law to calculate the resistor value:

$$R = \frac{V_{supply} - V_{LEDs}}{I}$$

Where $V_{supply} = 5V$, $V_{LEDs} = 4V$, and I is the desired current (typically 20mA = 0.02A).

$$R = \frac{5V - 4V}{0.02A} = \frac{1V}{0.02A} = 50 \text{ ohms}$$

   - Use a 50-ohm resistor or the closest standard value (e.g., 47 ohms or 56 ohms). If you use a 220-ohm resistor, the current will be lower but still acceptable for the LEDs.
3. **Write the Arduino Code:**

Here’s a simple sketch to blink both LEDs:

```
// Pin connected to the LEDs
```

```
const int ledPin = 13; // Change this if you're using a different pin

void setup() {
  // Set the LED pin as an OUTPUT
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Turn the LEDs on
  digitalWrite(ledPin, HIGH);
  // Wait for 1000 milliseconds (1 second)
  delay(1000);
  // Turn the LEDs off
  digitalWrite(ledPin, LOW);
  // Wait for 1000 milliseconds (1 second)
  delay(1000);
}
```

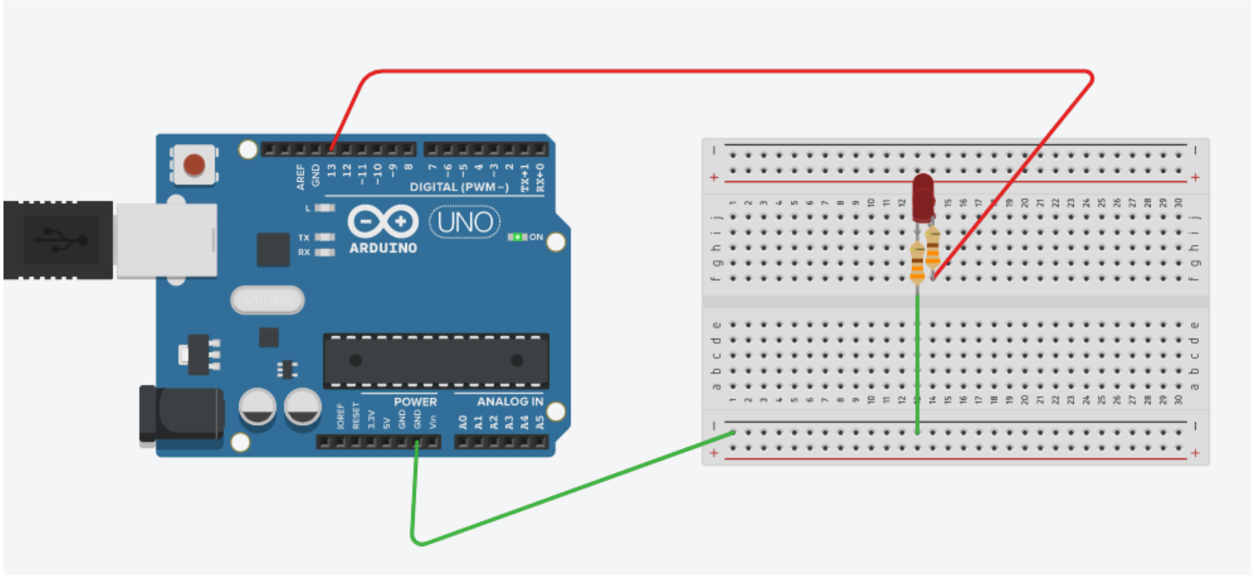
4. **Upload the Code:**
   - Connect your Arduino to your computer using a USB cable.
   - Open the Arduino IDE on your computer.
   - Copy and paste the code above into the IDE.
   - Select the appropriate board and port from the "Tools" menu.
   - Click the upload button (right arrow) to upload the code to your Arduino.
5. **Observe the LEDs:**
   - Both LEDs should light up simultaneously and blink on and off in unison.

