

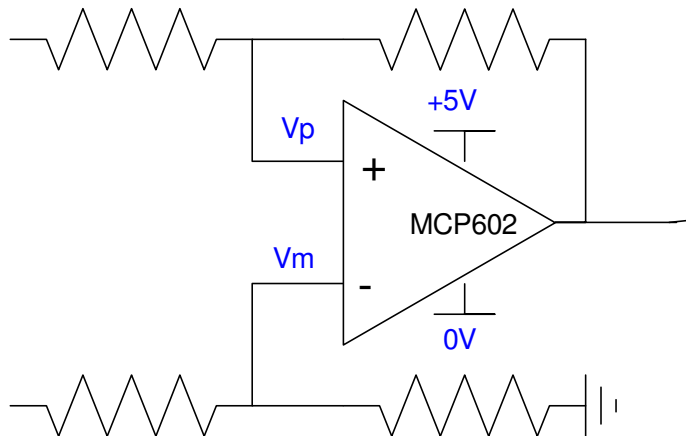
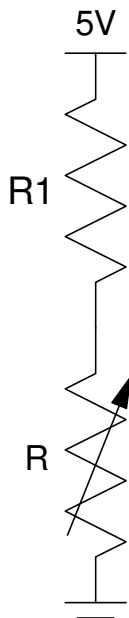
ECE 376 - Test #1: Name _____

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

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

Assume

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



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

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

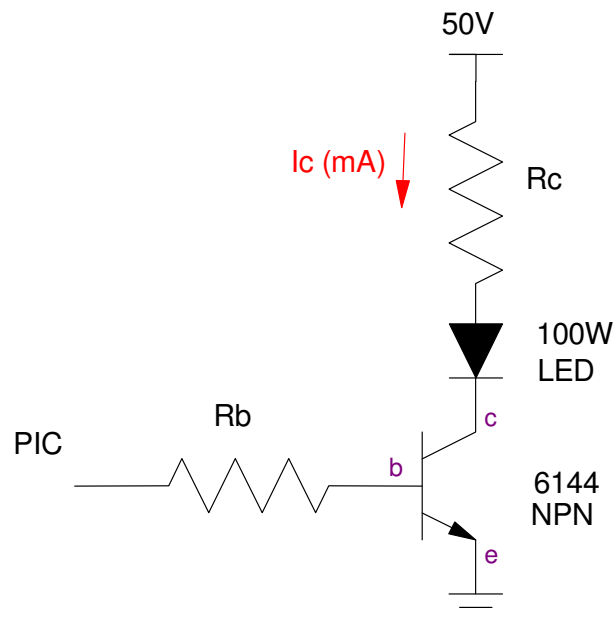
Assume a 100W LED has the following characteristics

- $V_f = 38\text{V} @ 3000\text{mA}$
- 10,000 Lumens @ 3000mA

Assume a 6144 NPN transistor

- $V_{be} = 700\text{mV}$
- $V_{ce(sat)} = 360\text{mV}$
- Current gain = $\beta = 200$

mcd	$I_c \text{ (mA)}$ $900 + 100 * \text{Month} + \text{Day}$	R_b	R_c
	1414 mA		



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)	
addwf PORTC,F			
subwf PORTB,W			
decf PORTB,F			
incf PORTC,W			
btg PORTB,0			
iorlw 12			
andlw 9			
iorlw 7			
negf PORTB,F			
comf PORTC,F			

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

- +/- 100,000 clocks (0.5%)

A	Month birth month 1..12	Day birth date 1..31	N number of clocks Wait takes
200			
A	B	C	N

Wait:

```
    movlw    200 (A)
    movwf    CNT2
    nop
    nop
```

W2:

```
    movlw    MONTH (B)
    movwf    CNT1
    nop
    nop
    nop
```

W1:

```
    movlw    DAY (C)
    movwf    CNT0
    nop
    nop
    nop
    nop
```

W0:

```
    nop
    nop
    nop
    nop
    nop
    decfsz   CNT0,F
    goto     W0
```

```
    decfsz   CNT1,F
    goto     W1
```

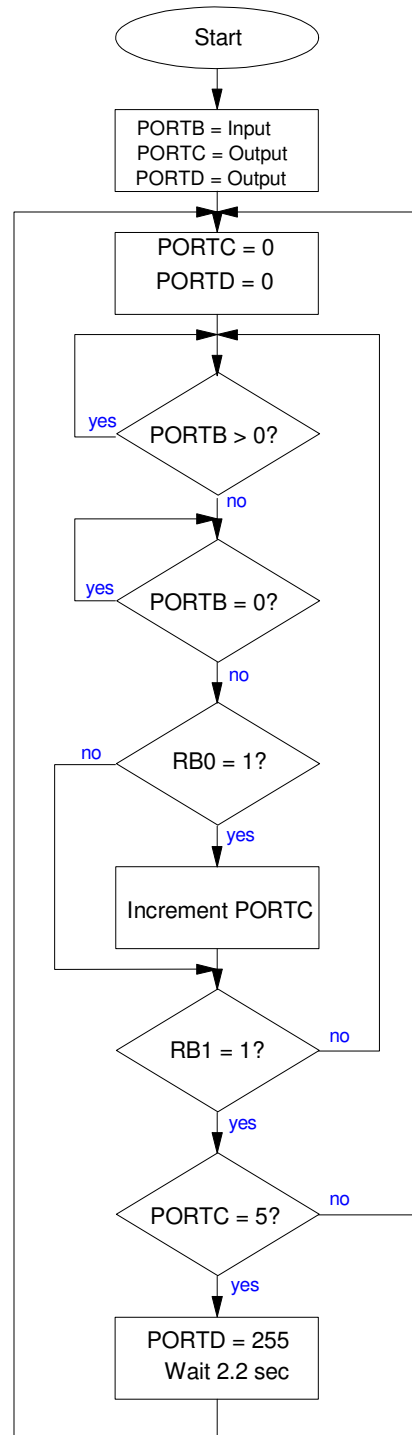
```
    decfsz   CNT2,F
    goto     W2
```

return

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

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

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



Bonus: (Due Monday 2pm): Program and demonstrate problem #5 on your 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	
BTFSZ	PORTA, 5	Bit Test f, Skip if Clear	
BTSS	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	