ECE 376 - Test #1: Name _____

Open book, open notes. Calculators permitted.
Individual effort (help from other people or web sites where other people help you solve the problems not permitted)

1) **Digital Inputs.** A thermistor has the following resistance vs temperature:

$$R = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

where T is the temperture in degrees Celsius. Design a circuit which outputs:

- +5V when T > 30C
- 0V when T < 25C
- No change for 25C < T < 30C

R1 1000 + 100(Birth Month) + Birth Date ex: May 14 = 1514 Ohms	**R1 = 1514**

30C (Vout = 5V)

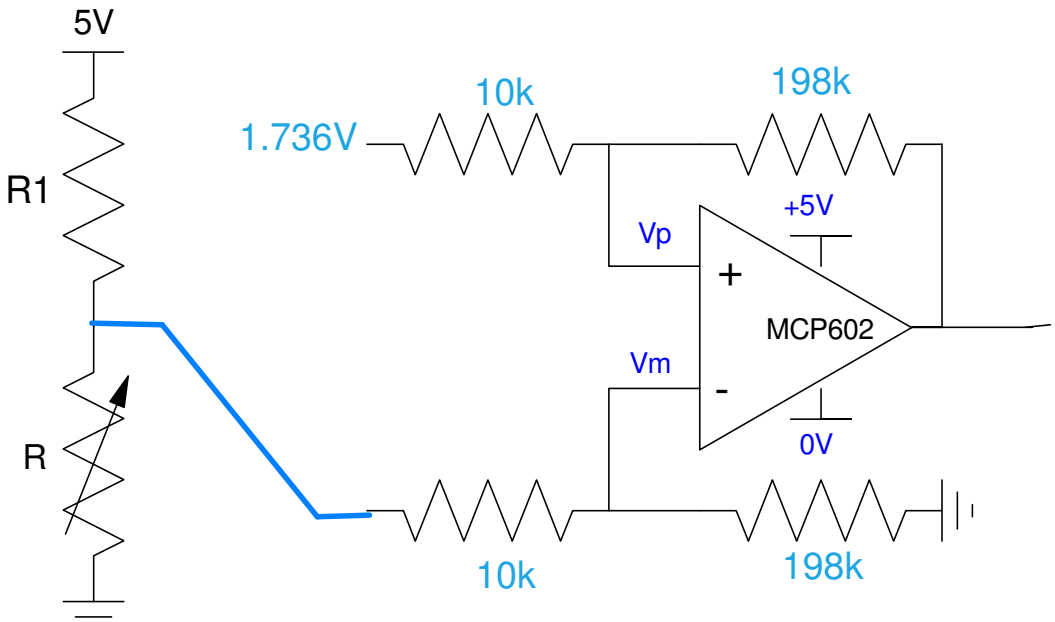
- R = 805.5 Ohms
- V1 = 1.736V

25C (Vout = 0V)

- R = 1000 Ohms
- V1 = 1.989V

Offset is the voltage where Vout goes high (1.736V)

$$gain = \left(\frac{5V-0V}{1.989V-1.736V}\right) = 19.807$$



2) Digital Outputs: Determine Rb and Rc so that your PIC can drive a white 3W Star LED at N mA

- Vf = 3.6V @ 750mA
- 200 Lumens @ 750mA

Assume a 6144 NPN transistor

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

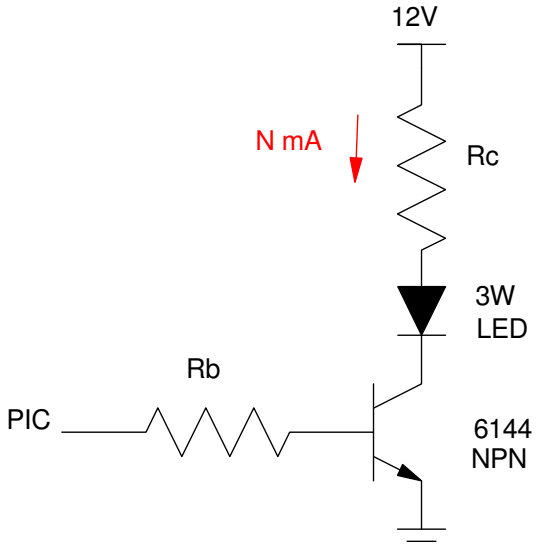
N mA 100*(Birth Month) + Birth Date ex: May 14th = 514mA	Rb	Rc
514mA	**1000 Ohms** 172 < Rb < 1637	**15.64 Ohms**

