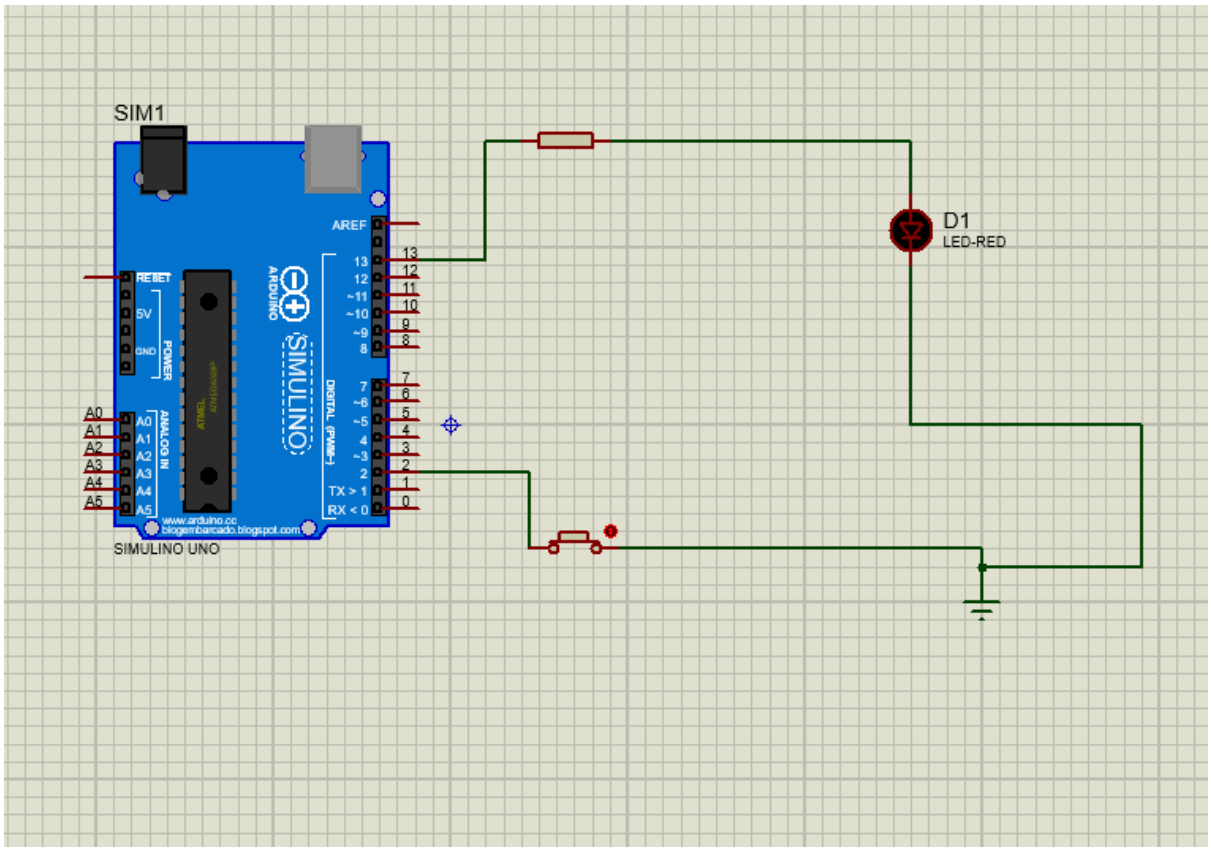
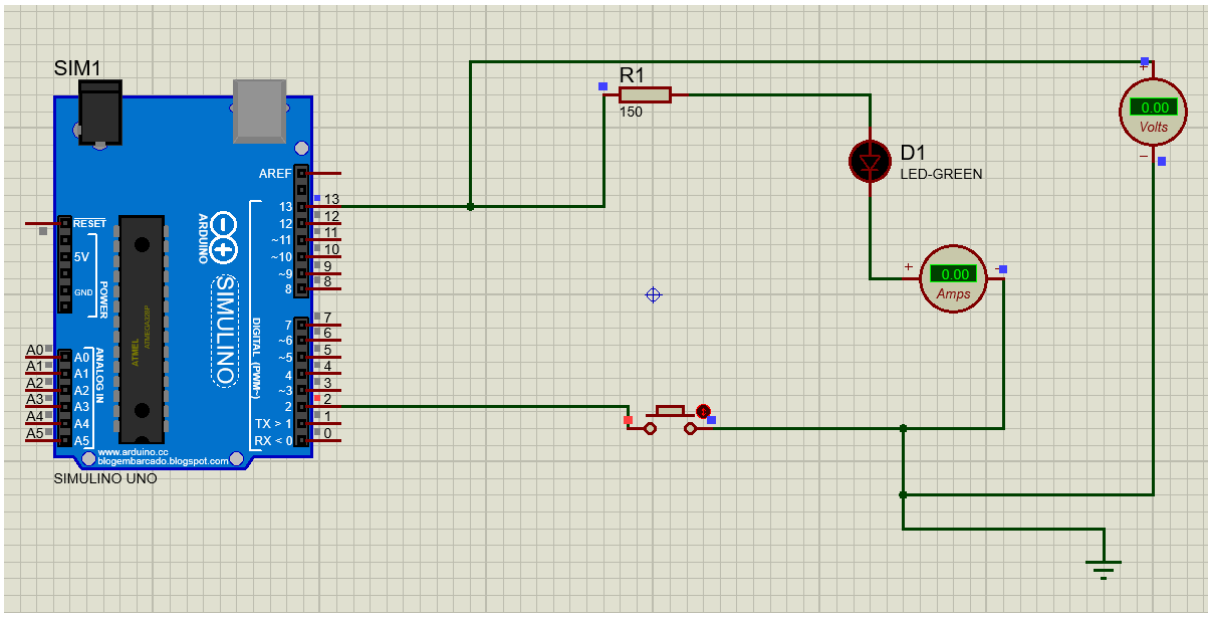
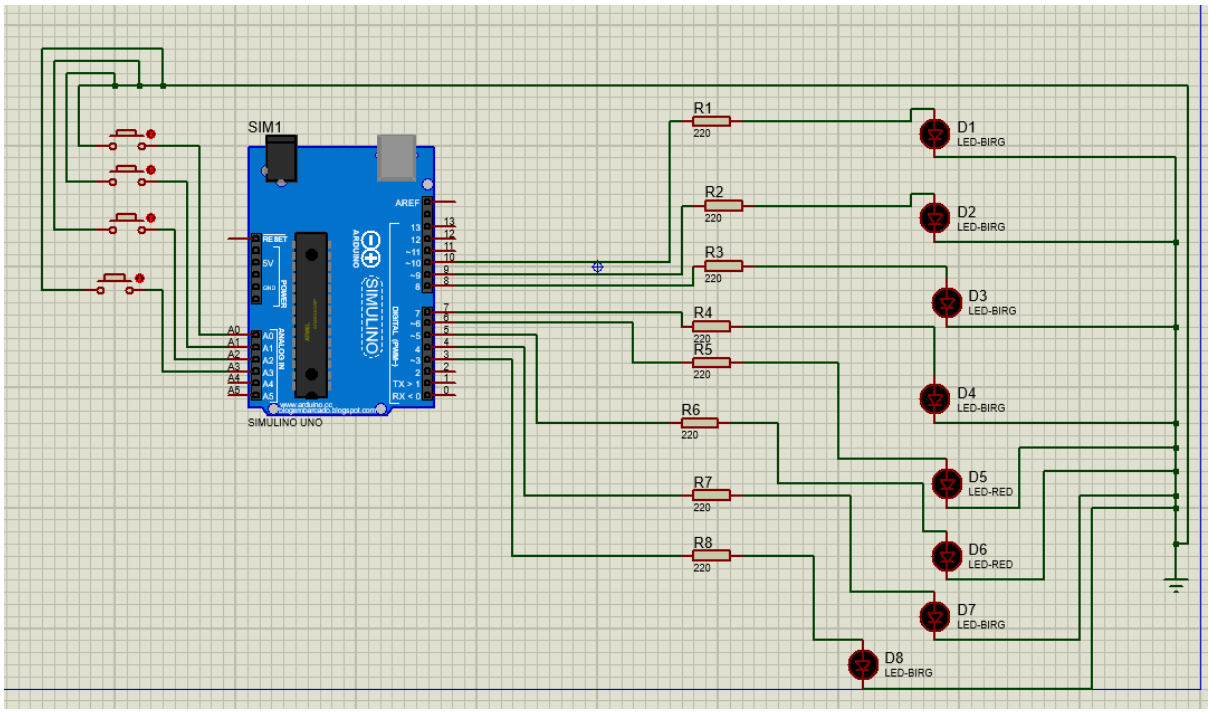
Bài 1:


SIM1
D1
LED-RED
SIMULINO UNO

Bài 2:


SIM1
R1
150
D1
LED-GREEN
0.00
Volts
0.00
Amps
SIMULINO UNO

Bài 3:




```
// ====== Cấu hình phần cứng (theo mạch của bạn) ======
// Trên hình: LED1 ở trên cùng lấy từ D10, rồi D9, D8, ... đến D3 ở dưới cùng
const byte ledPins[8] = {10, 9, 8, 7, 6, 5, 4, 3};  // LED1..LED8

const byte BTN_ALT    = A0;  // chớp tắt xen kẽ
const byte BTN_L2R    = A1;  // trái -> phải
const byte BTN_R2L    = A2;  // phải -> trái
const byte BTN_BOUNCE = A3;  // trái -> phải -> trái

// ====== Tham số hiệu ứng ======
unsigned long interval   = 150;   // ms giữa các bước
unsigned long lastStepMs = 0;

// ====== Chống dội & đọc cạnh xuống ======
struct Btn {
```

```
  byte pin;
  bool lastStable;
  bool lastRead;
  unsigned long lastChange;
};

Btn btns[4] = {
  {BTN_ALT,    HIGH, HIGH, 0},
  {BTN_L2R,    HIGH, HIGH, 0},
  {BTN_R2L,    HIGH, HIGH, 0},
  {BTN_BOUNCE, HIGH, HIGH, 0}
};

const unsigned long DEBOUNCE_MS = 40;

// ====== Trạng thái hệ thống ======
enum Mode : byte { IDLE=0, ALT, L2R, R2L, BOUNCE };
Mode mode = IDLE;