Additional Notes:

- **Power Considerations:** Ensure the total voltage drop across the series LEDs plus the resistor is within the Arduino’s output voltage capability.
- **Resistor Sizing:** Double-check the current limit and resistor value to avoid damaging the LEDs or the Arduino.

Lighting two LEDs connected in parallel using an Arduino

Lighting two LEDs connected in parallel using an Arduino is straightforward and quite common. In this setup, each LED will have its own current-limiting resistor, and both will be connected to the same digital pin on the Arduino. Here’s a step-by-step guide:



Components Needed:

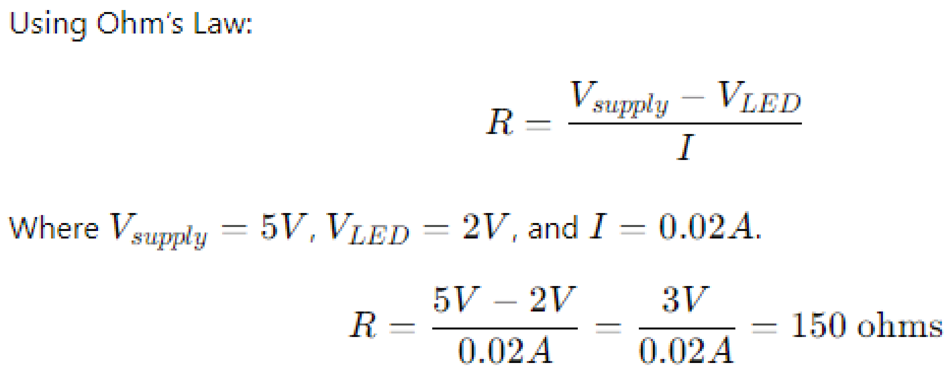
1. Arduino board (e.g., Arduino Uno)
2. Two LEDs
3. Two resistors (typically 220 ohms or 330 ohms each, depending on the LED specifications)
4. Breadboard
5. Jumper wires

Steps:

1. **Connect the LEDs and Resistors:**
    - **Connect the Anodes (Positive Legs) of Both LEDs:**
        - Connect the long leg (anode) of the first LED to the digital pin on the Arduino (e.g., pin 13).

  - Connect the long leg (anode) of the second LED to the same digital pin (e.g., pin 13).
- **Connect the Cathodes (Negative Legs) of Both LEDs:**
  - Connect the short leg (cathode) of the first LED to one end of a resistor.
  - Connect the short leg (cathode) of the second LED to the other end of a resistor.
- **Connect the Resistors to Ground:**
  - Connect the other end of the resistor from the first LED to the Arduino's GND (ground) pin.
  - Connect the other end of the resistor from the second LED to the Arduino's GND (ground) pin.

2. **Determine the Resistor Value:**
   - To calculate the appropriate resistor value, consider the forward voltage of the LEDs and the desired current. Assume each LED has a forward voltage of 2V and you want a current of 20mA (0.02A). The Arduino output voltage is 5V.



Using Ohm's Law:

$$R = \frac{V_{supply} - V_{LED}}{I}$$

Where $V_{supply} = 5V$, $V_{LED} = 2V$, and $I = 0.02A$.

$$R = \frac{5V - 2V}{0.02A} = \frac{3V}{0.02A} = 150 \text{ ohms}$$

   - Use a 150-ohm resistor for each LED, or the closest standard value like 220 ohms or 330 ohms if you prefer a lower current.

