ECE 376 - Test #1: Name _____

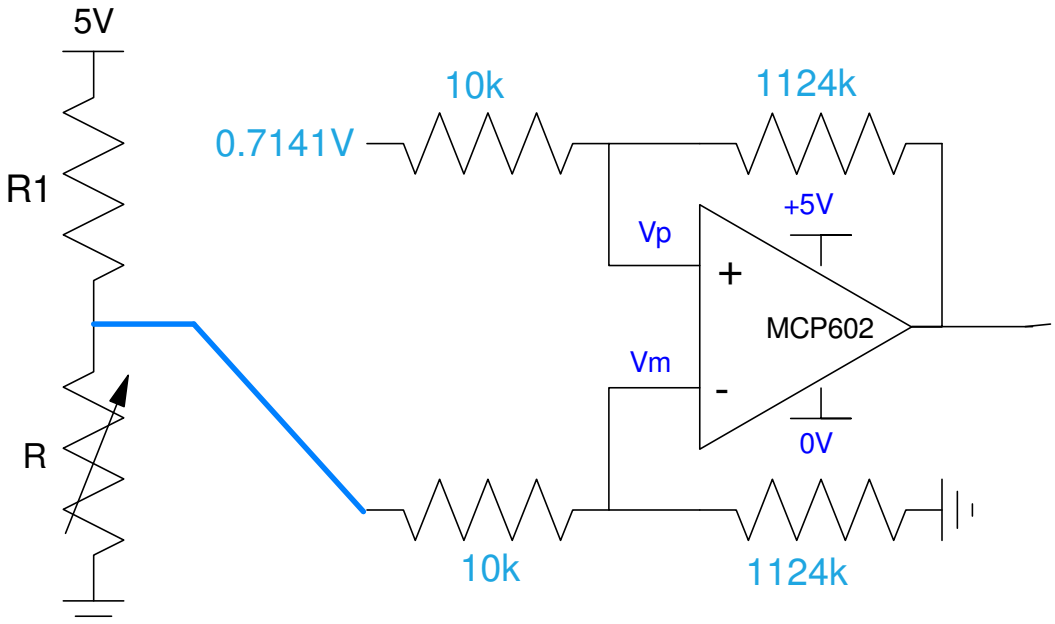
1) **Digital Inputs.** A thermistor has the following resistance vs temperature:

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

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

- +5V when T > 60C
- 0V when T < 58C
- No change for 58C < T < 60C

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



At 58C (output = 0V)

- R = 270.78 Ohms
- V = 0.7586V

at 60C (output = 5V)

- R = 252.26 Ohms
- V = 0.7141V

Offset 4.2859V

$$gain = \left(\frac{5V-0V}{0.7586V-0.7241V}\right) = 112.4$$

2) Digital Outputs: Determine Rb and Rc so that your PIC can drive a white 10W LED at N mA

- Vf = 10.0V @ 1A, 650 Lumens @ 1A

Assume a 6144 NPN transistor

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

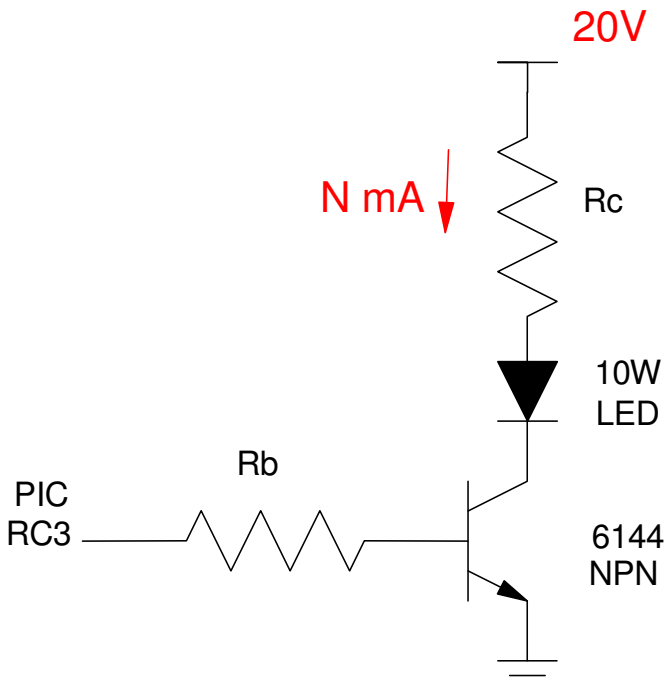
N mA 1000 + 100*(Birth Month) + Birth Date ex: May 14th = 1514mA	Rb	Rc
1514	**172 .. 568 Ohms**	**6.36 Ohms**

$$R_c = \left(\frac{20V - 10.0V - 360mV}{1514mA}\right) = 6.36\Omega$$

$$\frac{1514mA}{200} < I_b < 25mA$$

$$R_b = \left(\frac{5V - 0.7V}{I_b}\right)$$

$$568\Omega > R_b > 172\Omega$$



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**
movf PORTC,W	**14**	**5**	**14**
addwf PORTB,W	**19**	**5**	**14**
addwfc PORTC,F	**19**	**5**	**33**
andlw 0x05	**1**	**5**	**33**
movff PORTB,PORTC	**1**	**5**	**5**
bsf PORTB,4	**1**	**21**	**5**
btg PORTC,1	**1**	**21**	**7**

```
19 = 0001 0011
 5 = 0000 0101
---------------
and  0000 0001


5 = 0000 0101
       ^
      set

5 = 0000 0000
          ^
        toggle
```

