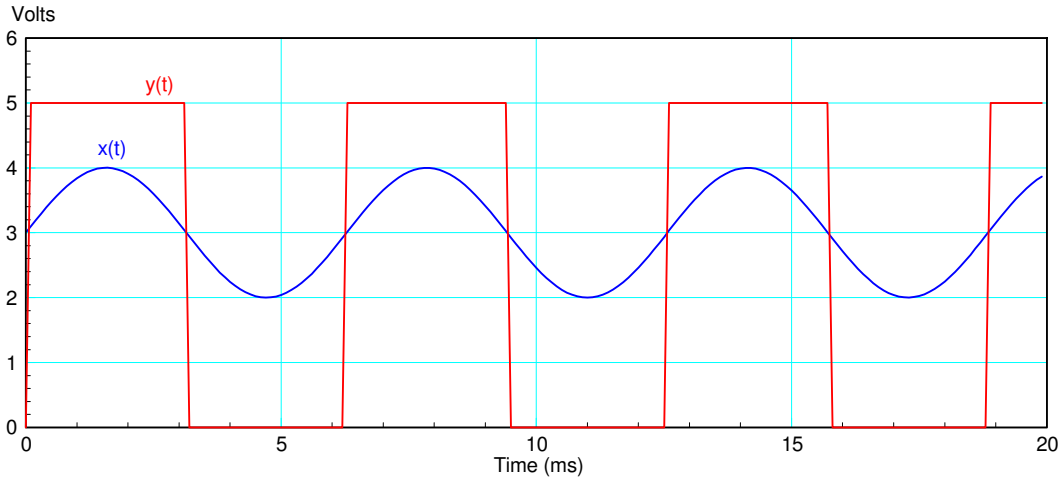
ECE 376 - Final Exam: Name _____

Open book. open notes

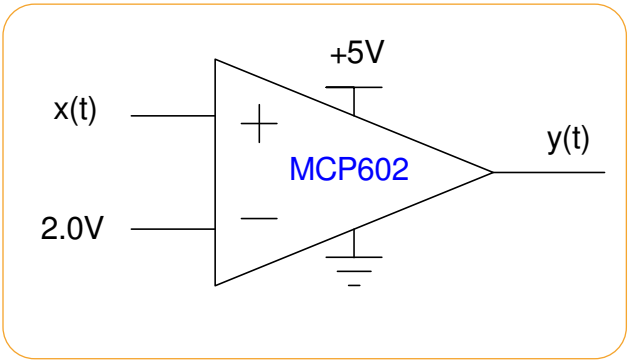
1) Hardware: Digital (0V/5V) signals are easy for a microcontroller to work with. Analog signals are more difficult. Design a circuit which converts a sine wave ranging from 2.0V to 4.0V into a TTL 0V/5V square wave at the same frequency as shown.

- Input: Analog sine wave going from 2.0V to 4.0V.
- Output: 0V/5V square wave



A comparitor or Schmitt trigger would work. Assume a comparitor:

- Send x(t) to the plus input
- Send 3.00V to the minus input (anything in the range of -2V to +4V works)



2) Hardware: Binary Outputs. A 5W yellow LED is to be used to create an electronic candle. Design a circuit which allows a PIC to turn on and off the LED

- Off: Id = 0mA
- On: Id = XmA where X is your birth date

Assume

- The LED has Vf = 5.0V @ 1A, Output = 5000 Lumens @ 1A
- A 6144 NPN transistor with β=300, Vce(sat) = 0.2V
- X = 900 + 100*(your birth month) + (your birth date). May 14th would give X = 1414mA.

