Develop a program to blink 5 LEDs back and forth.

```
// LED Knight Rider Effect with 5 LEDs
int leds[] = {2, 3, 4, 5, 6};   // Pins for LEDs
int numLeds = 5;
int delayTime = 150;          // Delay between shifts (ms)

void setup() {
  // Set all pins as OUTPUT
  for (int i = 0; i < numLeds; i++) {
    pinMode(leds[i], OUTPUT);
  }
}

void loop() {
  // Move LED from left to right
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(leds[i], HIGH);
    delay(delayTime);
    digitalWrite(leds[i], LOW);
  }

  // Move LED from right to left
  for (int i = numLeds - 2; i > 0; i--) {
    digitalWrite(leds[i], HIGH);
    delay(delayTime);
    digitalWrite(leds[i], LOW);
  }
}
```

How It Works

- `leds[]` holds the pin numbers for the LEDs.
- First `for` loop lights LEDs **left → right**.
- Second `for` loop lights LEDs **right → left** (skipping the edges to avoid double-blinking).
- Delay controls the speed of the effect.

Develop a program to interface a relay with Arduino board.

Hardware Connections

- **Relay module IN pin** → Arduino **pin 7**
- **Relay module VCC** → Arduino **5V**
- **Relay module GND** → Arduino **GND**
- The load (lamp, fan, etc.) connects to the **relay's NO/COM contacts**.

Be careful if you are using AC mains **— use proper insulation and do not touch live wires.**

```
// Relay Interfacing with Arduino

int relayPin = 7;  // Relay control pin connected to digital pin 7

void setup() {
  pinMode(relayPin, OUTPUT);   // Set relay pin as OUTPUT
  digitalWrite(relayPin, LOW); // Keep relay OFF initially
}

void loop() {
  digitalWrite(relayPin, HIGH);  // Turn relay ON
  delay(2000);                   // Wait 2 seconds

  digitalWrite(relayPin, LOW);   // Turn relay OFF
  delay(2000);                   // Wait 2 seconds
}
```

How It Works

- **HIGH** energizes the relay coil → switches ON the connected load.
- **LOW** de-energizes the coil → switches OFF the load.
- In this example, the relay toggles every **2 seconds**.

Arduino Code: Relay Controlled by Push Button

Hardware Connections

- **Relay module IN pin** → Arduino digital pin **7**
- **Push button** → Arduino digital pin **2** (use pull-down resistor or internal pull-up)
- **Relay VCC/GND** → Arduino 5V/GND
- Load connected to relay (**be careful with AC loads**).

```
// Relay control with Push Button

int relayPin = 7;     // Relay control pin
int buttonPin = 2;    // Push button pin
int buttonState = 0;  // Variable to store button state

void setup() {
  pinMode(relayPin, OUTPUT);       // Relay as OUTPUT
  pinMode(buttonPin, INPUT_PULLUP); // Button as INPUT with internal pull-up
  digitalWrite(relayPin, LOW);     // Relay OFF initially
}

void loop() {
  // Read the button state (LOW when pressed)
  buttonState = digitalRead(buttonPin);

  if (buttonState == LOW) {
    digitalWrite(relayPin, HIGH);  // Turn relay ON
  } else {
    digitalWrite(relayPin, LOW);   // Turn relay OFF
  }
}
```

How It Works

- The button is connected with **internal pull-up** (`INPUT_PULLUP`).
- When the button is **pressed**, the pin reads `LOW` → Relay turns **ON**.
- When the button is **released**, the pin reads `HIGH` → Relay turns **OFF**.