2a) The toggle_led() function is for playing around with the LEDs. It isn't part of the lab requirements.

ticks_to_count in delay() function was calibrated using the stop watch on my phone

In set_led(), each LED's bit is set individually leading to three statement groups. I know I can update all three LEDs in just one line but separating them into three is more explicit even if less efficient.

The gpiodata_register_test() function is for answering c.vi. It is commented out in main() but can be uncommented to confirm that my understanding of the data register is correct. The blinking order is rgb instead of rbg to differentiate it from question 1. This method of using the data register wasn't used in the answer to question 1 because I discovered it after finishing that question and I didn't want to redo the previous question.

2ci)
```
#define GPIO_PORTF_DATA_R (*((volatile unsigned long *)0x400583FC))
#define GPIO_PORTF_DIR_R (*((volatile unsigned long *)0x40058400))
#define GPIO_PORTF_AFSEL_R (*((volatile unsigned long *)0x40058420))
#define GPIO_PORTF_DEN_R (*((volatile unsigned long *)0x4005851C))
#define GPIO_PORTF_PUR_R (*((volatile unsigned long *)0x40058510))
```

2cii) To change SW1 to be active high instead of active low, the switch will need to be connected to +VBUS instead of ground. Instead of using the pull up resistor in the MCU using GPIOPUR, a pull down resistor will need to be used which is configured using the GPIOPDR register.

2cv) `SYSCTL_RCGC2_R |= 0x00000023;`

2cvi) Accessing the bits of the GPIODATA register is done differently than normal. This is explained on page 654 of the manual. For port F, the base for GPIODATA is 0x4002 5000 and the offset is 0x000. However, this register is configured to use bits [9:2] of 0x40025000 as a mask. The masking is handled by the cpu so bit masking in software is not necessary. Since GPIODATA is expected to be read and written to many times, this optimization decreases the number of instructions needed to read and write to a register. With the address 0x400253FC, bits [9:2] of the address are all 1 so all non-reserved bits of GPIODATA can be interacted with. With 0x33c = 0b0011 0011 1100, so bits 4 and 5 of GPIODATA won't be written to and when read will return 0. See gpiodata_register_test() in main.c to see an example of setting and clearing PF1, PF2, and PF3 without software level bit masking.