byte indexPos = 0;     // vị trí LED hiện tại (0..7)
int  dir = +1;         // hướng cho BOUNCE
bool phase = false;    // pha cho xen kẽ

// ====== Tiện ích hiển thị ======
void allOff() {
  for (byte i=0;i<8;i++) digitalWrite(ledPins[i], LOW);
```

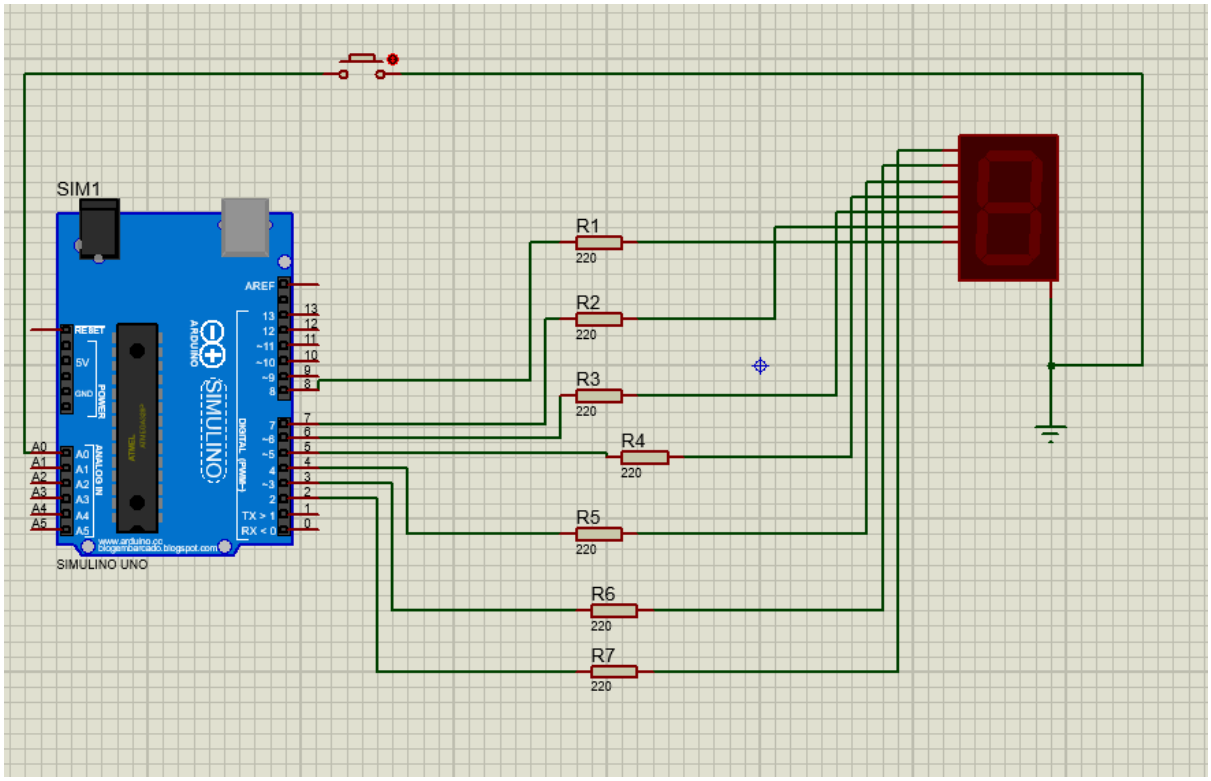
```
}

void showSingle(byte idx) {
  for (byte i=0;i<8;i++) digitalWrite(ledPins[i], (i==idx)? HIGH : LOW);
}

void showAlternate(bool ph) {
  for (byte i=0;i<8;i++) {
    digitalWrite(ledPins[i], ((i % 2) == (ph ? 1 : 0)) ? HIGH : LOW);
  }
}

// Đọc nút cạnh xuống (FALLING) có chống dội
bool buttonFalling(Btn &b) {
  bool raw = digitalRead(b.pin);
  unsigned long now = millis();

  if (raw != b.lastRead) {      // vừa đổi trạng thái thô
    b.lastRead = raw;
    b.lastChange = now;
  }

  if ((now - b.lastChange) > DEBOUNCE_MS && raw != b.lastStable) {
    b.lastStable = raw;         // trạng thái ổn định mới
    if (raw == LOW) return true;  // nhấn (HIGH->LOW)
  }
```

```
  return false;
}

// ====== Khởi tạo ======
void setup() {
  for (byte i=0;i<8;i++) {
    pinMode(ledPins[i], OUTPUT);
    digitalWrite(ledPins[i], LOW);
  }
  pinMode(BTN_ALT,    INPUT_PULLUP);
  pinMode(BTN_L2R,    INPUT_PULLUP);