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**	**69,154**

N = 6*DAY*MONTH*150 + 7*MONTH*150 + 6*150 + 4

N =

```
Wait:
     movlw     150      A = 53
     movwf     CNT2
W2:
          movlw     MONTH      B = 250
          movwf     CNT1
          nop
W1:
               movlw     DAY      C = 250
               movwf     CNT0
               nop
               nop

W0:
                    nop
                    nop
                    nop
                    decfsz CNT0,F
                    goto   W0

               decfsz  CNT1,F
               goto    W1

          decfsz    CNT2,F
          goto W2
     return
```

Modify this routine so that it takes 20,000,000 clocks (2 seconds) to execute (+/- 100,000 clocks)

- A = 53
- B = 250
- C = 251
- N = 20,047,572

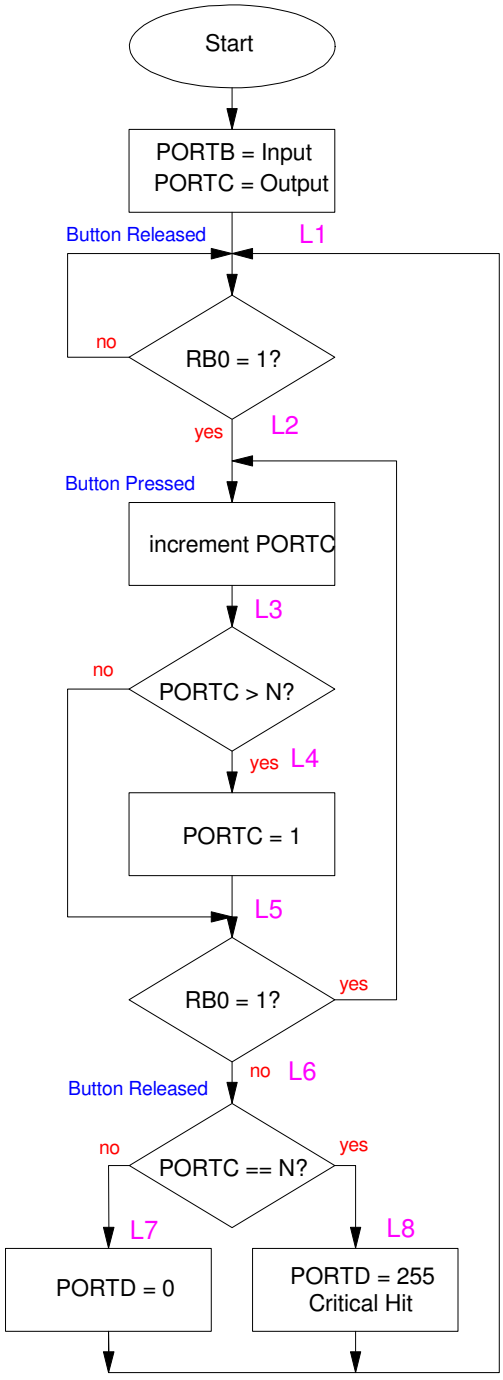
Other solutions exist

5) Assember & Flow Charts. Write an assembler program for an electronic N sided die

- Let N be Your Birth Date plus one (1..31 + 1). For example, if you were born on May 14th, N=15
- PORTC outputs a random number from 1..N when you press and release PortB button 0 (RB0)
- PORTD lights turn on if you rolled the maximum number, N (critical hit)

N Birth Date + 1 ex: May 14 = 15 (15 sided die)	**N = 15**

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

     org   0x800
     movlw 0x0F
     movwf ADCON1
     movlw     0xFF
     movwf     TRISB
     clrf      TRISC

L1:

     btfss     PORTB,0
     goto      L1

L2:
     incf PORTC,F

L3:
     movlw     15
     cpfsgt    PORTC
     goto      L5

L4:
     movlw     1
     movwf     PORTC

L5:
     btfsc     PORTB,0
     goto      L2

L6:
     movlw     15
     cpfseq    PORTC
     goto      L7

L8:
     movlw     255
     movwf     PORTD
     goto      L1

L7:
     clrf      PORTD
     goto      L1

     end
```



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