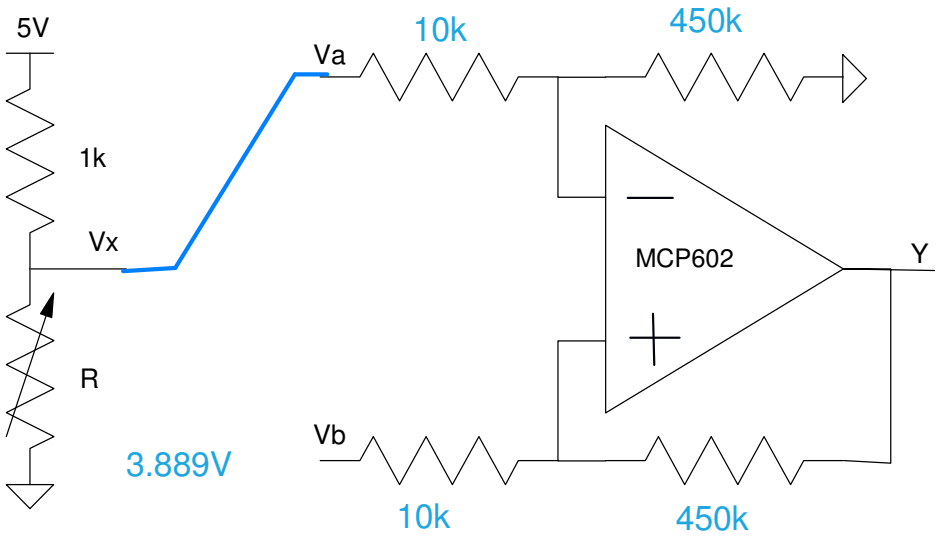
ECE 376 - Final: Name _____

Calculators Permitted.

1) Binary Input: Schmitt Trigger. Design a circuit which outputs
   - 0V when R = 4000 Ohms, and
   - 5V when R = 3500 Ohms



At 4000 Ohms (Y = 0V)

$$V_x = \left(\frac{4000}{4000+1000}\right)5V = 4.00V$$

At 3500 Ohms (Y = 5V)

$$V_x = \left(\frac{3500}{3500+10000}\right)5V = 3.889V$$

V(on) < V(off)

Connect to the minus nput

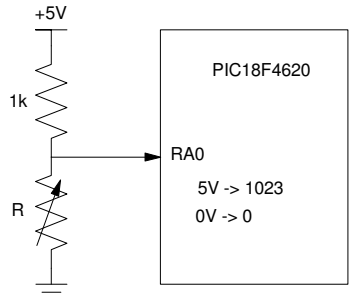
Offset = 3.889V

where the output goes high

$$gain = \left(\frac{5V-0V}{4.00V-3.889V}\right) = 45.00$$

2) Analog Input: A light sensor has the following resistance vs. light relationship

$$R = 10,000 \cdot \left(\frac{1}{Lux}\right)^{0.6} \Omega$$



2a) Determine the resistance, voltage, and A/D reading for the following circuit at 10 Lux and 1000 Lux

Lux	R	V	A/D
10	2511 Ohms	3.576V	732
1000	158 Ohms	0.684V	140

2b) Give a calibration function to compute the light level based upon the A/D reading

$$Lux = a \cdot A/D + b$$

Slope

$$a = \left(\frac{10-1000}{732-140}\right) = -1.67$$

Offset: Plug in a point

$$Lux = a \cdot A/D + b$$

$$10 = -1.67 \cdot 732 + b$$

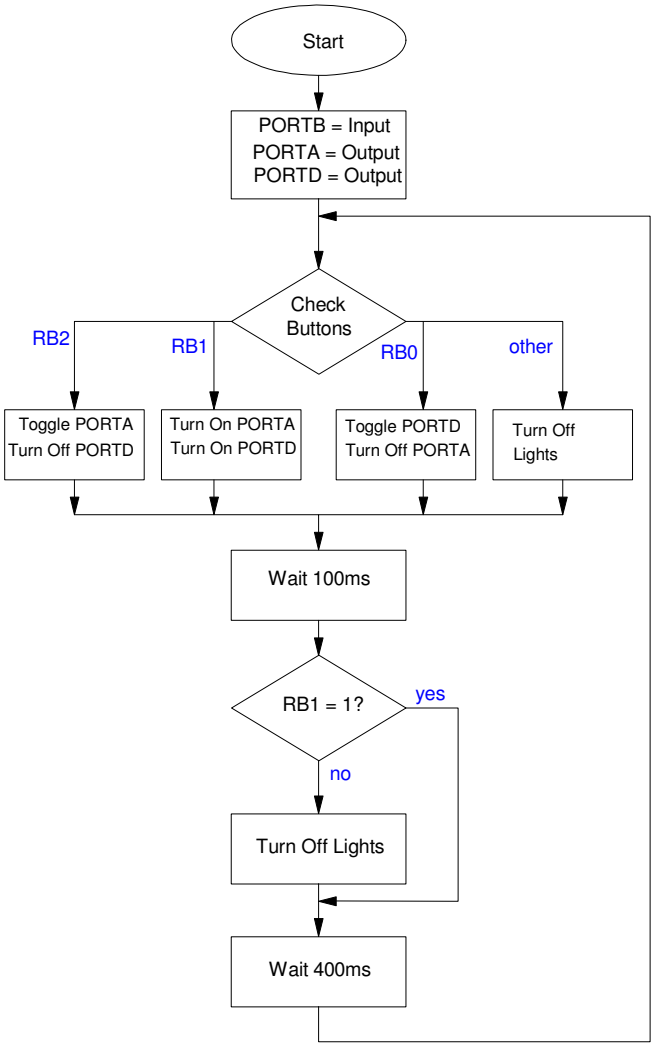
$$b = 1234$$

so

$$Lux \approx -1.67 \cdot A/D + 1234$$

3) C Coding: A PIC is to be used to control the tail lights of a car.

