ECE 376 - Homework #6

A/D Converters, Chi-squared Test - Due Monday, March 3rd

A/D Conversion

1) Write a C program which

- Uses the A/D converter and a thermistor to measure the temperature of something, and
- Displays on the LCD the temperature in degrees C with a resolution of 0.01C.

Give

- Your C code, and
- The compiled size of your code

Thermistor Resistance vs. Temperature relationship:

$$R = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

Code:

```
Memory Summary:
    Program space        used  1BB2h (  7090) of 10000h bytes   ( 10.8%)
    Data space           used    43h (    67) of   F80h bytes   (  1.7%)
    EEPROM space         used     0h (     0) of   400h bytes   (  0.0%)
    ID Location space    used     0h (     0) of     8h nibbles (  0.0%)
    Configuration bits   used     0h (     0) of     7h words   (  0.0%)
```

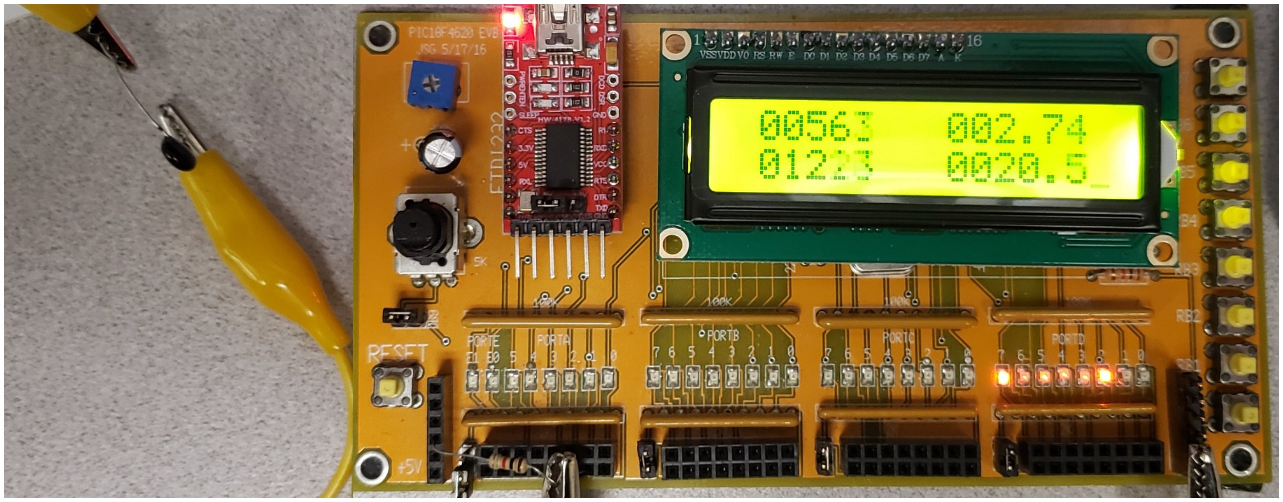
Main Loop (main routine)

```
// Initialize the A/D port
   TRISA = 0xFF;
   TRISE = 0x0F;
   ADCON2 = 0x85;
   ADCON1 = 0x07;
   ADCON0 = 0x01;

   while(1) {

      A2D = A2D_Read(1);
      VOLT = 0.488 * A2D;
      OHM = 1000.0 * (A2D / (1023.0 - A2D) );
      CELSIUS = 39300. / ( log( A2D / (1023. - A2D) ) + 13.1879 ) - 2730;

      LCD_Move(0,0);  LCD_Out(A2D, 5, 0);
      LCD_Move(0,8);  LCD_Out(VOLT, 5, 2);
      LCD_Move(1,0);  LCD_Out(OHM, 5, 0);
      LCD_Move(1,8);  LCD_Out(CELSIUS, 5, 1);

      Wait_ms(100);

      }
   }
```

2) With your temperature sensor, measure the temperature of four different things. Some suggestions are:

- The temperature of cold tap water
- The temperature of hot tap water
- The temperature of a refrigerator
- The temperature of a freezer
- Other

- Office: 20.6C
- Hot Tea: 45.6C
- Cold Water: 11.2C



Top Row: raw A/D reading, Volts
Second row: Resistance (Ohms) Temperature (C)

Chi-Squred Test

3) Determine experimentaly using a chi-squared test whether or not the following C code produces a fair 6-sided die:

```
while(1) {
   while(!RB0);
   while(RB0) DIE = (DIE + 1) % 6;
   DIE += 1;
   LCD_Move(1,0);  LCD_Out(DIE, 1, 0);
   SCI_Out(DIE, 1, 0);
   SCI_CRLF();
   }
```

