ECE 376 - Test #1: Name _____

1) **Digital Inputs.** Design a circuit which outputs

- 0V when R > 800 Ohms
- 5V when R < 700 Ohms

Assume

- R1 = 900 + 100*(your birth month) + (your birth date).
- May 14th, for example, gives R1 = 1414 Ohms

R = 800 Ohms (Y = 0V)

$$V_1 = \left(\frac{800}{800+1414}\right)5V = 1.8067V$$

R = 700 Ohms (Y = 5V)

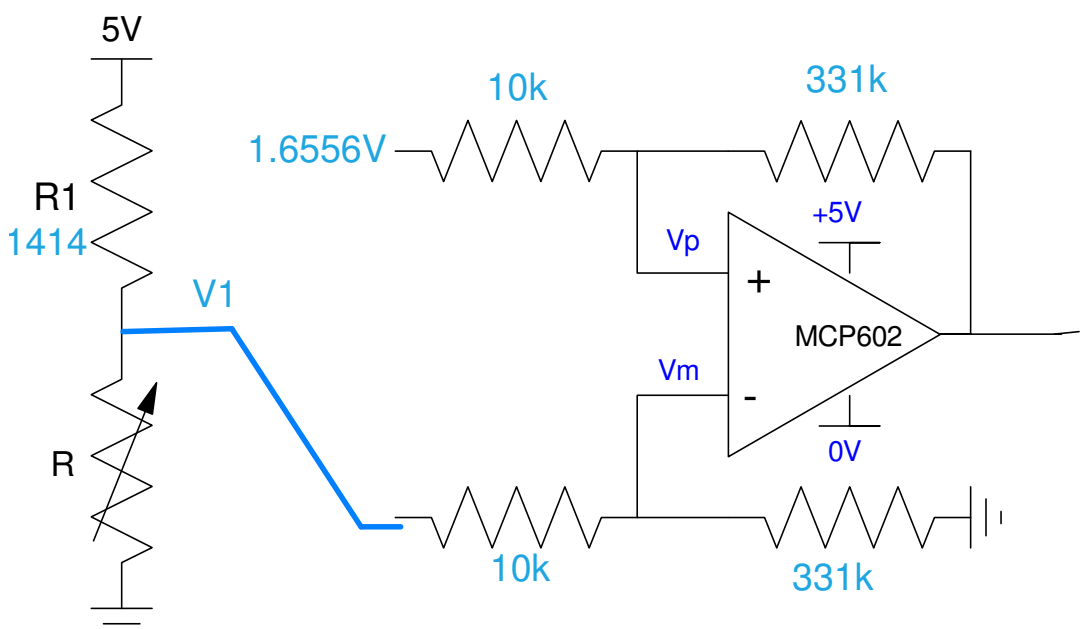
$$V_1 = \left(\frac{700}{700+1414}\right)5V = 1.6556V$$

Output goes up as the input goes down. Connect to the minus input

Output goes high when V1 = 1.6556V. Make the offset 1.6556V

Gain needed is

$$gain = \left(\frac{5V-0V}{1.8067V-1.6556V}\right) = 33.10$$



2) Digital Outputs: Design a circuit which allows your PIC to drive a 100W LED at N mA

- N = 900 + 100*(your birth month) + (your birth date)
- N = 1414 mA for May 14th, for example

Assume a 100W LED has the following characteristics

- Vf = 38V @ 3000mA
- 10,000 Lumens @ 3000mA

Assume a 6144 NPN transistor

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

mcd	Ic (mA) 900 + 100*Month + Day	Rb	Rc
	1414 mA		

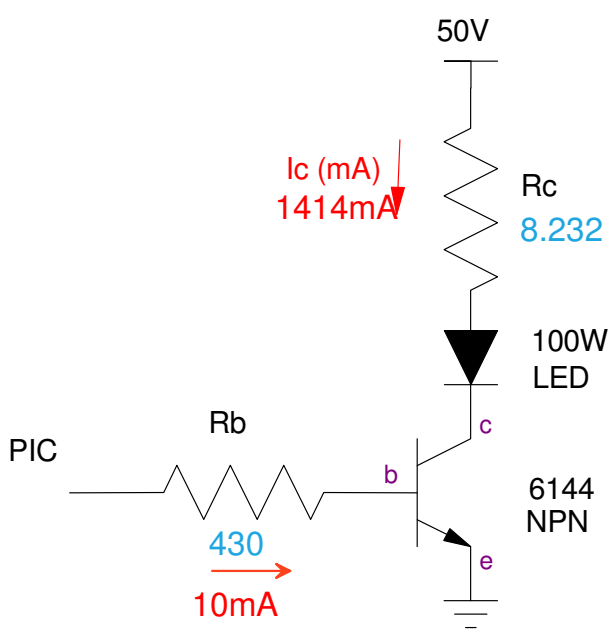
$$mcd = \left(\frac{1414mA}{3000mA}\right) \cdot 10,000Lumen = 4713Lumen$$