  pinMode(BTN_R2L,    INPUT_PULLUP);
  pinMode(BTN_BOUNCE, INPUT_PULLUP);

  allOff();
}

// ====== Vòng lặp chính ======
void loop() {
  // Bật/tắt từng chế độ bằng nút
  if (buttonFalling(btns[0])) {  // ALT
    mode = (mode == ALT) ? IDLE : ALT;
    phase = false;
  }
  if (buttonFalling(btns[1])) {  // L2R
    mode = (mode == L2R) ? IDLE : L2R;
```

```
  indexPos = 0;
}
if (buttonFalling(btns[2])) {  // R2L
  mode = (mode == R2L) ? IDLE : R2L;
  indexPos = 7;
}
if (buttonFalling(btns[3])) {  // BOUNCE
  mode = (mode == BOUNCE) ? IDLE : BOUNCE;
  indexPos = 0; dir = +1;
}

// Cập nhật hiệu ứng theo thời gian
unsigned long now = millis();
if (now - lastStepMs >= interval) {
  lastStepMs = now;

  switch (mode) {
    case IDLE:
      allOff();
      break;

    case ALT:
      phase = !phase;
      showAlternate(phase);
      break;
```

```
      case L2R:
        showSingle(indexPos);
        indexPos = (indexPos + 1) % 8;
        break;

      case R2L:
        showSingle(indexPos);
        indexPos = (indexPos == 0) ? 7 : indexPos - 1;
        break;

      case BOUNCE:
        showSingle(indexPos);
        if (indexPos == 7) dir = -1;
        else if (indexPos == 0) dir = +1;
