ECE 376 - Test #1: Name _____

1) **Digital Inputs.** A light sensor has the following resistance vs lumens:

$$R = 10,000 \cdot (L)^{-0.6}\,\Omega$$

where L is the light level in lumens. Design a circuit which outputs:

- +5V when L < 10 lumens
- 0V when L > 15 lumens
- No change for 10 lumens < L < 15 lumens

R1 900 + 100(Birth Month) + Birth Date ex: May 14 = 1414 Ohms	R1 = 1514 Ohms

10 Lumens (on)

- R = 2511.88
- Vx = 3.1197V

15 Lumens (off)

- R = 1969.45 Ohms
- Vx = 2.8269V

V(on) > V(off)

- connect to the plus input

Offset = V(on)

$$gain = \left(\frac{\text{change in output}}{\text{change in input}}\right) = \left(\frac{5V-0V}{3/1197V-2.8269V}\right) = 17.07$$
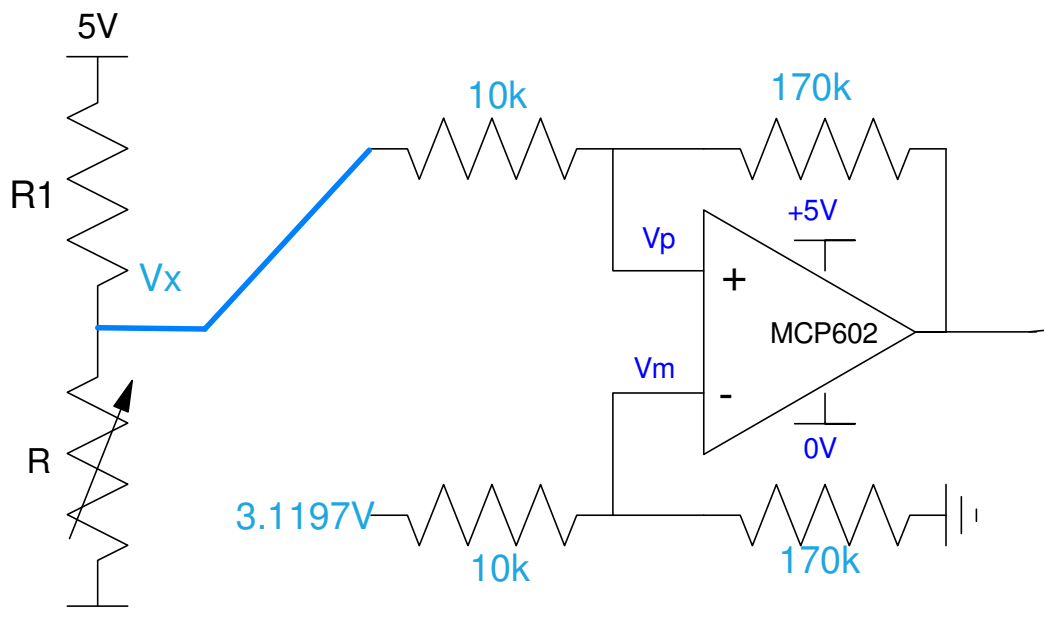
2) Digital Outputs: Determine Rb and Rc so that your PIC can drive a white 5W yellow LED at N mA where N is related to your birthday

- Vf = 2.4V @ 1200mA
- 600 Lumens @ 1200mA
- N = 900 + 100*(birth month) + (birth date).

Assume a 6144 NPN transistor

- Vbe = 700mV
- Vce(sat) = 360mV
- Current gain = β = 200

N mA 900 + 100*(Birth Month) + Birth Date ex: May 14th = 1414mA	Rb	Rc
1414 mA	430 Ohms	5.12 Ohms

Rc:

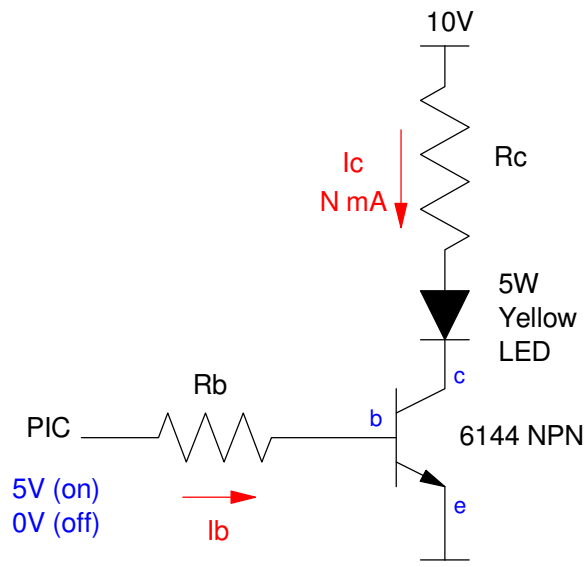$$R_c = \left(\frac{10V - 2.4V - 0.36V}{1414mA} \right) = 5.12\Omega$$

Rb:

$$\max(I_b) = 25mA \quad \text{(limit of PIC)}$$

$$\min(I_b) = \left(\frac{I_c}{\beta} \right) = \left(\frac{1414mA}{200} \right) = 7.07mA$$

Let Ib = 10mA

$$R_b = \left(\frac{5V - 0.7V}{10mA} \right) = 430\Omega$$

3) Assembler: Determine the contents of the W, PORTB, and PORTC registers after each operation. Assume

- PORTB and PORTC are output.
- Default is decimal

	W	PORTB	PORTC
Start:	5 *Birth Month*	0x23 = 35	14 *Birth Date*
movf PORTC,W	14	35	14
movff PORTB, PORTC	14	35	35
movwf PORTB	14	14	35
movlw 7	7	14	35
xorwf PORTB,F	7	9	35
btg PORTC,0	7	9	34
negf PORTB, W	-9	9	34

4) Assembler & Timing: Determine the number of clocks the following assembler subroutine takes to execute. Assume MONTH and DAY be your birth month and day.

MONTH (birth month: 1..12)	DAY (birth day: 1..31)	N Number of clocks Wait routine takes
5	14	149,350
Find A / B / C for N = 22,000,000 +/- 500,000 (2.2 seconds +/- 0.05 second)		
A = 255	B = 250	C = 28 N = 21,995,540 (off by 4460)

$$N = 12ABC + 9AB + 7A + 5$$

$$N = 149,350$$

```
Wait:
    movlw    MONTH (A)                              5
    movwf    CNT2
  nop
W2:
        movlw    DAY (B)                            7 * A
        movwf    CNT1
        nop
        nop
W1:
            movlw    177 (C)                        9 * A * B
            movwf    CNT0
            nop
            nop
            nop
            nop
W0:
                nop                                 12 * A * B * C
                nop
                nop
                nop
                nop
                nop
                nop
                nop
                nop
                decfsz CNT0,F
                goto   W0

            decfsz  CNT1,F
            goto    W1

        decfsz    CNT2,F
        goto W2
    return
```

5) Assembler & Flow Charts. Write an assembler program to turn your PIC processor into a combination lock

- Press RB0 (PORTB pin 0) N times, (number of presses is the combination) then
- Press RB7 (PORTB pin 7) one time to try to open the lock

If you pressed RB0 42 times, the lock opens (PORTD = 255)