3. **Write the Arduino Code:**

Here's a simple sketch to blink both LEDs:

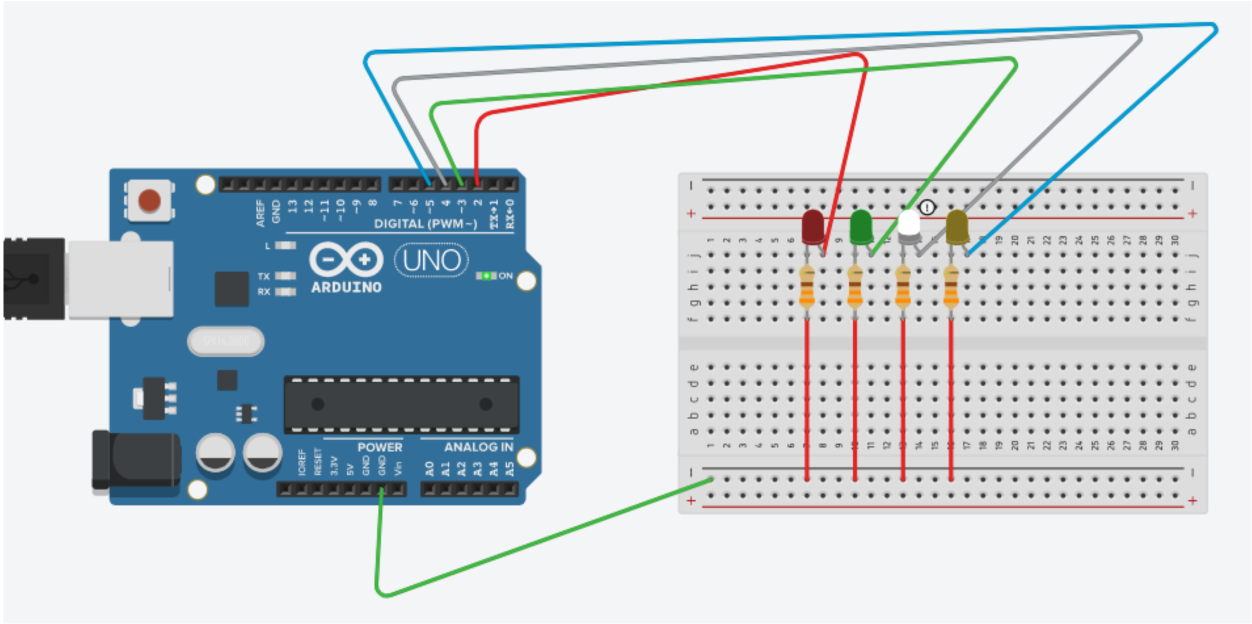
```
// Pin connected to the LEDs
const int ledPin = 13; // Change this if you're using a different pin

void setup() {
  // Set the LED pin as an OUTPUT
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Turn the LEDs on
  digitalWrite(ledPin, HIGH);
  // Wait for 1000 milliseconds (1 second)
  delay(1000);
  // Turn the LEDs off
  digitalWrite(ledPin, LOW);
  // Wait for 1000 milliseconds (1 second)
  delay(1000);
}
```

4. **Upload the Code:**
    - Connect your Arduino to your computer using a USB cable.
    - Open the Arduino IDE on your computer.
    - Copy and paste the code above into the IDE.
    - Select the appropriate board and port from the "Tools" menu.
    - Click the upload button (right arrow) to upload the code to your Arduino.
5. **Observe the LEDs:**
    - Both LEDs should light up simultaneously and blink on and off in unison.

Additional Notes:

- **Resistor Value:** Using the same resistor value for both LEDs ensures they receive the same current. If you use different values, one LED might be brighter than the other.
- **Current Limiting:** Each LED needs its own current-limiting resistor to prevent excessive current and potential damage.