        indexPos += dir;
        break;
    }
  }
}
```

Bài 4:

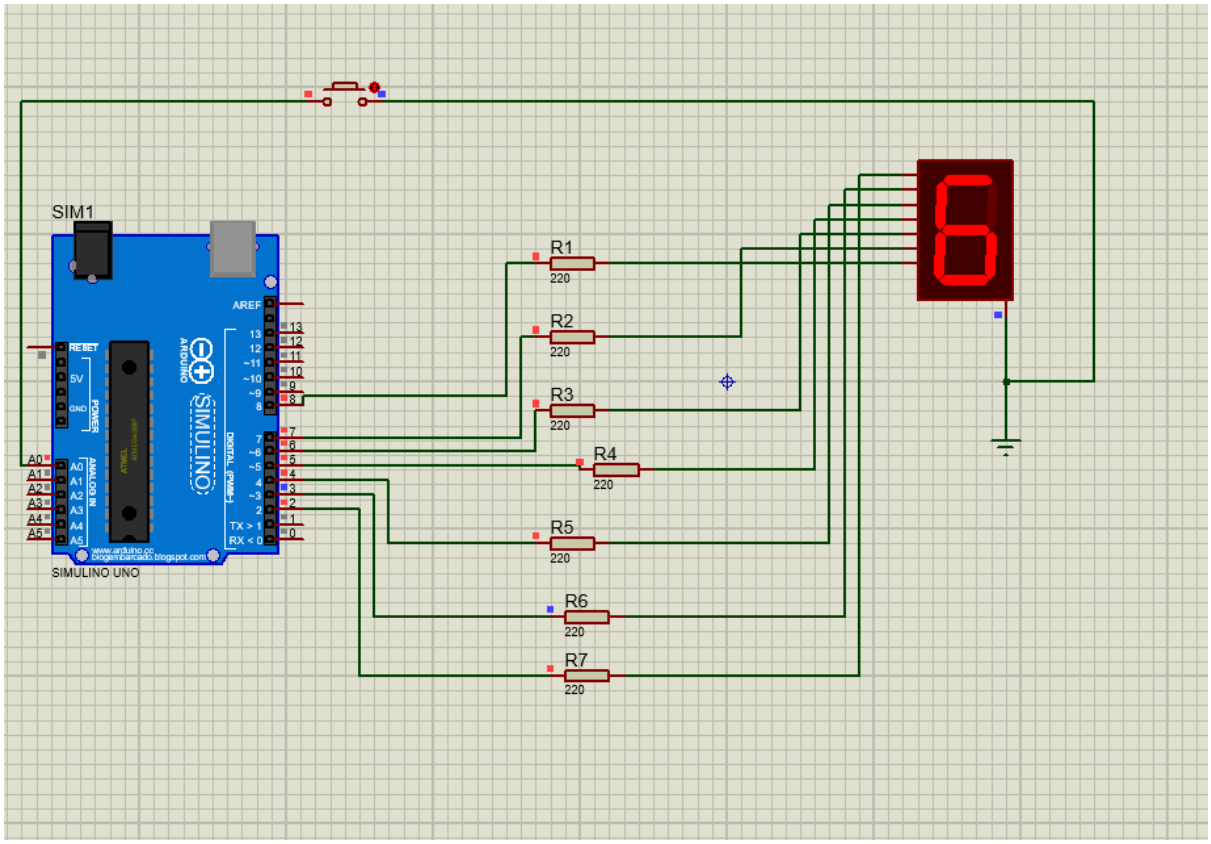



```
// ======= Sơ đồ chân theo hình của bạn =======
// D2->a, D3->b, D4->c, D5->d, D6->e, D7->f, D8->g
// COM của LED 7 đoạn nối GND (Common Cathode)
// Button: A0 <-> GND (INPUT_PULLUP)

const byte segPins[7] = {2, 3, 4, 5, 6, 7, 8}; // a,b,c,d,e,f,g
const byte btnPin     = A0;

// Common Cathode (COM xuống GND)
const byte SEG_ON  = HIGH;
const byte SEG_OFF = LOW;

// Nếu bạn dùng Common Anode (COM lên +5V) thì đổi 2 dòng trên:
// const byte SEG_ON  = LOW;
```

```
// const byte SEG_OFF = HIGH;

// Bảng a..g cho các số 0..9 (1=bật, 0=tắt) theo CC
const byte digits[10][7] = {
  /*0*/ {1,1,1,1,1,1,0},
  /*1*/ {0,1,1,0,0,0,0},
  /*2*/ {1,1,0,1,1,0,1},
  /*3*/ {1,1,1,1,0,0,1},
  /*4*/ {0,1,1,0,0,1,1},
  /*5*/ {1,0,1,1,0,1,1},
  /*6*/ {1,0,1,1,1,1,1},
  /*7*/ {1,1,1,0,0,0,0},
  /*8*/ {1,1,1,1,1,1,1},
  /*9*/ {1,1,1,1,0,1,1}
};

unsigned long lastStep = 0;
const unsigned long STEP_MS = 500;   // tốc độ đếm
int cur = 0;                         // số đang hiển thị (0..9)

void showDigit(int n) {
  for (byte i = 0; i < 7; i++)
    digitalWrite(segPins[i], digits[n][i] ? SEG_ON : SEG_OFF);
