

```

#include <stdint.h>
#include "eecs388_lib.h"

/*
    task 1.4: build and deploy/upload the code as is
    task 1.5: try out debugging. See lab sheet.
    task 2.1: review the files eecs388_lib.h and eecs388_lib.c.
               on vscode, it will usually be on the file browser pane on the left

    task 2.2: implement RGB blinky.
    - set red, green, and blue LEDs for output mode, using the gpio_mode() function
    - in the main loop, turn on/off a different LED at each iteration.
      for example,
      at 1st iteration
        turn on RED led
        delay 500 ms
        turn off RED led
        delay 300 ms.
      at 2nd iteration
        do the same but for GREEN led
      at 3rd iteration
        do the same but for BLUE led
      at 4th iteration
        back to RED led.
      ...
    - for extra marks, modify your code so that it also flashes the WHITE color
      so sequence would be something like this: RED -> BLUE -> GREEN -> WHITE
*/

int main() {
    // GREEN_LED is a macro.
    // It is essentially a placeholder that is replaced by the compiler during
    compilation into the correct pin number
    // To see the actual pin numbers, look for GREEN_LED (or the other colors) in the
    eecs388_lib.h file

    // redeclared variables to R representing the RED_LED macro
    // G representing the GREEN_LED macro
    // B representing the BLUE_LED macro
    int R = RED_LED;
    int G = GREEN_LED;
    int B = BLUE_LED;

    // this function sets the pin referred to by the variable gpio to act as an OUTPUT
    pin, i.e. send signals out of it
    // called the functions again to set the other pins to output mode
    gpio_mode(R, OUTPUT);
    gpio_mode(G, OUTPUT);
    gpio_mode(B, OUTPUT);

    // your code will need to be inside this while loop
    // but feel free to do it any other way if you like. As long the expected behaviour
    is present, you will be credited
    while (1) {
        // the gpio_write() function sets a pin (referred to by gpio in this case) to the
        logic high or ON state.
        gpio_write(R, ON);
        // you could also think of it as switching on a pin so that it's now sending out a
        signal
        // this function asks the processor to just do nothing and wait for 1000
        milliseconds (1 sec)
        delay(500);
        // same as before, but this time we are setting the pin to logic low or OFF state.
        Same as turning the pin off
        gpio_write(R, OFF);
        delay(300);

        // the write function will take the green macro as the first parameter and the ON
        macro as the second parameter
        gpio_write(G, ON);
        // the delay function takes a millisecond of type integer as the single parameter
        value
        delay(500);
        // the write function will take the green macro as the first parameter and the OFF
        macro as the second parameter
        gpio_write(G, OFF);
        // delay function takes a millisecond of type integer as the single parameter
        value
        delay(300);

        // same thing with the blue LED and RED LED
        gpio_write(B, ON);
        delay(500);
        gpio_write(B, OFF);
        delay(300);

        gpio_write(R, ON);
    }

}

/*
    - the main function is the entry point of the program
    - we first initialize the variables R, G, and B to the RED_LED, GREEN_LED, and
    BLUE_LED macros respectively
    - the gpio_mode() function is used to set the pin to either input or output mode
    - in our case we are setting the pin to output mode which will allow the pin to send
    signals to the HiFive1 Microcontroller
    - the parameter values R, B, and G are the pin numbers
    - gpio stands for general purpose input output
    - then we call a while loop which will iterate indefinitely
    - and we call the following functions 3 times each for each LED
    - gpio_write function is used to set the pin to either high or low voltage
    - the parameter values R, B, and G are the pin numbers
    - the parameter values ON and OFF are the state variables which represent high
    voltage as 1 or low voltage as 0

    gpio_mode() function
    (if R then the pin is the RED_LED macro at position 22)
    (if B then the pin is the RED_LED macro at position 21)
    (if G then the pin is the RED_LED macro at position 19)
    - the first parameter value is the pin number
    - the second parameter value is the output/input value which is defined in the header
    file as a macro as either 0 or 1
    - 0 represents input mode which will allow the pin to receive signals from this
    device to the board
    - 1 represents output mode which will allow the pin to send signals to the board
    - this function in general will set the pin to either receive or send signals
*/

```