Name and surname:		Student number:			Signature:	
Course: BME 3321 Introduction to MCU Programming		**Date / Time:**			**Time:** 90 minutes	
Exam Type:		**Midterm1**	**Midterm2**	**Make-up** for Midterms	**Final**	**Make-up**
Title Name-Surname: Assist. Prof. Dr. İsmail CANTÜRK **(Instructor)**						

1. GPIO registers in STM32 MCUs are managed with below C struct, whose references are used to point to real peripheral address.

```
typedef struct {
volatile uint32_t MODER;
…
volatile uint32_t IDR;
volatile uint32_t ODR;
…
} GPIO_TypeDef;
```

The below code is written for STM32F4 GPIO registers. Explain the C code line by line. Also express what this code does. Assume that the code is executed after resetting.

```
#include "main.h"
…
GPIO_TypeDef *GPIOC = 0x 0x4002 0800;
GPIO_TypeDef *GPIOD = 0x 0x4002 0C00;
…
int main () {
GPIOC->MODER |= 0x0000 0000;
GPIOD-> MODER |= 0x5555 5555;
…
if (GPIOC->IDR >0x0000&& GPIOC->IDR <0x0100)
       {
        GPIOD->ODR=0x00FF;
       }
if (GPIOC->IDR >0x00FF&& GPIOC->IDR <0xFFFF)
       {
        GPIOD->ODR=0xFF00;
       }
…
```

2. GPIO registers in STM32 MCUs are managed with above C struct. Both A and D ports of STM32F4 have 16 pins. A push button is connected to the PA0 pin in pull-down connection. Green, orange, red, and blue leds are connected to PD12, PD13, PD14, and PD15 pins respectively as shown in below circuit. Assume that all required register settings are done and your code will be executed after resetting. You should write only a few lines of codes for below question. Just like "if" cases in question 1.

(i) Write the piece of code below that reads the GPIOA->IDR with half word and that code must be able to detect arrival of logic 1 to PA0 pin. Explain your code shortly. Use bitwise operations.

(ii) Write the piece of code below that writes GPIOD->ODR to turn on the leds at the same time upon pressing of the push button which is connected to PA0 pin. (Hint: use C control structures and bitwise operations)

(iii) Write the code below that writes GPIOD->ODR to turn on the leds consecutively. The order of turning on: PD12, PD13, PD14, PD15.
(Hint: use C loops and bitwise operations)
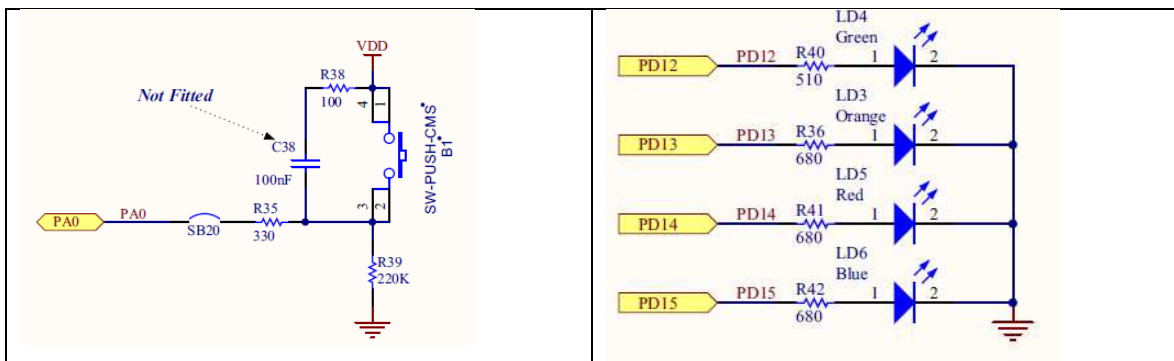
3. Nested Vectored Interrupt Controller (NVIC) of STM32 MCU receives three external interrupts at different times as shown below. Priorities of Interrupt X, Interrupt Y, and Interrupt Z are given as 1, 2, and 3 respectively. Sketch the processor modes and processor operations. Show stacking, unstacking, vector fetches, and interrupt service routines (ISR) for all interrupts.

Hint:
Clear the interrupt requests (IRQ) during ISRs.
ISR of X is completed before IRQ of Y.
IRQ of Y is received while executing ISR of Z.