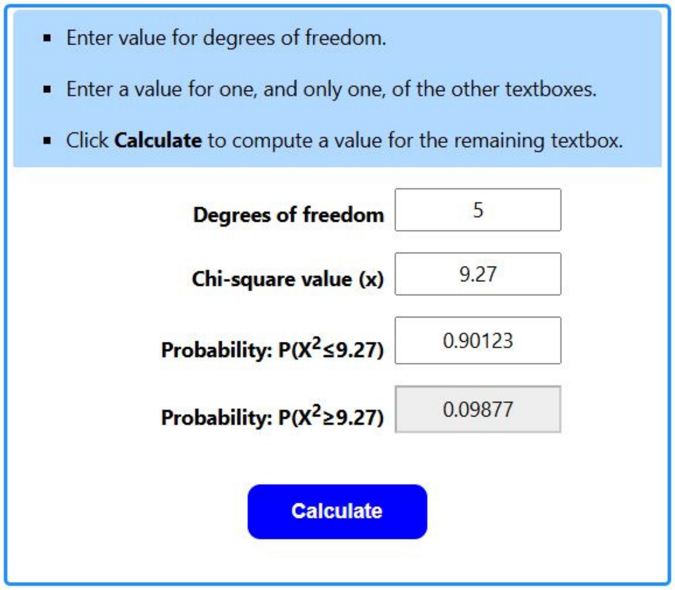
Collect data

Frequency of Numbers					
1	2	3	4	5	6
10	10	19	16	24	15

Compute the chi-squared score

Die Roll	p	np	N	chi-squared
1	1/6	15.67	10	2.05
2	1/6	15.67	10	2.05
3	1/6	15.67	19	0.71
4	1/6	15.67	16	0.01
5	1/6	15.67	24	4.43
6	1/6	15.67	15	0.03
			Total	9.27

From StatTrek, a chi-squared score of 9.27 with 5 degrees of freedom corresponds to a probability of 0.90123. **There is a 90.123% chance this is a loaded die**



- Enter value for degrees of freedom.
- Enter a value for one, and only one, of the other textboxes.
- Click **Calculate** to compute a value for the remaining textbox.

Degrees of freedom 5

Chi-square value (x) 9.27

Probability: P(X^2≤9.27) 0.90123

Probability: P(X^2≥9.27) 0.09877

4) Determine experimentaly using a chi-squared test whether or not the following C code produces a fair 6-sided die:

```
while(1) {
   while(!RB0);
   while(RB0) {
      DIE = (DIE + 1) % 13;
      }
   DIE = (DIE % 6) + 1;
   LCD_Move(1,0);   LCD_Out(DIE, 1, 0);
   SCI_Out(DIE, 1, 0);
   SCI_CRLF();
   }
```

Again, collect data

Frequency of Numbers					
1	2	3	4	5	6
31	21	29	22	31	27

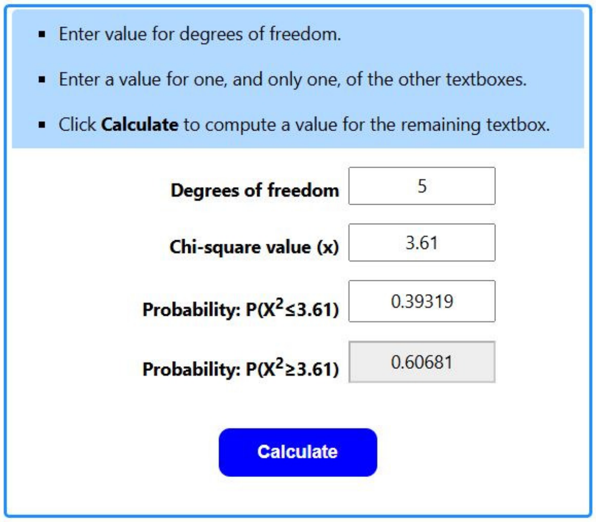
Compute the chi-squared score

Die Roll	p	np	N	chi-squared
1	1/6	26.83	31	0.65
2	1/6	26.83	21	1.27
3	1/6	26.83	29	0.18
4	1/6	26.83	22	0.87
5	1/6	26.83	31	0.65
6	1/6	26.83	27	0
			Total	3.61

From StatTrek, a chi-squred score of 3.61 corresponds to a probability of 39%

There is a 39% chance this is a loaded die

- It actually *is* a loaded die (0 comes up 3/13 of the time, other numbers come up 2/13 of the time)
- 161 rolls weren't enough to tell this die is loaded



- Enter value for degrees of freedom.
- Enter a value for one, and only one, of the other textboxes.
- Click **Calculate** to compute a value for the remaining textbox.