Design a program to infinitely blink a sequence of 4 LEDs connected to arduino uno, one after the other with the delay of 500ms



To create a program that infinitely blinks a sequence of 4 LEDs connected to an Arduino Uno, turning them on one after the other with a delay of 500 milliseconds, you need to follow these steps:

1. **Connect the LEDs to the Arduino:**
   - Connect each LED to a separate digital pin on the Arduino.
   - Add current-limiting resistors (typically 220Ω or 330Ω) in series with each LED to prevent them from drawing too much current.
2. **Write the Arduino Code:**

Here’s a simple sketch to achieve this:

```
// Define the pins for the LEDs
const int ledPins[] = {2, 3, 4, 5}; // Pins where LEDs are connected
const int numLeds = 4; // Number of LEDs
const int delayTime = 500; // Delay time in milliseconds

void setup() {
  // Initialize each LED pin as an OUTPUT
  for (int i = 0; i < numLeds; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
}

void loop() {
  // Turn each LED on and off in sequence
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH); // Turn LED on
    delay(delayTime);               // Wait for 500 milliseconds
    digitalWrite(ledPins[i], LOW);  // Turn LED off
    delay(delayTime);               // Wait for 500 milliseconds
  }
}
```

Explanation

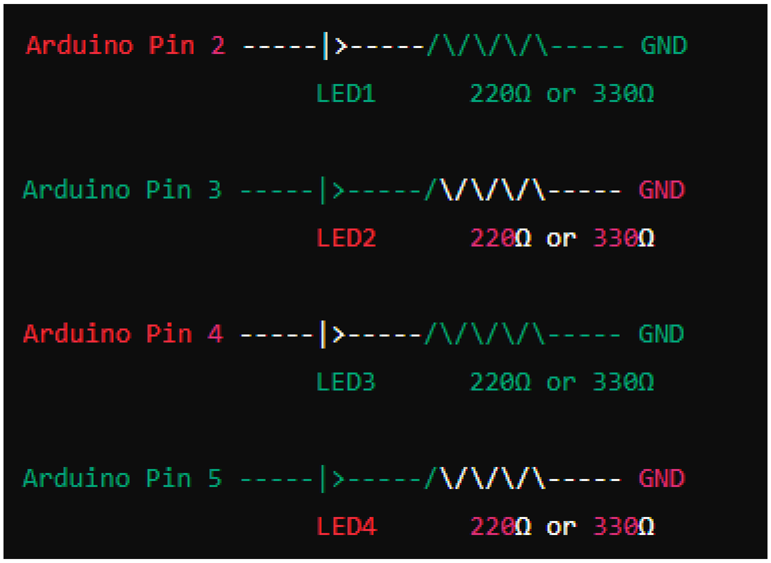
1. **Pin Definitions:**
   - const int ledPins[] = {2, 3, 4, 5}; defines the digital pins to which the LEDs are connected. You can change these pin numbers according to your actual connections.
2. **Setup Function:**
   - pinMode(ledPins[i], OUTPUT); sets each LED pin as an output.
3. **Loop Function:**
   - The for loop iterates through each LED pin, turning the LED on (digitalWrite(ledPins[i], HIGH);), waiting for 500 milliseconds, turning the LED off (digitalWrite(ledPins[i], LOW);), and waiting again for 500 milliseconds.
4. **Delay Time:**
   - const int delayTime = 500; sets the delay between turning LEDs on and off to 500 milliseconds.

Connections

- **LED1:** Connect the anode (long leg) to pin 2, cathode to ground via a resistor.
- **LED2:** Connect the anode to pin 3, cathode to ground via a resistor.
- **LED3:** Connect the anode to pin 4, cathode to ground via a resistor.
- **LED4:** Connect the anode to pin 5, cathode to ground via a resistor.

Circuit Diagram

Here’s a simplified view of the connections:



```
Arduino Pin 2 -----|>-----/\/\/\/\----- GND
                   LED1      220Ω or 330Ω

Arduino Pin 3 -----|>-----/\/\/\/\----- GND
                   LED2      220Ω or 330Ω

Arduino Pin 4 -----|>-----/\/\/\/\----- GND
                   LED3      220Ω or 330Ω

Arduino Pin 5 -----|>-----/\/\/\/\----- GND
                   LED4      220Ω or 330Ω
```

This code will continuously cycle through turning each LED on for 500 milliseconds, then off for 500 milliseconds, creating a sequential blinking effect. Adjust the pin numbers and delays as needed for your specific setup.