Otherwise, the lock remains closed (PORTD = 0) and the count starts over (PORTC = 0)

```
#include <pic18f4620.inc>

      org   0x800
      movlw 0x0F
      movwf ADCON1

      movlw     0xFF
      movwf     TRISB
      clrf      TRISC
      clrf      TRISD
L1:
      clrf      PORTC
      clrf      PORTD
L2:
      movlw     0
      cpfseq    PORTB
      goto      L2
L3:
      movlw     0
      cpfseq    PORTB
      goto      L4
      goto      L3
L4:
      btfsc     PORTB,0
L5:
      incf      PORTC,F
L6:
      btfss     PORTB,7
      goto      L2
L7:
      movlw     42
      cpfseq    PORTC
      goto      L1
L8:
      movlw     0xFF
      movwf     PORTD
      call      Wait
      goto      L1
```
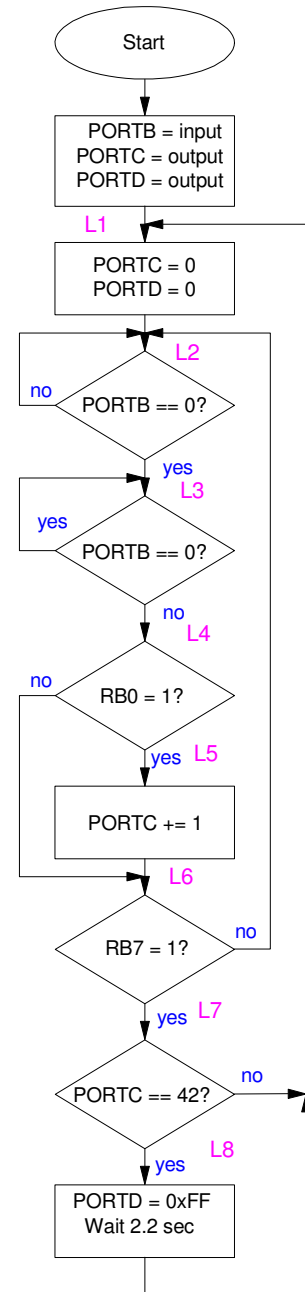
Memory Read & Write		
MOVWF PORTA	memory write	w → PORTA
MOVFF PORTA PORTB	copy	PORTA → PORTB
MOVF PORTA,W	memory read	PORTA → W
MOVLW 234	Move Literal to WREG	123 → W
Memory Clear, Negation		
CLRF PORTA	clear memory	0x00 → PORTA
COMF PORTA, W	toggle bits	!PORTA → W (bit toggle)
NEGF PORTA, W	negate	−PORTA → W (2's compliment)
Addition & Subtraction		
INCF PORTA,F	increment	PORTA + 1 → PORTA
ADDWF PORTA, F	add	PORTA + W → PORTA
ADDWFC PORTA, W	add with carry	PORTA + W + carry → W
ADDLW	Add Literal and WREG	
DECF PORTA,F	decrement	PORTA −1 → PORTA
SUBFWB PORTA,F	subtract with borrow	PORTA − W − c → PORTA
SUBWF PORTA,F	subtract no borrow	PORTA − W → PORTA
SUBWFB PORTA,F	subtract with borrow	PORTA − W − c → PORTA
SUBLW 223	Subtract WREG from #	223 − W → W
Shift left (*2), shift right (/2)		
RLCF PORTA,F	rotate left through carry (9−bit rotate)	
RLNCF PORTA,F	rotate left no carry	
RRCF PORTA,F	rotate right through carry	
RRNCF PORTA,F	rotate right no carry	
Bit Operations		
BCF PORTA, 3	Bit Clear f	clear bit 3 of PORTA
BSF PORTA, 4	Bit Set f	set bit 4 of PORTA
BTG PORTA, 2	Bit Toggle f	toggle bit 2 of PORTA
Logical Operations		
ANDWF PORTA, F	logical and	PORTA = PORTA and W
ANDLW 0x23	AND Literal with WREG	W = W and 0x23
IORWF PORTA,F	logical or	PORTA = PORTA or W
IORLW 0x23	Inclusive OR Literal	W = W or 0x23
XORWF PORTA,F	logical exclusive or	PORTA = PORTA xor W
XORLW 0x23	Exclusive OR Literal	W = W xor 0x23
Tests (skip the next instruction if...)		
CPFSEQ PORTA	Compare PORTA to W, skip if PORTA = W	
CPFSGT PORTA	Compare PORTA to W, Skip if PORTA > W	
CPFSLT PORTA	Compare PORTA to W, Skip if PORTA < W	
DECFSZ PORTA,F	decrement, skip if zero	
DCFSNZ PORTA,F	decrement, skip if not zero	
INCFSZ PORTA,F	increment, skip if zero	
INFSNZ PORTA,F	increment, skip if not zero	
BTFSC PORTA, 5	Bit Test f, Skip if Clear	
BTFSS PORTA, 1	Bit Test f, Skip if Set	
Flow Control		
GOTO Label	Go to Address 1st word	
CALL Label	Call Subroutine 1st word	
RETURN	Return from Subroutine	
RETLW 0x23	Return with 0x23 in WREG	