ECE 376 - Test #1: Name ____________________

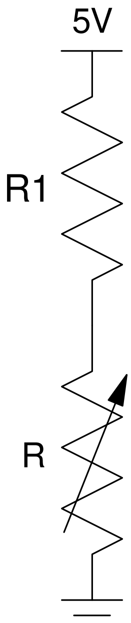
Spring 2023

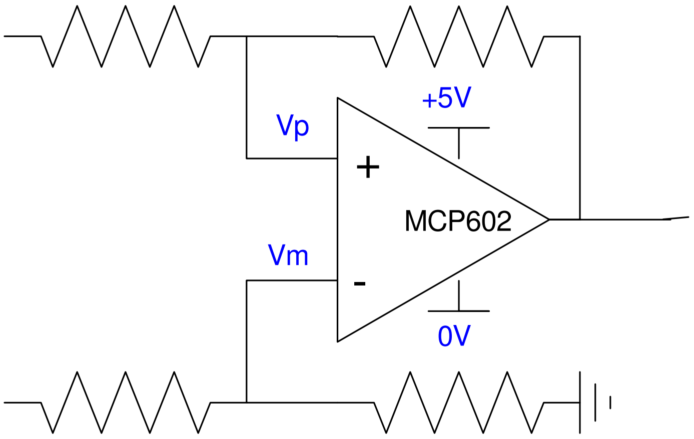
1) **Digital Inputs.** Design a circuit which outputs

- 0V when R < 1500 Ohms
- 5V when R > 1700 Ohms

Assume

- R1 = 800 + 100*(your birth month) + (your birth date).
- May 14th, for example, gives R1 = 1314 Ohms





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

- N = 800 + 100*(your birth month) + (your birth date)
- N = 1314 mA for May 14th, for example

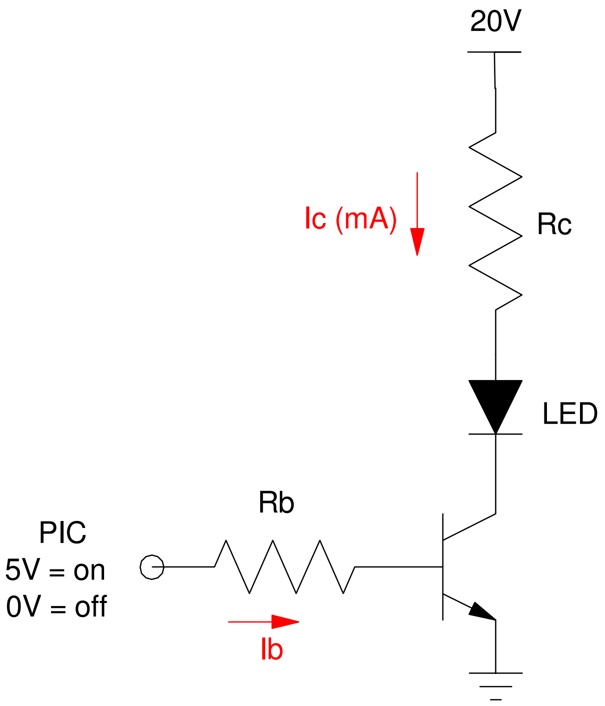
Assume a 20W LED has the following characteristics

- Vf = 10V @ 2000mA
- 2,000 Lumens @ 2000mA

Assume a 6144 NPN transistor

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

Ic (mA) 800 + 100*Month + Day	Lumens Light output when on	Rb	Rc



3) **Assembler:** Determine the contents of the W, A, and B after each operation. Assume

- A and B are 8-bit registers (spots in memory).
- Default is decimal

	W	A	B
Start:	13	Birth Month (1..12)	Birth Date (1..31)
`incf A,W`			
`decf B,W`			
`addlw 5`			
`addwf A,F`			
`subwf B,W`			
`movf A,W`			
`movff A,B`			
`andlw 7`			
`btg A,1`			
`movwf B`			

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 2,500,000 clocks (250ms) to execute

- +/- 50,000 clocks

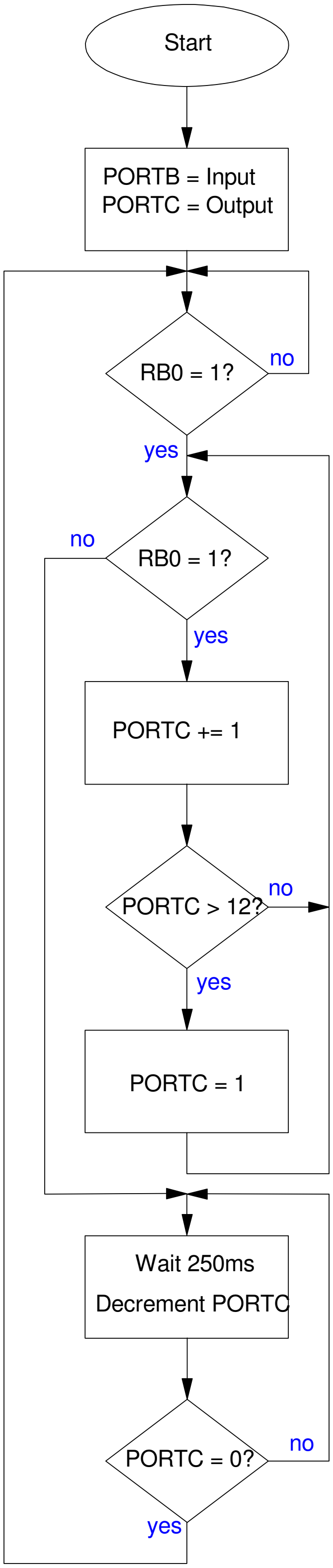
A	Month birth month 1..12	Day birth date 1..31	N number of clocks Wait takes
150			
A	B	C	N 2,500,000 +/- 50,000

```
Wait:
     movlw     150  (A)
     movwf     CNT2
     nop
     nop
     nop
     nop
W2:
          movlw     MONTH    (B)
          movwf     CNT1
          nop
          nop

W1:
               movlw     DAY  (C)
               movwf     CNT0
W0:
                    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 random count-down timer

- When RB0 is pressed, PORTC counts from 1..12 really fast
- When RB0 is released, PORTC then contains a random number from 1..12
- PORTC then counts down every 250ms
- When PORTC reaches zero, it then goes back to the beginning and waits for RB0.

Assume a 250ms wait routine exists (call Wait)



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