I tried to have the onLED function do the turning on of the appropriate LED and used an “else” to turn in off instead of having a separate offLED function. The timing part works right, but when the LED is supposed to be on, it blinks on and off. It is because when the times are right, the “if” makes the LEDstate HIGH and then the next time around it goes to the “else” statement and gets made LOW, then the next time around it gets made HIGH again in the “if” statement.

```
/*3LEDs_same_cycle_time_Functions_shortened_version
 *
 * The program has the start time each cycle that the LEDs going on and off is timed from.
 Uses functions to simplify the code.

 The 2nd red LED changes state at the beginning of the cycle.
The red, green, and yellow LEDs are started independently from the a delay from the start
of the cycle.  They are each on for a certain amount of time, then turn off.

*/

unsigned long cycleStartMillis = millis(); // will last time the cycle started
long cycleLength = 10000; // how long each cycle is

unsigned long currentMillis = millis();

int ledPin4 = 12; // the number of the 2nd red LED pin
int ledState4 = LOW; // ledState used to set the LED
int ledState1 = LOW; // ledState used to set the LED
int ledState2 = LOW; // ledState used to set the LED
int ledState3 = LOW; // ledState used to set the LED

void setup()
{
// set the digital pin as output:-
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
pinMode(12, OUTPUT);
Serial.begin(9600);
}

int onLED(int ledPin, int ledState, long ledOnDelay, long ledOnTime)
{
if ((ledState == LOW) && (currentMillis - cycleStartMillis >= ledOnDelay)  &&
(currentMillis - cycleStartMillis <= ledOnDelay + ledOnTime))
{
  ledState = HIGH;
digitalWrite(ledPin, ledState); // turn on the LED
Serial.print("ON Command for LED:");
Serial.println(ledPin);
```

```
return ledState;
}
else     //put the else here to do the LED off;
{
  ledState = LOW;
digitalWrite(ledPin, ledState); // turn off the LED
Serial.print("OFF Command for LED:");
Serial.println(ledPin);
//Serial.println("LED OFF Command");
return ledState;  //returns LOW as the state and this just makes it go into the if and gets made HIGH
}
//return ledState;   had this here, but didn't work so I moved it to the end of the "if" and the "else" parts above, but it still didn't work

}

/*
int offLED(int ledPin, int ledState, long ledOnDelay, long ledOnTime)
{
if((ledState == HIGH) && (currentMillis - cycleStartMillis > ledOnDelay + ledOnTime))
{
  ledState = LOW;
digitalWrite(ledPin, ledState); // turn off the LED
Serial.print("OFF Command for LED:");
Serial.println(ledPin);
//Serial.println("LED OFF Command");
}
return ledState;
}

*/

void loop()
{
// get the current time and put in currentMillis
currentMillis = millis();
if (currentMillis - cycleStartMillis >= cycleLength)
{
cycleStartMillis = currentMillis; // Reset the cycle start time to the current time
if (ledState4 == LOW)
{
  ledState4 = HIGH;
  digitalWrite(ledPin4, ledState4); // change the state of the 2nd red LED; cycle length indicator
  Serial.println("Start of Cycle - LED4 high");
}
else
{
```

```
  ledState4 = LOW;
  digitalWrite(ledPin4, ledState4); // change the state of the 2nd red LED; cycle length indicator
  Serial.println("Start of Cycle - LED4 low");
}

}



ledState1 = onLED(9, ledState1, 2000, 2000);
ledState2 = onLED(10, ledState2, 3000, 4000);
ledState3 = onLED(11, ledState3, 5000, 3000);
//ledState1 = offLED(9, ledState1, 2000, 2000);  tried to do both the on and the off in the above functions
//ledState2 = offLED(10, ledState2, 3000, 4000);
//ledState3 = offLED(11, ledState3, 5000, 3000);


}  // void loop bracket
```