Degrees of freedom 5

Chi-square value (x) 3.61

Probability: P(X^2≤3.61) 0.39319

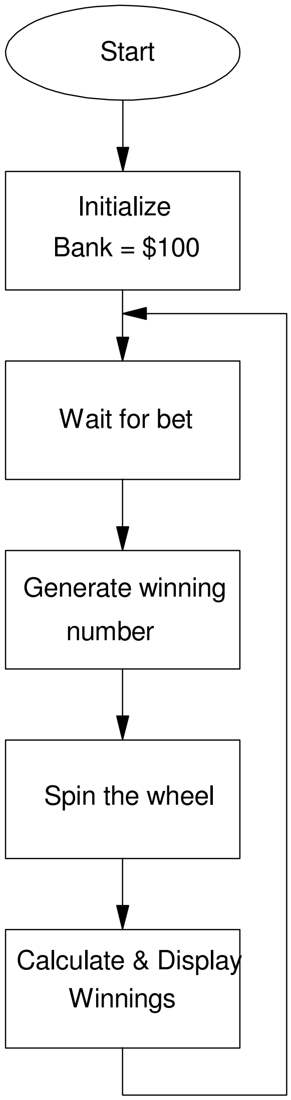
Probability: P(X^2≥3.61) 0.60681

Roulette!

Write a C program to play roulette:

- On reset, you start with $100 in your bank
- To start a game, press buttons RB0..RB7 to bet on the numbers 0..7
- When you release the button, a random number in the range of 0..8 is generated (nine numbers).
  - 0-7 corresponds to the winning number being 0..7
  - If you roll an eight, the winning number is the number you picked minus one (just missed)
- The PIC processor then counts 32 + N times (N = the random number), with 100ms between counts
- As it counts, the LCD display shows the ball position (0..7)
- When done counting, it checks if you won
  - If the number you pressed is equal to the final ball position, you win $8
  - If different, you lose $1

5) Give a flow chart for this program



6) Write the C code for this program

```
while(1) {
   while(PORTB == 0);
   while(PORTB != 0) {
      if(RB0) Bet = 0;
      if(RB1) Bet = 1;
      if(RB2) Bet = 2;
      if(RB3) Bet = 3;
      if(RB4) Bet = 4;
      if(RB5) Bet = 5;
      if(RB6) Bet = 6;
      if(RB7) Bet = 7;
      N = (N+1)%9;
      }
   if(N < 9) M = 32 - Die + N;
   else M = (32 - Die + Bet - 1);

   LCD_Move(1,0);  LCD_Out(Bet, 1, 0);

   for(i=M; i>0; i--) {
      Die = (Die + 1) % 8;
      LCD_Move(1,5);  LCD_Out(Die, 1, 0);
     Wait_ms(100);
      }
   if(Die == Bet) Bank += 8;
   else Bank -= 1;

   LCD_Move(1,10);  LCD_Out(Bank, 3, 0);

   SCI_Out(Bet, 1, 0);
   SCI_Out(Die, 1, 0);
   SCI_Out(Bank, 3, 0);
   SCI_CRLF();
   }


   }
```

7) Collect data to verify your program is working correctly

Try some test code:

Change the condition to if(N > 9)

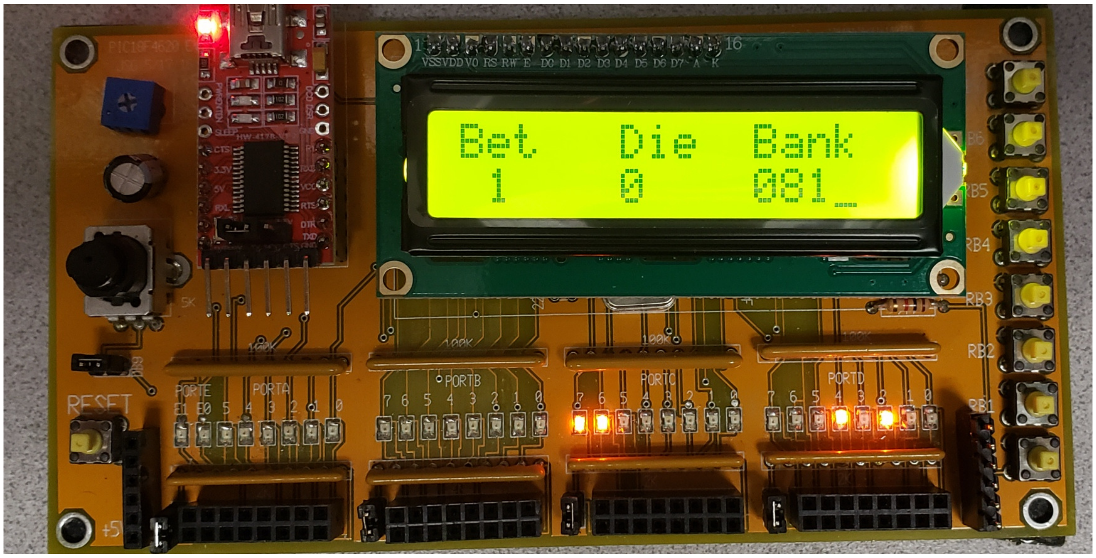
- Bet on 1, winning number = 0
- Bet on 2, winning number = 1
- Bet on 3, winning number = 2
- etc

