Q1.

a) What is TMR0IF? When is it set to 1?

b) You set T0CON=0b10001001, TIMER0H=0 and TIMER0L=0.
After a while you read that TIMER0H=0, TIMER0L=0x14.
How many clock cycles has been passed since the timer was set?

c) If a TRISB=0x34 is executed, which pins of RB[0..7] are input pin? Put a checkmark.

   RB7 ( );  RB6 ( );  RB5 ( );  RB4 ( );  RB3 ( );  RB2 ( );  RB1 ( );  RB0 ( );

d) If you set T0CON=0b11000000, TIMER0H=0, TIMER0L=0x0F6; how many clock
cycles must pass to have time-over of TIMER0?

e) What is the main difference of the HS and XT Crystals?

Q2.  A PIC18F452 running on a 4MHz crystal (Tc=1μs) shall read two 8-bit numbers PB and
PC from PORTB and PORTC at every 500ms, then calculate

   BC = (low-nibble of PB)*(high-nibble of PC),
and output BC from PORTD.  For example, if PORTB=0x45; and PORTC=0x0BA, then
BC=0x05*0x0B = 0x37 shall appear at PORTD.
a) Specify the process in Program Description Language (PDL) or in Pseudo Code by the
following items:
   i) Initialization module that initializes ports and timer0.
   ii) Mainloop module that inputs PB and PC, then calculates and writes BC to output, waits
       for time-over, and sets timer0 for next 0.1s in an endless loop.

b) The 500ms timing specified in the problem tolerates 10% timing errors. Your boss wants a
cheaper design with RC clock, but RC clock runs at maximum 1MHz oscillator rate
(Tc=250kHz). What kind of modifications are necessary in your software to run it?

c) Write the simplest possible CC8E code that is suitable to test the timer0, and the output port
PORTD (i.e., increment PORTD at every 500ms).

d) Write the complete CC8E code for Q2 considering the following items:
   • initialize ports for input and output, set the timer0 for 500ms period,
   • declare all variables properly,
   • in the mainloop, read the ports PB=PORTB and PC=PORTC, and calculate BC. Then write
     BC to PORTD.
   • Before you loop back to the beginning of the mainloop wait until the timer0 period is over,
     and set timer0 for the next 500 ms period.
Q3.
A light-timer is a device to turn on the lights by pushing a button, and keep them on for a pre specified time after the button is released. The given circuit runs on a PIC18F452 device with a 40 kHz RC oscillator. The relay closes the mains circuit to light the lamps while the relay coil is 5 V.

Indicate with a check-sign [✓] or fill in the answer:

a) 2p. Which pin is button input-pin?
   [ ] RB0, [ ] RB1, [ ] RB2, [ ] RB3.

b) 2p. What will be the button pin input if the button is pushed? [ ] low, [ ] high.

c) 2p. Which pin is relay coil output?
   [ ] RB0, [ ] RB1, [ ] RB2, [ ] RB3.

d) 4p. According to C-code which of the following pins are configured to be an input pin?
   [ ] RB0, [ ] RB1, [ ] RB2, [ ] RB3;
   and which pins are configured as output?
   [ ] RB0, [ ] RB1, [ ] RB2, [ ] RB3;

e) 2p. Counter size: [ ] 8-bit, [ ] 16-bit.

f) 2p. Source Signal: [ ] Clock, [ ] RA4/T0CKI.

g) 2p. Prescaler: [ ] enabled, [ ] disabled.

h) 4p. Prescaler Ratio: [ ] 2, [ ] 4, [ ] 8, [ ] 16, [ ] 32, [ ] 64, [ ] 128, [ ] 256.

e) 5p. How many controller clock cycles shall Timer0 count to overflow? [………].

f) 5p. Processor clock rate with a 40 kHz RC clock is 10kHz. What is the processor clock time period of the processor in milliseconds? [………………..] millisecond.

g) 5p. How long does the processor keep relay coil voltage at 5V after releasing the button?
   […………………………………….] second
Q4.
For the following PIC18 assembly code write how many clock cycles passed from the rising edge of RB0 to the rising edges of RB1, RB2, and RB3.

```assembly
include <p18f452.inc>
mainloop:
    clr TRISB   ; TRISB <-- 00000000
    clr PORTB   ; PORTB <-- 00000000
    bsf PORTB.0 ; RB0 rising edge
    movlw 5    
    movwf PORTB ; PORTB <-- 00000101
    movlw 5    ; wreg <-- 5
    innerloop:
        decf WREG ; WREG <-- WREG - 1
        bnz innerloop ; branch if ZF=0 to innerloop
        bsf PORTB,1 ; RB1 rising edge
        bcf PORTB,0 ; RB0 falling edge
        movlw b'11001000'
        movwf T0CON ; Start 8-bit timer0 no prescaler
        movlw -30   ; decimal -30
        movwf TMR0L ; TMR0L <-- -30
        bcf INTCON,TMR0IF ; TMR0IF <-- 0
    tloop:
        btfss INTCON,TMR0IF ; test TMR0IF bit
        bra tloop ; if TMR0IF==0 branch to tloop
        bsf PORTB,3 ; RB3 rising edge
        bra mainloop
end
```

(solution for RB1:
list of instructions from RB0 rising edge to RB1 rising edge is
{movlw, movwf, movlw, 5x{decf, bnz}, bsf.}
1cc + 1cc + 1cc + 5x( 1cc +2cc)+ 1cc  = 4+5x3=19 clock cycles.)