- When RB2 is pressed, the lights on PORTA blink (left turn)
- When RB0 is pressed, the lights on PORTD blink (right turn)
- When RB1 is pressed, both lights turn on (brake)
- When braking (RB1), the lights stay on for 500ms total. Otherwise, they're on for 100ms (dim)



Write the corresponding C code

```
void main(void) {

   TRISB = 0xFF;
   TRISC = 0;
   TRISD = 0;

   ADCON1 = 0x0F;  // no points off

   while(1) {

      if(RB2) {
         PORTA = !PORTA;
         PORTD = 0;
         }

      else if(RB1) {
         PORTA = 0xFF;
         PORTD = 0xFF;
         }

      else if(RB0) {
         PORTD = !PORTD;
         PORTA = 0;
         }

      else {
          PORTA = 0;
          PORTD = 0;
          }

      Wait_ms(100);

      if(!RB1) {
          PORTA = 0;
          PORTD = 0;
          }

      Wait_ms(400);

      }
```

4) C Subroutines Write subroutine which is passed two parameters and controls the tail lights of a car

- LEFT = 0: PORTA = 0000 0000
- LEFT = 1: PORTA = 0000 1111
- LEFT = 2: PORTA = 1111 1111
- RIGHT = 0: PORTD = 0000 0000
- RIGHT = 1: PORTD = 0000 1111
- RIGHT = 2: PORTD = 1111 1111

```
void TailLights(unsigned char LEFT, unsigned char RIGHT)
{

   if(LEFT == 0) PORTA = 0;
   if(LEFT == 1) PORTA = 0x0F;
   if(LEFT == 2) PORTA = 0xFF;

   if(RIGHT == 0) PORTD = 0;
   if(RIGHT == 1) PORTD = 0x0F;
   if(RIGHT == 2) PORTD = 0xFF;

   }
```

5) Interrupts: Write a program using interrupt to play a game of Hungry-Hungry Hippo where three players are playing with odds:

- Player A: Count every rising edge on RB0
- Player B: Count 90% of the rising edges on RB1
- Player C: Count 80% of the rising edges on RB2

a) Interrupt Set-Up

Specify which interrupt you're using and its initialization (pre-scalar, rising/falling edge, etc)

Interrupt	INT0	INT1	INT2	Timer0
Initialization	rising	rising	rising	PS=1

b) Interrupt Service Routines

Option #1: Using a random number generator

- X = 0 .. 99 based upon the time of the interrupt
- Better: harder to detect that B and C are missing counts

INT0 Player A counts every edge	INT1 Player B counts 90% of edges	INT2 Player C counts 80% of edges	TMR0IF

`if(INT0IF) {`
 	
`A = A + 1;`
 	
`INT0IF = 0;`
`}`
	`if(INT1IF) {`
	
	`X = TMR0 % 100;`
	
	`if(X >= 10)`
	`B = B + 1;`
	
	`INT1IF = 0;`
	
	`}`
		`if(INT2IF) {`
		
		`X = TMR0 % 100;`
		
		`if(X >= 20)`
		`C = C + 1;`
		
		`INT2IF = 0;`
		
		`}`
			`if(TMR0IF) {`
			`TMR0IF = 0;`
			`}`

Option #2: Using a counter

- Deterministic: skip every 10th edge (INT1) or 5th edge (INT2)

INT0 Player A counts every edge	INT1 Player B counts 90% of edges	INT2 Player C counts 80% of edges

`if(INT0IF) {`
 	
`A = A + 1;`
 	
`INT0IF = 0;`
`}`
	`if(INT1IF) {`
	
	`N1 = (N1 + 1)%10;`
	
	`if(N1 > 0)`
	`B = B + 1;`
	
	`INT1IF = 0;`
	
	`}`
		`if(INT2IF) {`
		
		`N2 = (N2 + 1)%5;`
		
		`if(N2 > 0)`
		`C = C + 1;`
		
		`INT2IF = 0;`
		
		`}`

6) Interrupts: A PIC is to be used to control the tail lights of a car. Use one or more interrupts to control the brightness of PORTA (left turn signal) and PORTD (right turn signal) using two global variables, LEFT and RIGHT

- LEFT = 0 to 100
  - 0 to 100% duty cycle on PORTA
- RIGHT = 0 to 100
  - 0 to 100% duty cycle on PORTD

Option #1: Using Timer0 called every 0.1ms and a counter

Timer0 set up for PS=1

```
void interrupt IntServe(void) {

   if(TMR0IF) {
      TMR0 = -1000;
      N = (N + 1) % 100;
      if(LEFT < N) PORTA = 0xFF;
      else PORTA = 0;
      if(RIGHT < N) PORTD = 0xFF;
      else PORTD = 0;
      TMR0IF = 0;
      }
```

Option #2: Using Timer0, Timer1, & Timer3

- Timer0 is called every 1ms
- Timer0 turns on the lights if PWM > 0
- Timer1 & 3 turn off the lights 100*X clocks in the future if PWM < 100

```
void interrupt IntServe(void) {
   if(TMR0IF) {
      TMR1 = -10000;
      if(LEFT) PORTA = 0xFF;
      if(RIGHT) PORTD = 0xFF;
      TMR1 = -LEFT*100;
      TMR3 = -RIGHT*100;
      TMR1IF = 0;
      }
   if(TMR1IF) {
      if(LEFT < 100) PORTA = 0;
      TMR1IF = 0;
      }
   if(TMR3IF) {
      if(RIGHT < 100) PORTD = 0;
      TMR3IF = 0;
      }
```