}

void setup() {
```

```
  for (byte i = 0; i < 7; i++) {
    pinMode(segPins[i], OUTPUT);
    digitalWrite(segPins[i], SEG_OFF);
  }
  pinMode(btnPin, INPUT_PULLUP);      // nhả=HIGH, nhấn=LOW
  showDigit(cur);
}

void loop() {
  bool pressed = (digitalRead(btnPin) == LOW);  // giữ nút = đếm ngược

  if (millis() - lastStep >= STEP_MS) {
    lastStep = millis();

    if (pressed) {                // đếm ngược khi bấm
      cur = (cur == 0) ? 9 : cur - 1;
    } else {                      // đếm xuôi khi thả
      cur = (cur == 9) ? 0 : cur + 1;
    }
    showDigit(cur);
  }
}
```

Bài 5:




```
// ===== Chân theo sơ đồ =====
// D2->a, D3->b, D4->c, D5->d, D6->e, D7->f, D8->g
// A0: button -> GND (INPUT_PULLUP)

const byte segPins[7] = {2, 3, 4, 5, 6, 7, 8}; // a..g
const byte btnPin     = A0;

// Common Cathode (COM xuống GND)
const byte SEG_ON  = HIGH;
const byte SEG_OFF = LOW;

// Nếu dùng Common Anode (COM lên +5V), dùng 2 dòng dưới:
```

```
// const byte SEG_ON  = LOW;
// const byte SEG_OFF = HIGH;

// Bảng a..g cho 0..9 (1=bật, 0=tắt) theo CC
const byte digits[10][7] = {
  /*0*/ {1,1,1,1,1,1,0},
  /*1*/ {0,1,1,0,0,0,0},
  /*2*/ {1,1,0,1,1,0,1},
  /*3*/ {1,1,1,1,0,0,1},