Change back to if(N < 9)

- Bet on 1, winning number = 1 (win $8)
- Bet on 2, winning number = 3 (lose $1)
- Bet on 3, winning number = 1
- Bet on 4, winning number = 5
- Bet on 5, winning number = 0
- Bet on 6, winning number = 3
- Bet on 7, winning number = 6

Seems to work

- Winning numbers look random
- Win $8 when correct
- Lose $1 when incorrect



Roulette Wheel after 46 games (I've lost $19 so far, but I was *that* close to winning)

8) Determine using a chi-squared test if this is a fair game (the probability of each number coming up is 1/8, and/or the probability of winning is 1/8

Comment out the Wait_ms(100) code to speed it up

```
Bet Die Bank
 1 6 099
 3 5 098
 5 2 097
 3 0 096
 0 0 104
 7 0 103
 5 2 102
 4 1 101
 6 5 100
 3 5 099
 2 2 107
 5 0 106
 2 6 105
 3 0 104
 1 6 103
 2 0 102
 4 1 101
 7 0 100
 5 3 099
 3 6 098
 2 1 097
 4 5 096
 0 1 095
 2 7 094
 4 4 102
 6 0 101
 4 7 100
 4 2 099
 2 0 098
 5 3 097
 7 4 096
 5 3 095
 4 0 094
 7 1 093
 3 1 092
 2 0 091
 1 3 090
 5 6 089
 3 4 088
 1 2 087
 7 6 086
 0 1 085
 3 5 084
 5 2 083
 3 2 082
 1 0 081
```

I'm losing overall - looks like a riggend game

Chi-Squared Test: There are several ways to define N bins. Use two bins since there's not a lot of data

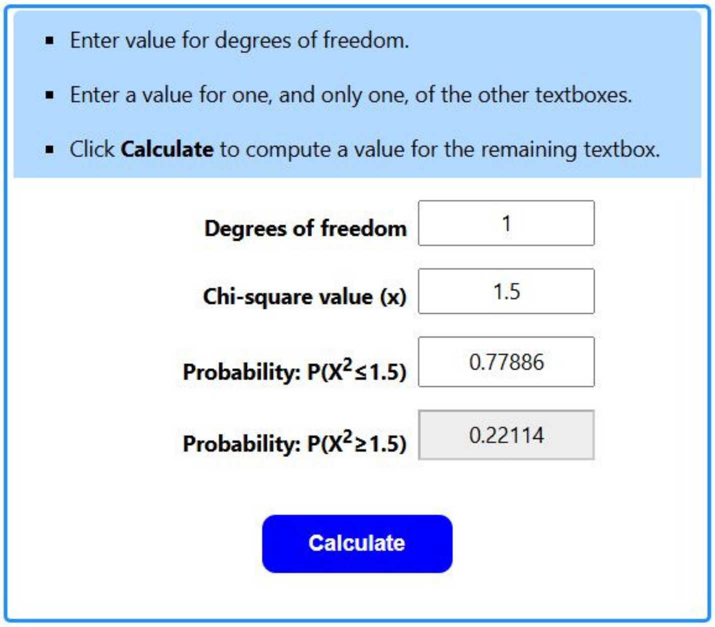
- Win (p = 1/8)
- Lose (p = 7/8)

With 3 wins in 46 games, the chi-squared score is 1.5

Bin	p	np	N	chi-squared
Win	1/8	5.75	3	1.32
Lose	7/8	40.25	43	0.19
			Total	1.5

From StatTrek, a chi-squared score of 1.5 corresponds to a probability of 0.7788

I'm 77.88% certain this is a rigged game



- Enter value for degrees of freedom.
- Enter a value for one, and only one, of the other textboxes.
- Click **Calculate** to compute a value for the remaining textbox.

Degrees of freedom	1
Chi-square value (x)	1.5
Probability: P($X^2 \leq 1.5$)	0.77886
Probability: P($X^2 \geq 1.5$)	0.22114