1. **What are the GPIO control registers that the lab mentions? Briefly describe each of their functions.**
   The function of the GPIO control registers are:
   - GPIO Port Mode Register (GPIOx_MODER): this is the most important because it determines the function of a pin within a GPIO port.
   - GPIO port output type register (GPIOx_OTYPER): it sets the output type for each GPIO pin, specifying if it is a push-pull or an open-drain configuration.
   - GPIO port output speed register (GPIOx_OSPEEDR): this register configures the operating speed for the GPIO pins, which will impact power consumption and signal rise times.
   - GPIO port pull-up/pull-down register (GPIOx_PUPDR): enables internal pull-up or pull-down resistors to set a defined logic level when the pin is floating.
   - GPIO port input data register (GPIOx_IDR): this gives the input's current value for the GPIO pins.
   - GPIO port output data register (GPIOx_ODR): this is responsible for setting the output value for the GPIO pins.
   - GPIO port bit set/reset register (GPIOx_BSRR): this register enables setting or clearing specific output bits within an atomic operation while avoiding read-modify-write sequences.
   - GPIO port configuration lock register (GPIOx_LCKR) prevents accidental changes to the GPIO pins' configuration.
   - GPIO alternate function low/high registers (GPIOx_AFRL/GPIOx_AFRH): when configured in alternate function mode, this selects the alternate function for the GPIO pins.
   - GPIO port bit reset register (GPIOx_BRR) is similar to the bit set/reset register. However, it allows the clearing of bits using the bit set/reset register. All of the clearing bits lie in the upper half 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?**
   To set it to analog mode, you would write the value ‘11’ to the bits that control that pin. Each pin is controlled by two bits, and the value of ‘11’ specifies analog mode.

3. **Examine the bit descriptions in GPIOx_BSRR register: which bit would you want to set to clear the fourth bit in the ODR?**
   To clear the fourth bit inside the ODR, you should set the 20th bit since the bit number starts at 0. Doing this will clear the corresponding bit in the ODR but not affect other bits.

4. **Perform the following bitwise operations:**
   - **0xAD | 0xC7 =** 1010 1101 | 1100 0111 = 1110 1111 = 0xEF

- **0xAD & 0xC7 =** 1010 1101 & 1100 0111 = 1000 0101 = 0x85
- **0xAD & ~(0xC7) =** 1010 1101 & ~(1100 0111) = 0010 1000 = 0x28
- **0xAD ^0xC7 =** 1010 1101 ^1100 0111 = 0110 1010 = 0x6A

5. **How would you clear the 5th and 6th bits in a register while leaving the other's alone?**
   To clear only the 5th and 6th bits of a register without affecting any other bits, you can perform a read-modify-write operation. This involves reading the current value of the register, then applying an AND operation to the value and a mask where the 5th and 6th bits are set to '0' and all the other bits are set to '1'. Finally, write the result back to the register to complete the operation.

6. **What is the maximum speed the STM32F072R8 GPIO pins can handle in the lowest speed setting?**
   - **Use the chip datasheet: lab section 1.4.1 gives a hint to the location. You'll want to search the I/O AC characteristics table. You will also need to view the OSPEEDR settings to find the bit pattern indicating the slowest speed.**

   The lowest speed setting for the GPIO pins of the STM32F072R8, the maximum frequency ($f_{max(IO)out}$) the GPIO pins can handle is 2 MHz for the condition of C_L = 50 pF, $V_{DDIOx} \geq 2$ V. This information is found in the I/O AC characteristics table from the *(STM32F072RBT6 Datasheet) DM00090510.pdf* under OSPEEDR settings that indicate a bit pattern for the slowest speed; in this case, it will be OSPEEDRy[1:0] value of 'x0' in pg. 82.

7. **What RCC register would you manipulate to enable the following peripherals: (use the comments next to the bit defines for better peripheral descriptions)**
   From the *(Peripheral Manual) DM00031936.pdf* in pg. 121-125:
   - **TIM1 (TIMER1):** to enable TIM1, you need to set the TIM1EN bit to '1' in the APB peripheral clock enable register 2 (RCC_APB2ENR).
   - **DMA1:** to enable DMA, you need to set the DMAEN bit to '1' in the AHB peripheral clock enable register (RCC_AHBENR).
   - **I2C1:** to enable I2C1, you need to set the I2C1EN bit to '1' in the APB peripheral clock enable register 1 (RCC_APB1ENR).