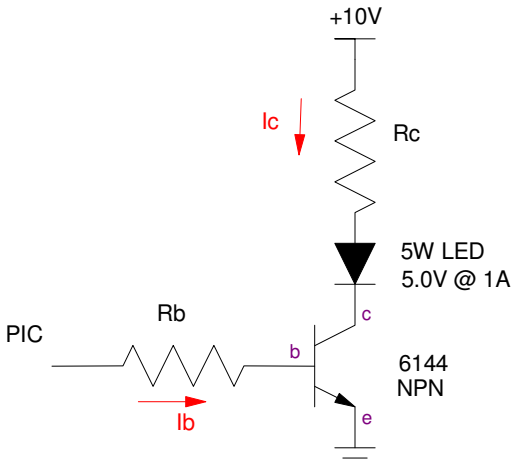
For your circuit, determine the following:

X (mA) 900 + 100*mo + day	Lumens when LED is on	Rc	Rb	Ib when the PIC outputs 5V
1414mA varies	7070 varies	3.3946 Ohms answers vary	430 Ohms	10mA

Lumens:

$$L = \left(\frac{1414mA}{1000mA}\right) 5000 \text{ lumens}$$

$$L = 7070 \text{ lumens}$$



Rc:

$$R_c = \left(\frac{10V - 5.0V - 0.2V}{1414mA}\right) = 3.3946\Omega$$

Ib: Pick Ib to be in the range of

- 25mA (max a PIC can output)
- $\left(\frac{1414mA}{300}\right) = 4.713mA$
- Let Ib = 10mA

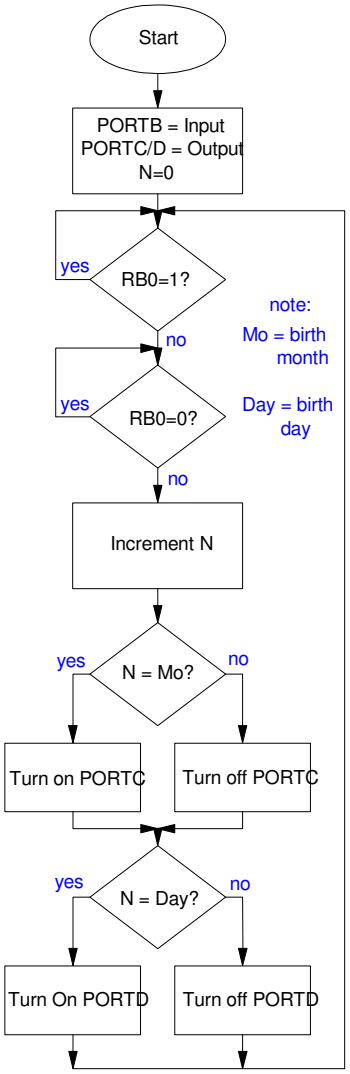
Rb is then

$$R_b = \left(\frac{5V - 0.7V}{10mA}\right) = 430\Omega$$

) C Coding. The following flow chart counts the number of times button RB0 is pressed:

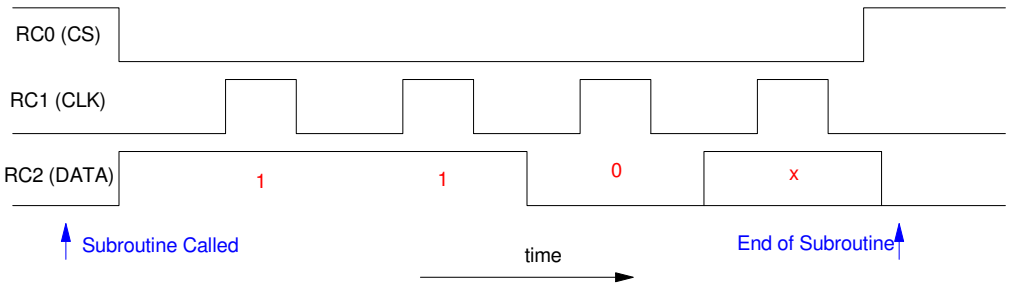
- N = the number of button presses
- When N matches your birth month (1..12), the lights on PORTC turn on
- When N matches your birth day (1..31), the lights on PORTD turn on

Write the corresponding C code




```
unsigned char N, Mo, Day;
Mo = 5;
Day = 14;

TRISB = 0xFF;
TRISC = 0;
TRISD = 0;
ADCON1 = 0x0F;

while(1) {

   while(RB0);

   while(!RB0);

   N += 1;

   if(N == Mo) PORTC = 0xFF;
   else PORTC = 0;

   if(N == Day) PORTD = 0xFF;
   else PORTD = 0;

   }
```

