Goals:
The goal of the lab is to modify the given code to produce a waveform duty cycle over a suitable period corresponding to an internal 1 MHz clock.

Equipment Usage
For this lab the following equipment will be used:

- LED
- Atmega328P
- AVR Studio
- AVRISP mkII

Background:
Clear Timer On Compare
In previous labs, timer overflow was automatic occurring once the register had reached its peak value. For this lab, we will gain more control by choosing to clear our timer when it meets a specific value as opposed to it running a certain number of clock cycles. By initializing the proper WGM bits, the timer can be set to clear when it reaches the specified value placed in the OCRx register.

Output Compare Register
The **Output Compare Register** (OCRx) is related to the **Output Compare Pin** (OCx) and timer (TCNTx). By placing a specific value in the register, you can dictate that point where the timer set and resets it timer. The behavior of the OCRx is similar to normal timer behavior. The important use of the OCRx is that it is tied to a specific pin on the microcontroller.

Compare Match Output
**Compare Match Output** bits (COMx) are the bits that control the operation of the pin tied to the OCRx register. These bits are located in the TCCRx register. Manipulating these bits allows the user to control the behavior of the pin connected to the OCRx register. It is important to remember that the manipulation of these bits is also dependent of the mode of operation set by the WGM bits (i.e. Normal, CTC, Fast PWM, etc). Below is a table for the bit manipulation for Timer0.
Calculations:
On the next page are two figures that provide examples on how to calculate the proper OCR value for a specific frequency. In Figure 1, the examples show that given a specific frequency and duty cycle, you can calculate both the required prescaler and OCR value. In the second figure, shows how to calculate the frequency given a desired clock and OCR value.

Assuming XTAL = 8 MHz, make a pulse with duty cycle = 50% and frequency = 500KHz

\[
F_{\text{oc}} = \frac{f_{\text{ck}}}{2N(\text{OCR0}+1)} = \frac{8\text{MHz}}{2N(\text{OCR0}+1)} = \frac{500\text{KHz}}{1\text{MHz}}
\]

\[N(\text{OCR0}+1) = 8 \quad \left\{ \begin{array}{l}
N = 1 \text{ and } \text{OCR0} = 7 \\
N = 8 \text{ and } \text{OCR0} = 0
\end{array} \right. \]

LDI R20,7
OUT OCR0,R20
LDI R20,0x19
OUT TCCR0,R20

LDI R20,0
OUT OCR0,R20
LDI R20,0x1A
OUT TCCR0,R20

FIGURE 1

There are 256 clocks between two consecutive matches. Therefore

\[
T_{\text{timer clock}} = \frac{1}{8} \text{ MHz} = 0.125 \mu s
\]

\[
T_{\text{wave}} = 2 \times 256 \times 0.125 \mu s = 64 \mu s
\]

\[
F_{\text{wave}} = \frac{1}{64} \mu s = 15.625 \text{ Hz} = 15.625 \text{ kHz}
\]

FIGURE 2
Quick Calculation:
S*XTAL/N = # of clock cycles in period
XTAL = internal clock frequency
S = length of period (seconds)
N = prescaler value (1, 8, 64, 256, 1024)

Prelab:
Design 1: Modify the given Atmega32 code to produce a waveform on the Atmega328P with a 50% duty cycle. Select a suitable period corresponding to an internal 1MHz clock.

```
SBI DDRB, 3
BEGIN: LDI R20,69
    OUT OCR0,R20 ;OCR0 = 69
    LDI R20,0x19
    OUT TCCR0,R20 ;CTC, no prescaler, set on match
L1:   IN R20,TIFR
    SBR R20,OCF0 ; skip next instruction if OCF0 = 1
    RJMP L1
    LDI R16,1<<OCF0
    OUT TIFR,R16 ;clear OCF0
    LDI R20,99
    OUT OCR0,R20 ;OCR0 = 99
    LDI R20,0x29
    OUT TCCR0,R20 ;CTC no prescaler, clear on match
L2:   IN R20,TIFR
    SBR R20,OCF0 ; skip next instruction if OCF0 = 1
    RJMP L2
    LDI R16,1<<OCF0 ;clear OCF0
    OUT TIFR,R16
    RJMP BEGIN
```

Show how you calculated the values for TCNT, OCR, and TCCR

Lab Experiments:
Experiment 1: Program your Atmega328P chip with the modified code. Build the circuit and verify the program works. Demonstrate the working code to the TA.

Experiment 2: Modify the code to have the following duty cycles:

a) LED is on 33% over a period of 3 seconds connected to PB1 (Timer1)
b) LED is on 75% over a period of 4 seconds connected to PB2 (Timer1)
c) Toggle back and forth using PB3 and PB4 with 80% duty cycle with 5 second period
Questions:

1. What is differences must you be aware of when using OCR1A versus OCR1B?
2. Rewrite the code in part a using Timer0. Do you notice any complications using Timer0? If so what are they?
3. If you wanted to use OCR2A what pin would you connect to the LED to see it light up? Explain why you cannot connect it to any pin of your choice.
4. What is the period and duty cycle of the following code? (assume 1 MHz clock)

   ```
   SBI DDRB, 1
   
   BEGIN:  LDI R20, 0x7A
   STS OCR1AH, R20
   LDI R20,0xAF
   STS OCR1AL,R20
   LDI R20,0x0B
   STS TCCR1B,R20 ;CTC mode, prescaler = 64, WGM = 0100 CS = 101
   LDI R20,0x40
   STS TCCR1A,R20
   ;CTC mode, prescaler = 64, WGM = 0100 CS = 101
   
   L1:    IN R20,TIFR1 ;read in TIFR to find compare flag
   SBR R20,OCF1A ;skip next instruction if OCF1A = 1
   RJMP L1
   LDI R16,1<<OCF1A ;R16 = HEX value of ocf1a
   OUT TIFR1,R16 ;clear OCF1A (xor w/r16 to toggle)
   
   ;load second pulse of duty cycle
   LDI R20,0x3C
   STS OCR1AH, R20
   LDI R20,0x6D
   STS OCR1AL, R20
   LDI R20,0x0B
   STS TCCR1B,R20 ;CTC mode, prescaler = 64 CS = 101, WGM = 0100
   LDI R20,0x40
   STS TCCR1A,R20 ;TOGGLE on match COM1A = 01
   
   L2:    IN R20,TIFR1
   SBR R20,OCF1A ;skip next instruction if OCF1A = 1
   RJMP L2
   LDI R16,1<<OCF1A
   OUT TIFR1,R16 ;toggle OCF1A (same as before)
   RJMP BEGIN
   ```

Post-Lab Deliverables:

1) Submit a copy of your code from each of the above parts (a-d)
   a. Comment you code extensively
   b. Include calculations/justifications of your values (OCR TCCR)
2) Answer to attached questions
3) Altium PCB and Netlist of you circuit