Q1. Turn LEDs ON/OFF: Write a program to control 4 LEDs connected to GPIO pins. Implement a function void controlLED( ) that turns the specified LED ON (true) or OFF (false)

```
#include "stm32f407xx.h"

// Delay function
void delay() {
        for (uint32_t i = 0; i < 500000; i++);
}

// Control LED ON/OFF
void controlLED(uint8_t PinNumber, bool state) {
        if (state) {
        GPIO_WriteToOutputPin(GPIOD, PinNumber, GPIO_PIN_SET);
        } else {
        GPIO_WriteToOutputPin(GPIOD, PinNumber, GPIO_PIN_RESET);
        }
}

int main(void) {
        GPIO_Handle_t GPIOLed;
        GPIOLed.pGPIOx = GPIOD;

        // Initialize all LEDs
        for (uint8_t i = 0; i < 4; i++) {
        GPIOLed.GPIO_PinConfig.GPIO_PinNumber = GPIO_PIN_NO_12 + i;
        GPIOLed.GPIO_PinConfig.GPIO_PinMode = GPIO_MODE_OUT;
        GPIOLed.GPIO_PinConfig.GPIO_PinSpeed = GPIO_SPEED_FAST;
        GPIOLed.GPIO_PinConfig.GPIO_PinOPType = GPIO_OP_TYPE_PP;
        GPIOLed.GPIO_PinConfig.GPIO_PinPuPdCOntrol = GPIO_NO_PUPD;
        GPIO_Init(&GPIOLed);
        }

        // Turn LED1 ON, LED2 OFF
        controlLED(GPIO_PIN_NO_12, true);
        controlLED(GPIO_PIN_NO_13, false);
        delay();
        // Turn LED1 OFF, LED2 ON
        controlLED(GPIO_PIN_NO_12, false);
        controlLED(GPIO_PIN_NO_13, true);
        delay();

        while(1);
}
```

Q2. Blink LEDs in Sequence: Write a program that blinks the 4 LEDs in sequence (LED1 -> LED2 -> LED3 -> LED4) with a delay between each. After LED4, the sequence should repeat.

```
#include "stm32f407xx.h"

// Delay function
void delay() {
	for (uint32_t i = 0; i < 500000; i++);
}

// Control LED ON/OFF
void controlLED(uint8_t PinNumber, bool state) {
	if (state) {
	GPIO_WriteToOutputPin(GPIOD, PinNumber, GPIO_PIN_SET);
	} else {
	GPIO_WriteToOutputPin(GPIOD, PinNumber, GPIO_PIN_RESET);
	}
}

void blinkLEDsInSequence() {
	for (uint8_t i = 0; i < 4; i++) {
	controlLED(GPIO_PIN_NO_12 + i, true);  // Turn ON the current LED
	delay();
	controlLED(GPIO_PIN_NO_12 + i, false); // Turn OFF the current LED
	delay();
	}
}

int main(void) {
	GPIO_Handle_t GPIOLed;
	GPIOLed.pGPIOx = GPIOD;

	// Initialize all LEDs
	for (uint8_t i = 0; i < 4; i++) {
	GPIOLed.GPIO_PinConfig.GPIO_PinNumber = GPIO_PIN_NO_12 + i;
	GPIOLed.GPIO_PinConfig.GPIO_PinMode = GPIO_MODE_OUT;
	GPIOLed.GPIO_PinConfig.GPIO_PinSpeed = GPIO_SPEED_FAST;
	GPIOLed.GPIO_PinConfig.GPIO_PinOPType = GPIO_OP_TYPE_PP;
	GPIOLed.GPIO_PinConfig.GPIO_PinPuPdCOntrol = GPIO_NO_PUPD;
	GPIO_Init(&GPIOLed);
	}

	while(1) {
	blinkLEDsInSequence();
	}
}
```

