

Lab 6

Task 1

```
#include <xc.h>

void Delay(){
    int i;
    for (i=0;i<732;i++){
        TMR0=0x00;

        while(INTCONbits.TMR0IF==0);

        INTCONbits.TMR0IF=0;
    }

    TMR0 = 0x94;
    while(INTCONbits.TMR0IF==0);
}

int main(){
    TRISB=0xFF;
    TRISC=0x00;
    OPTION_REG=0x03;

    INTCONbits.TMR0IF=0;

    while(PORTBbits.RB0==1){
        PORTCbits.RC0=1;
        Delay();
        PORTCbits.RC0=0;
        Delay();
    }

    return 0;
}
```

Task 2

```

#include <xc.h>

void Delay_2s(){
    int i;
    for (i=0;i<488;i++){
        TMR0=0x00;           // Reset Timer
        while(INTCONbits.TMR0IF==0); // Wait for overflow
        INTCNbits.TMR0IF=0;  // Clear overflow flag
    }
    TMR0 = 0xBD;             // Preload Timer0 for partial cycle
    while(INTCONbits.TMR0IF==0); // Wait for last overflow
}

int main() {

    TRISC = 0x00;            // PORTC as output (LED on RC0)
    TRISD = 0x08;            // RD3 (bit 3) as input for push button
    OPTION_REG = 0x03;       // Prescaler 1:16 assigned to Timer0
    INTCNbits.TMR0IF = 0;    // Clear Timer0 flag

    while (1) {
        if (PORTDbits.RD3 == 1) { // Button pressed?
            PORTCbits.RC0 = 1;     // LED ON
            Delay_2s();             // Wait 2 sec
            PORTCbits.RC0 = 0;     // LED OFF
            Delay_2s();             // Wait 2 sec
        }
        else {
            PORTCbits.RC0 = 0;     // LED stays OFF
        }
    }
    return 0;
}

```

Task 4

```
#include <xc.h>

void delay(){
    int i;
    for (i=0;i<488;i++){
        TMR0=0x00;
        while(INTCONbits.TMR0IF==0);
        INTCONbits.TMR0IF=0;
    }
    TMR0 = 0xBD;
    while(INTCONbits.TMR0IF==0);
}

int main() {
    int count;
    TRISA = 0x01;           // RA0 = input (button)
    TRISB = 0x00;           // PORTB = output (LEDs)
    PORTB = 0x00;           // Clear LEDs initially

    while (true) {
        if (PORTAbits.RA0 == 1) {           // Button pressed
            Delay();

            if (PORTAbits.RA0 == 1) {       // Check again after debounce
                count++;                     // increment counter

                if (count >= 10) {           // if 10 reached → reset to 0
                    count = 0;
                }

                PORTB = count;               // show counter value on LEDs

                // Wait until button released (to avoid fast repeat counts)
                while (PORTAbits.RA0 == 1);
            }
        }
    }
    return 0;
}
```