  /*4*/ {0,1,1,0,0,1,1},
  /*5*/ {1,0,1,1,0,1,1},
  /*6*/ {1,0,1,1,1,1,1},
  /*7*/ {1,1,1,0,0,0,0},
  /*8*/ {1,1,1,1,1,1,1},
  /*9*/ {1,1,1,1,0,1,1}
};

void showDigit(byte n) {
  for (byte i = 0; i < 7; i++)
    digitalWrite(segPins[i], digits[n][i] ? SEG_ON : SEG_OFF);
}

// Biến đếm
byte countPress = 0;

// Chống dội + đọc cạnh xuống
```

```
bool lastStable = HIGH, lastRead = HIGH;
unsigned long lastChange = 0;
const unsigned long DEBOUNCE_MS = 40;

void setup() {
  for (byte i = 0; i < 7; i++) {
    pinMode(segPins[i], OUTPUT);
    digitalWrite(segPins[i], SEG_OFF);
  }
  pinMode(btnPin, INPUT_PULLUP);
  showDigit(countPress); // hiển thị 0 lúc khởi động
}

void loop() {
  bool raw = digitalRead(btnPin);          // nhả=HIGH, nhấn=LOW
  unsigned long now = millis();

  // theo dõi thay đổi thô
  if (raw != lastRead) { lastRead = raw; lastChange = now; }

  // ổn định > DEBOUNCE_MS và khác trạng thái ổn định trước đó?
  if ((now - lastChange) > DEBOUNCE_MS && raw != lastStable) {
    lastStable = raw;
    if (raw == LOW) {                      // cạnh xuống: vừa nhấn
      countPress = (countPress + 1) % 10;  // 0..9 rồi quay 0
      showDigit(countPress);
```

```
    }
  }
}
```