$$R_c = \left(\frac{50V - 38V - 0.36V}{1414mA}\right) = 8.232\Omega$$

$$I_b > \frac{I_c}{200} = 7.07mA$$

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:	Birth Month (1..12) 5	Birth Date (1..31) 14	15
`addwf PORTC,F`	5	14	**20**
`subwf PORTB,W`	**9**	14	20
`decf PORTB,F`	9	**13**	20
`incf PORTC,W`	**21**	13	20
`btg PORTB,0`	21	**12**	20
`iorlw 12`	**29**	12	20
`andlw 9`	**9**	12	20
`iorlw 7`	**15**	12	20
`negf PORTB,F`	15	**−12 = 244**	20
`comf PORTC,F`	15	−12 = 244	**−21 = 235**

4) Assembler & Timing:

a) Determine the number of clocks the following assembler subroutine takes to execute.

- Assume MONTH and DAY be your birth month and day.

b) Modify this routine (change A, B, and C) so that it takes 22,000,000 clocks (2.2 seconds) to execute

- +/- 20,000 clocks

A	Month birth month 1..12	Day birth date 1..31	N number of clocks Wait takes
200	5	14	8ABC + 8AB + 8A + 6 **N = 121,606**
A	B	C	N
43	**255**	**250**	**N = 22,018,070** +/- 20,000

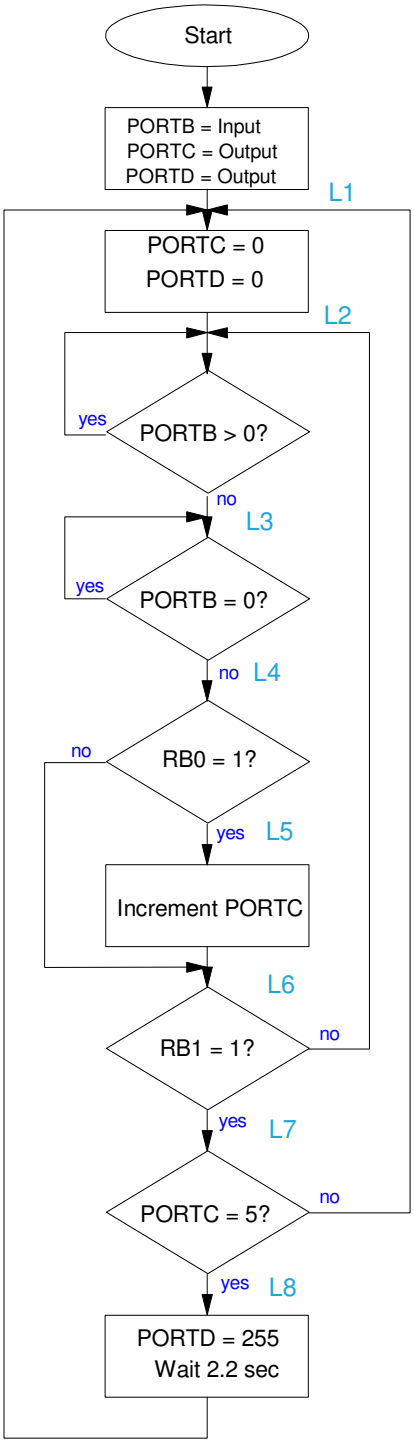
```
Wait:
     movlw     200 (A)        6
     movwf     CNT2
      nop
      nop
W2:
          movlw     MONTH    (B)    8 * 200
          movwf     CNT1
          nop
          nop
          nop

W1:
               movlw     DAY (C)   9 * 5 * 200
               movwf     CNT0
               nop
               nop
               nop
               nop
W0:
                    nop
                    nop       8 * 14 * 5 * 200
                    nop
                    nop
                    nop
                    decfsz CNT0,F
                    goto   W0

               decfsz  CNT1,F
               goto    W1

          decfsz    CNT2,F
          goto W2

     return
```

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

- Press RB0 five times, then
- Press RB1 one time

Results in PORTD lights turning on for 2.2 seconds (door unlocked)

```
     movlw     0xFF
     movwf     TRISB
     clrf      TRISC

L1:

     clrf      PORTC
     clrf      PORTD

L2:
     movlw     0
     cpfsgt    PORTB
     goto      L3
     goto      L2

L3:
     movlw     0
     cpfseq    PORTB
     goto      L4
     goto      L3

L4:
     btfsc     PORTB,0

L5:
     incf      PORTC,F

L6:
     btfss     PORTB,1
     goto      L2

L7:
     movlw     5
     cpfseq    PORTC
     goto      L1
L8:
     movlw     255
     movwf     PORTD

     call      Wait

     goto      L1
```



Bonus: (Due Monday 2pm): Program and demonstrate problem #5 on yor PIC board

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	