Nathan Donaldson/U0632803

1. What are the GPIO control registers that the lab mentions?
    - MODER - These bits are written by software to configure the I/O mode.
    - OTYPER - These bits are written by software to configure the I/O output type.
    - OSPEEDR - These bits are written by software to configure the I/O output speed.
    - PUPDR - These bits are written by software to configure the I/O pull-up or pull-down.
    - ODR - These bits can be read and written by software.
    - IDR - These bits are read-only. They contain the input value of the corresponding I/O port.
    - BSSR - BSRR is also used for setting a GPIO pin HIGH or LOW, but it's done is a different way. Setting a bit to zero in the register will not actually update the GPIO pin and send it LOW. To set a GPIO low you actually have to set its corresponding bit HIGH in the upper 16 bits of the register.

2. What values would you want to write to the bits controlling a pin in the GPIOx_MODER register in order to set it to analog mode?
    - 11

3. Examine the bit descriptions in GPIOx_BSSR register, what bit would you want to set to clear the fourth bit in the ODR?
    - 4 or 20

4. Perform the following bitwise operations:
    - 0xAD | 0xC7 = 0xEF
    - 0xAD & 0xC7 = 85
    - 0xAD & ~(0xC7) = 28
    - 0xAD ^ 0xC7 = 6A

5. How would you clear the 5th and 6th bits in a register while leaving the other's alone?
    - Register &= ~(0x30)

6. What is the maximum speed the STM32F072R8 GPIO pins can handle in the lowest speed setting?
    - 2MHZ

7. What RCC register would you manipulate to enable the following peripherals: (use the comments next to the bit defines for better peripheral descriptions)
    - TIM1 (TIMER1) – You would use register RCC_APB2ENR).

- DMA1 - RCC_AHBENR_DMA1EN
- I2C1 - RCC_APB1ENR