ECE 376 - Homework #8

Timer 0/1/2/3, INT.   Due Monday, November 1st

1) Write a C routine using Timer0 interrupts to measure time to 100ns. Using this routine, determine how long a the following operations in C take:

a) Double precision floating point sine

```
#include <math.h>
double A;

A = 1.23456;
A = sin(A);                // time to execute this instuction
```

b) LCD display routine

```
long int A;
A = 3141592653;
LCD_Out(A, 10, 7);   // time to execute this instruction
```

c) The time it takes you to press RB0 ten times (without using INT interrupts)

```
TRISB = 0xFF;
for(i=0; i<10; i++) {  // Start
   while(!RB0);
   while(RB0);
   }
```

2) Write a C routine using INT0 and Timer0 interrupts to measure time to 100ns. Using this routine, determine how long it takes to press RB0 ten times (ten INT0 interrupts)

3) Write a C routine using Timer0 / Timer1 / Tirme2 / Timer3 interrupts to play 4 notes at the same time when you press button RB0

Output Pin	RC0	RC1	RC2	RC3
Note	E2	F2	F#2	G2
Frequency (Hz)	82.41 Hz	87.31 Hz	92.50 Hz	98.00 Hz
Interrupt	Timer0	Timer1	Timer2	Timer3

3) Use INT interrupts to determine

Problem 4-8) Write a C program which uses at least two interrupts Timer0/1/3 interrupts. Some suggestions are

- Random number generator: Generate a 6-sided die (RB0), 8 sided die (RB1), or 12 sided die (RB2) based upon the time you press a button. Use statistics to see if the 6-sided die is really random.
- Hungry-Hungry Hippo: Press RB1 to start the game. Two players (RB0 and RB2) then press their buttons as fast as they can for 10 seconds (timed witth Timer0). After 10 seconds, the number of button presses by each player is displayed.
- Reflex Timer: Press RB1 to start. Between 3 and 10 seconds later, all of the lights on PORTA turn on. When they turn on, press RB0. The time delay from the lights turning on and pressing RB0 is displayed to 100ns.
- Music Box. Play a tune with your PIC processor
  - Timer0: Set the duration of each note
  - Timer1: Set the frequency of each note
- Other

4) Requirements: Explain what the inputs are / what the outputs are / and how they relate. Also explain how each timer interrupt is used in your embedded system.

5) C-Code and flow chart.

6) Test: Collect data in lab to verify that your interrupts are working properly.

7) Validation: Collect data in lab to verify you met your requirements

8) Demo (in person during Zoom office hours or in a video)