4) Subroutines & C Coding: The signals a PIC sends to a sensor are as follows. Write a C program which outputs the data 001x to the sensor where x is either 1 or 0 depending upon what is passed to the subroutine. Assume RC0 / RC1 / RC2 are all output pins




```
void Sensor(unsigned char x) {

    RC0 = 1;
    RC1 = 0;
    RC2 = 0;

    RC0 = 0;

    RC2 = 1;
    RC1 = 1;
    RC1 = 0;

    RC2 = 1;
    RC1 = 1;
    RC1 = 0;

    RC2 = 0;
    RC1 = 1;
    RC1 = 0;

    RC2 = x;
    RC1 = 1;
    RC1 = 0;

    RC0 = 1;

    }
```

5) Interrupts: Using interrupts

- Count how many times you press a button (detect rising edge on A)
- Clear the count when you press B
- When the count is equal to your birth month, play 100Hz for one second on RC0
- When the count is equal to yor birth date, play 200Hz for two seconds on RC1

Specify the interrupts used and how they're set up

Inputs		Outputs	
A count	B clear count	100Hz for 1 second	200Hz for 2 seconds
INT0	INT1	Timer0 (Hz) Timer2 (time)	Timer1 (Hz) Timer2 (time)

Write the corresponding interrupt service routines

Inputs		Outputs	
A count on A	B clear count on B	100Hz for 1 second	200Hz for 2 seconds

```
if(INT0IF) {
   N += 1;

   if(N == 5)
      Tc = 1000;

   if(N == 14)
       Td = 2000;

   INT0IF = 0;
   }
```

```
if(INT1IF) {
   N = 0;
   Tc = 0;
   Td = 0;
   INT1IF = 0;
   }
```

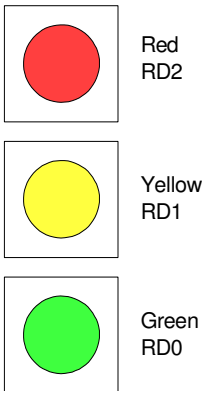
```
if( TMR2IF ) {
   if(Tc) Tc -= 1;
   if(Td) Td -= 1;
   TMR2IF = 0;

if(TMR0IF) {
   TMR0 = -50000;
   if(Tc) RC0 = !RC0;
   TMR0IF = 0;
   }
```

```
if(TMR1IF) {
   TMR1 = -25000;

   if(Td) RC1 = !RC1;
   TMR1IF = 0;
   }
```

6) Interrupt Stoplight: Write the interrupt routines to drive a stoplight with four different modes:

Input: Push Button

- Each time you press, you switch from mode: 0 → 1 → 2 → 3 → *repeat*



Output: LEDs

- Red = RD2
- Yellow = RD1
- Green = RD0

Mode:

- 0: Normal Mode. Cycle from green (5s) to yellow (1s) to red (6s) and repeat
- 1: Stop: Solid red (green and yellow off0
- 2: Flashing Red: Turn the red light on and off every 1 second. (Y/G off)
- 3: Flashing Yellow: Turn the yellow light on and off every 1 second. (R/G off)

Specify the interrupts used (one or more)

Interrupt #1 Interrupt name, set-up (PS, edge, etc)	Interrupt #2 (optional) Interrupt name, set-up (PS, edge, etc)
INT0 Rising Edge	Time0 1 second PS = 256

Specify the interrupt service routine(s). The main routine does nothing.

Interrupt #1	Interrupt #2 (optional)

`if(INT0IF) {`
`  Mode = (Mode + 1) % 4;`
`  INT0IF = 0;`
`  }`
	`if(TMR0IF) {`
	`  TMR0 = -39062;`
	
	`  if(Mode == 0) {`
	`  Time = (Time + 1) % 12;`
	`  if(Time < 5) PORTD = 1;`
	`  else if(Time < 6) PORTD = 2;`
	`  else PORTD = 4;`
	`  }`
	`  if(Mode == 1) PORTD = 4;`
	`  if(Mode == 2) {`
	`  RD1 = 0;`
	`  RD0 = 0;`
	`  RD2 = !RD2;`
	`  }`
	`  if(Mode == 3) {`
	`  RD2 = 0;`
	`  RD1 = !RD1;`
	`  RD0 = 0;`
	`  }`
	`  TMR0IF = 0;`
	`  }`