Q3.Binary Counter with LEDs: Implement a binary counter using the 4 LEDs. Starting from 0000 (all OFF), increment the count every second, displaying the binary representation of the counter on the LEDs (ON = 1, OFF = 0).

```
#include "stm32f407xx.h"

// Delay function
void delay() {
    for (uint32_t i = 0; i < 500000; i++);
}

// Control LED ON/OFF
void controlLED(uint8_t PinNumber, bool state) {
    if (state) {
    GPIO_WriteToOutputPin(GPIOD, PinNumber, GPIO_PIN_SET);
    } else {
    GPIO_WriteToOutputPin(GPIOD, PinNumber, GPIO_PIN_RESET);
    }
}

void binaryCounter() {
    uint8_t counter = 0;
    while (1) {
    for (uint8_t i = 0; i < 4; i++) {
    if ((counter >> i) & 1) {
        controlLED(GPIO_PIN_NO_12 + i, true); // LED ON for 1
    } else {
        controlLED(GPIO_PIN_NO_12 + i, false); // LED OFF for 0
    }
    }
    delay();
    counter++; // Increment the counter
    }
}

int main(void) {
    GPIO_Handle_t GPIOLed;
    GPIOLed.pGPIOx = GPIOD;

    // Initialize all LEDs
    for (uint8_t i = 0; i < 4; i++) {
    GPIOLed.GPIO_PinConfig.GPIO_PinNumber = GPIO_PIN_NO_12 + i;
    GPIOLed.GPIO_PinConfig.GPIO_PinMode = GPIO_MODE_OUT;
    GPIOLed.GPIO_PinConfig.GPIO_PinSpeed = GPIO_SPEED_FAST;
    GPIOLed.GPIO_PinConfig.GPIO_PinOPType = GPIO_OP_TYPE_PP;
    GPIOLed.GPIO_PinConfig.GPIO_PinPuPdCOntrol = GPIO_NO_PUPD;
```

```
        GPIO_Init(&GPIOLed);
        }

        binaryCounter();
}
```

Q4. Alternate Blinking: Create a program that makes LED1 and LED3 blink alternately with LED2 and LED4, each group toggling every second.

```
#include "stm32f407xx.h"

// Delay function
void delay() {
        for (uint32_t i = 0; i < 500000; i++);
}

// Control LED ON/OFF
void controlLED(uint8_t PinNumber, bool state) {
        if (state) {
        GPIO_WriteToOutputPin(GPIOD, PinNumber, GPIO_PIN_SET);
        } else {
        GPIO_WriteToOutputPin(GPIOD, PinNumber, GPIO_PIN_RESET);
        }
}

void alternateBlinking() {
        while (1) {
        // Group 1 (LED1 and LED3)
        controlLED(GPIO_PIN_NO_12, true);  // LED1 ON
        controlLED(GPIO_PIN_NO_14, true);  // LED3 ON
        delay();
        controlLED(GPIO_PIN_NO_12, false); // LED1 OFF
        controlLED(GPIO_PIN_NO_14, false); // LED3 OFF

        // Group 2 (LED2 and LED4)
        controlLED(GPIO_PIN_NO_13, true);  // LED2 ON
        controlLED(GPIO_PIN_NO_15, true);  // LED4 ON
        delay();
        controlLED(GPIO_PIN_NO_13, false); // LED2 OFF
        controlLED(GPIO_PIN_NO_15, false); // LED4 OFF
        }
}

int main(void) {
        GPIO_Handle_t GPIOLed;
        GPIOLed.pGPIOx = GPIOD;
```

```
	// Initialize all LEDs
	for (uint8_t i = 0; i < 4; i++) {
	GPIOLed.GPIO_PinConfig.GPIO_PinNumber = GPIO_PIN_NO_12 + i;
	GPIOLed.GPIO_PinConfig.GPIO_PinMode = GPIO_MODE_OUT;
	GPIOLed.GPIO_PinConfig.GPIO_PinSpeed = GPIO_SPEED_FAST;
	GPIOLed.GPIO_PinConfig.GPIO_PinOPType = GPIO_OP_TYPE_PP;
	GPIOLed.GPIO_PinConfig.GPIO_PinPuPdCOntrol = GPIO_NO_PUPD;
	GPIO_Init(&GPIOLed);
	}

	alternateBlinking();
}
```