$$R_c = \left(\frac{12V - 3.6V - 0.36V}{514mA}\right) = 15.64\Omega$$

$$\frac{514mA}{200} = 2.57mA < I_b < 25mA$$

$$\left(\frac{5V - 0.7V}{2.57mA}\right) > R_b > \left(\frac{5.0V - 0.7V}{25mA}\right)$$

$$1637\Omega > R_b > 172\Omega$$



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:	**0**	Birth Month (1..12) **5**	Birth Date (1..31) **14**
movlw 12	**12**	**5**	**14**
addwf PORTB,F	**12**	**17**	**14**
subwf PORTC,W	**2**	**17**	**14**
andlw 0x0F	**2**	**17**	**14**
incf PORTB,W	**18**	**17**	**14**
decf PORTC,F	**18**	**17**	**13**
nop	**18**	**17**	**13**
comf PORTB,W	**238 (-18)**	**17**	**13**
negf PORTC,F	**238**	**17**	**243 (-13)**

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**	**60,102**

```
Wait:
     movlw     MONTH (5) (A=58)                    7 clocks
     movwf     CNT2
      nop
      nop
      nop
W2:
          movlw     DAY (14) (B=252)               7 clocks * 5
          movwf     CNT1
          nop
          nop
W1:
               movlw     213 (C=255)               6 clocks * 14 * 5
               movwf     CNT0
               nop
W0:
                    nop                            4 clocks * 213 * 14 * 5
                    decfsz CNT0,F
                    goto   W0

               decfsz  CNT1,F
               goto    W1

          decfsz    CNT2,F
          goto W2

return
```

Modify this routine so that it takes 15,000,000 clocks (1.5 seconds) to execute (+/- 50,000 clocks)

4*A*B*C + 6*A*B + 7*A + 7 = 15,000,000

Assume B = 255,CB = 255. Solve for A

A = 57.33

Assume A = 58, C = 255. Solve for B

B = 252.06

Let A = 58, B = 252, C = 255

N = 4*A*B*C + 6*A*B + 7*A + 7

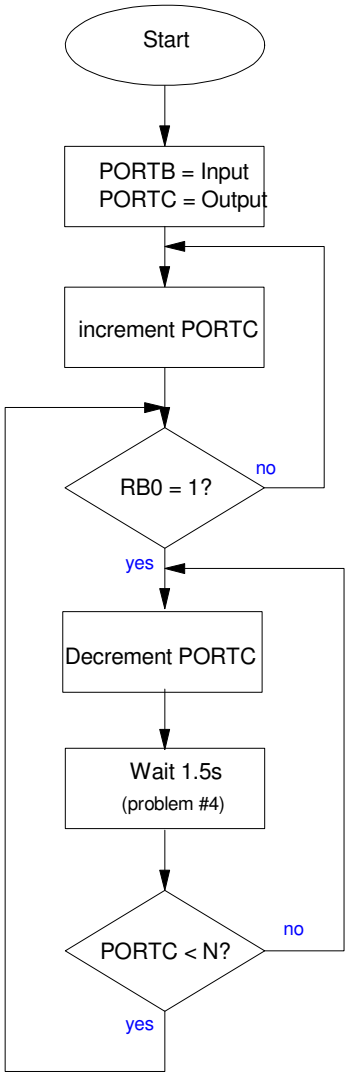
N = 14,996,429

5) Assember & Flow Charts. Write an assembler program for an random count-down timer.

- Let N be Your Birth Date (1..31).
- When you press RB0 (PORTB pin 0), a random number (0..255) is placed in PORTC
- The counter then counts down, one count every 1.5 seconds (i.e. problem #4), until PORTC < N
- It then repeats, waiting for you to press RB0

N Birth Date (1..31)	N =

```
#include <pic18f4620.inc>
TOTAL  EQU 0

     org   0x800
     movlw 0x0F
     movwf ADCON1

     movlw     0x0F
     movwf     TRISB
     clrf      TRISC
L1:
     incf      PORTC,F

L2:
     btfss     PORTB,0
     goto      L1

L3:
     decf      PORTC,F

     call      Wait

     movlw     14
     cmfslt    PORTC
     goto      L3

     goto      L2
```



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	