1. How many tasks are running in blinky2?

- One task (task1)

2. How many threads of control are operating in blinky2?

- One thread (main)

3. What would happen to the blink rate of blinky2 if the value of the configCPU-CLOCK_HZ were configured as 36000000?

- It would become slower 2 times.

4. Where does task1's stack come from?

- From the HEAP of Freertos which comes from the SRAM of the board

5. Exactly when does task1() begin?

- When 'vTaskStartScheduler' has been called

6. Why is a message queue needed?

- To ensure communication between tasks without loss of information and race conditions

Change to the project in stm32/rtos/blink, build it, and run it. Then, answer the following:

7. Even though it uses an execution delay loop, why does it seem to work with a nearly 50 percent duty cycle?

- Because the premption of Freertos

8. How difficult is it to estimate how long the LED on PC13 is on for? Why?

- The Scheduler will cut the execution of task1() irregardless if the loop is finished or not

9. Using a scope, measure the on and off times of PC13 (or count how many blinks per second and compute the inverse). How many milliseconds is the LED on for?

- 30 bps, 0.5ms

10. If another task were added to this project that consumed most of the CPU, how would the blink rate be affected?

- The blink rate will become variable (less blinking) but this depends on how much the other task reserves the cpu

11. Add to the file main.c a task2 that does nothing but execut **asm**("nop") in a loop. Create that task in main() prior to starting the scheduler. How did that impact the blink rate? Why?

```
static void
task2(void *args) {
  int i;
  (void)args;
  for (;;) {
    for (i = 0; i < 10000000; i++)
      __asm("nop");
  }
}

int
main(void) {

    rcc_clock_setup_in_hse_8mhz_out_72mhz();    // Use this for "blue pill"
    rcc_periph_clock_enable(RCC_GPIOC);
    gpio_set_mode(GPIOC,GPIO_MODE_OUTPUT_2_MHZ,GPIO_CNF_OUTPUT_PUSHPULL,GPIO13);

    xTaskCreate(task1,"LED",100,NULL,configMAX_PRIORITIES-1,NULL);
    xTaskCreate(task2,"NOPS",100,NULL,configMAX_PRIORITIES-1,NULL);
    vTaskStartScheduler();
    for (;;);

    return 0;
}
```

- The blink rate got reduced because of the premption.