Q5.	Traffic Light Simulation: Simulate a traffic light system using the 4 LEDs. Assign them as Red, Yellow, Green, and a Pedestrian light. Use appropriate timing sequences to mimic real-world behavior.

```
#include "stm32f407xx.h"

void delay(uint32_t delay_time);

int main(void)
{
	GPIO_Handle_t GPIOLed;
	GPIOLed.pGPIOx = GPIOD;

	// Enable the clock for GPIOD Peripheral
	GPIO_PeriClockControl(GPIOD, ENABLE);

	// Initialize all 4 LEDs (PD12, PD13, PD14, PD15)
	for (uint8_t i = 0; i < 4; i++) {
	GPIOLed.GPIO_PinConfig.GPIO_PinNumber = GPIO_PIN_NO_12 + i;
	GPIOLed.GPIO_PinConfig.GPIO_PinMode = GPIO_MODE_OUT;
	GPIOLed.GPIO_PinConfig.GPIO_PinSpeed = GPIO_SPEED_FAST;
	GPIOLed.GPIO_PinConfig.GPIO_PinOPType = GPIO_OP_TYPE_PP;
	GPIOLed.GPIO_PinConfig.GPIO_PinPuPdCOntrol = GPIO_NO_PUPD;
	GPIO_Init(&GPIOLed);
	}

	while (1) {
	// Red light ON, Yellow and Green OFF, Pedestrian light OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_12, GPIO_PIN_SET); // Red ON
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_13, GPIO_PIN_RESET); // Yellow
OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_14, GPIO_PIN_RESET); // Green
OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_15, GPIO_PIN_RESET); //
Pedestrian OFF
```

```
	delay(5000000); // Red light duration

	// Yellow light ON, Red and Green OFF, Pedestrian light OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_12, GPIO_PIN_RESET); // Red OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_13, GPIO_PIN_SET); // Yellow ON
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_14, GPIO_PIN_RESET); // Green OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_15, GPIO_PIN_RESET); // Pedestrian OFF
	delay(2000000); // Yellow light duration

	// Green light ON, Red and Yellow OFF, Pedestrian light OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_12, GPIO_PIN_RESET); // Red OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_13, GPIO_PIN_RESET); // Yellow OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_14, GPIO_PIN_SET); // Green ON
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_15, GPIO_PIN_RESET); // Pedestrian OFF
	delay(5000000); // Green light duration

	// Pedestrian light ON, others OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_12, GPIO_PIN_RESET); // Red OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_13, GPIO_PIN_RESET); // Yellow OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_14, GPIO_PIN_RESET); // Green OFF
	GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_15, GPIO_PIN_SET); // Pedestrian ON
	delay(3000000); // Pedestrian light duration
	}
}

void delay(uint32_t delay_time) {
	for (uint32_t i = 0; i < delay_time; i++);
}
```

Q6.	LED Pattern Generator: Allow the user to define custom ON/OFF patterns for the 4 LEDs via an array. For example, the input [1, 0, 1, 0] should turn LED1 and LED3 ON, and LED2 and LED4 OFF.

```
#include "stm32f407xx.h"

void delay(void);
void controlLED(uint8_t led_num, uint8_t state);
```

```
int main(void)
{
    GPIO_Handle_t GPIOLed;
    GPIOLed.pGPIOx = GPIOD;

    // Enable the clock for GPIOD Peripheral
    GPIO_PeriClockControl(GPIOD, ENABLE);

    // Initialize all 4 LEDs (PD12, PD13, PD14, PD15)
    for (uint8_t i = 0; i < 4; i++) {
    GPIOLed.GPIO_PinConfig.GPIO_PinNumber = GPIO_PIN_NO_12 + i;
    GPIOLed.GPIO_PinConfig.GPIO_PinMode = GPIO_MODE_OUT;
    GPIOLed.GPIO_PinConfig.GPIO_PinSpeed = GPIO_SPEED_FAST;
    GPIOLed.GPIO_PinConfig.GPIO_PinOPType = GPIO_OP_TYPE_PP;
    GPIOLed.GPIO_PinConfig.GPIO_PinPuPdCOntrol = GPIO_NO_PUPD;
    GPIO_Init(&GPIOLed);
    }

    // Define the pattern for LEDs
    uint8_t led_pattern[4] = {1, 0, 1, 0};  // LED1 and LED3 ON, LED2 and LED4 OFF

    while (1) {
    // Apply the pattern
    for (uint8_t i = 0; i < 4; i++) {
    controlLED(i, led_pattern[i]);
    }

    delay();
    }
}

void controlLED(uint8_t led_num, uint8_t state) {
    if (state == 1) {
    GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_12 + led_num, GPIO_PIN_SET); //
LED ON
    } else {
    GPIO_WriteToOutputPin(GPIOD, GPIO_PIN_NO_12 + led_num,
GPIO_PIN_RESET); // LED OFF
    }
}

void delay(void) {
    for (uint32_t i = 0; i < 500000; i++);
}
```