Homework 6; Your Name Here

Due Tuesday, November 17th at 10:00am

Worth 6 Course Points

There are 2 problems total, so each problem is worth 3 points.

Note: For this assignment, you will be turning in the following files.

1. Your .ipynb file with the filename "YourLastName_HW6.ipynb".
2. Your .pdf file that accompanies your .ipynb file.
3. YourLastName_3DayWSD.png for Problem 2.

In [2]:
```
from IPython.display import Image     #best way to import images so they render in a PDF
```

Problem 1 (Loops and Conditionals)

1. Write a script that finds the smallest odd integer that is divisible by 11 and whose square root is greater than 132. Use a loop in the script. The loop should start from 1 and stop when the number is found. It should also print the message, “The required number is: ” and print the number.
2. An array is given by `x = [-3.5, 5, -6.2, 11.1, 0, 8.1, -9, 0, 3, -1, 3, 2.5]`. Using conditional statements and loops (do not use Boolean slicing), write a program that creates two arrays from x. The first, call it P, should contain the positive elements of x. The second, call it N, should contain the negative elements of x. The elements of P and N should remain in the same order as x.
3. Repeat 1.2, but use Boolean slicing to find the positive numbers of x (call it P), and the negative numbers of x (call it N).
   - In your opinion, which method (Boolean slicing or using conditional statements and loops) was easier? You can answer in a comment line.
4. Write a script that finds a positive integer, n, such that the sum of all integers 1 + 2 + 3 + ... + n is a number between 100 and 1000 whose digits are identical. As the output, the program should display the integer n and the corresponding sum.
   - For example, for the integer n = 5, the corresponding sum would be 15, since 1 + 2 + 3 + 4 + 5 = 15.

Problem 2 (3-Day Weather Station Import)

In this problem, we will import from 3, 24-hours data files collected from a weather station that Jon set up in his office window a few years back. The files contain time, air pressure, temperature, and light intensity data. Each file contains 24 hours worth of data, acquired at a sampling frequency of 0.10 Hz (1 sample every 10 seconds). The purpose of this problem is to successfully import the data, append the 3 files to make 3-day long data arrays, and then make a nice multi-panel plot of the data.

1. Use `np.loadtxt()` to load the files "Weather_Data_13-04-XX_1306.lvm" where XX is 19, 20, or 21 depending on the data.
   For full credit, use a single loop to read in the data files.
   Note: Although they are ".lvm" files, you can import them just as you would import a ".txt" file.
   The first file records 24 hours of information starting at 13:06:38.5 (or 1:06 PM + 39.5 seconds) on April 19, 2013.

   - Unpack the data into four columns: time (seconds), atmospheric pressure (kPa), temperature (degrees Celsius) and light intensity (lux).
   - Please open the file and determine how many lines are required to remove the header.
2. Append the data together from these three imports, creating four 1D arrays for time, pressure, temperature, and light intensity. *Hint*: check the length of the final arrays to make sure you're on the right track. The final length should be 25920.
3. We would like to plot time in units of days, so convert your time array from seconds to days. Further, we will set time $t = 0$ at 00:00 on April 19, 2013. However, the data in the first file starts, not at 00:00 (midnight), but at around 13:06:38.5 (about 1:06 pm). If you open one of the data files, Line 11 of the header gives the start time of the data acquisition, 13:06:38.5 (see the image in the next block for more information about the format of the data files).
   In units of days, this means that the time array should start at $t = 0.54627$ days (AKA, a bit after noon). It should continue for the next three days, and end at $t = 3.54627$ days.

   - Manipulte the time array accordingly. After you are done, your time array should have an intial value of 0.54627, and end at about 3.54627 (where the units are days). As a check, you could print the first and last values of your time array and make sure your values match the ones given.
4. Make a nice subplot. Use any of the subplotting methods that we have discussed in Lectures 10 and 11 to create a three-row, single column, three panel plot of temperature (° C), pressure (kPa), and light intensity (lux) all versus time (days).

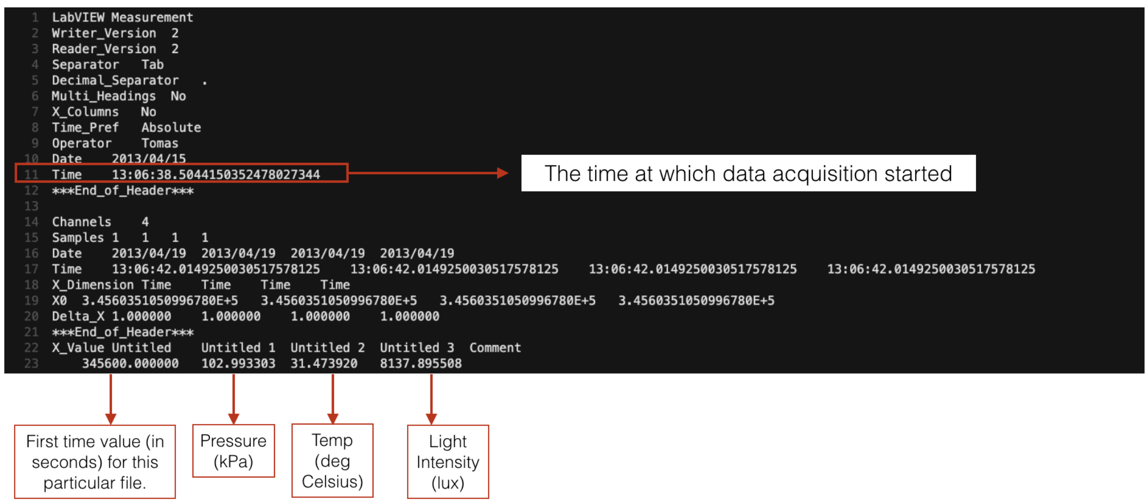
   - Use **generic codeblocks** to set up your subplots. If you are confused about this, see page 13 of Lecture 10. AKA, create a generic list of axes labels and data arrays that you can either (1) index or (2) loop through to create your subplot.
     - Set the figure size equal to (7.5,10).
     - The limts of your x-axis in all subplots should be 0 to 4 (days).
     - The limits of your y-axis should be 12.0 to 35.0 (degrees Celsius) for the temperature; 0 to 10000 (lux) for the light intensity, and 102.1 to 103.1 (kPa) for the pressure.
     - Set the linewidth of all plots equal to 0.50.
     - Denote the time that corresponds to the maximum and minimum temp/light intensity/pressure values with blue (min) and red (max) vertical lines. This step involves finding the indices that correspond to the minimum and maximum values of each data array.
   - Title the plot "3Day WSD Import: Your First Initial. Your Last Name, Today's Date"

- Make sure to label your axes with the appropriate units.
- Output the image as a .png file with the name YourLastName_3DayWSD.png

In [17]:
```
#Image of one of the data files, Weather_Data_13-04-19_1306.lvm,
#The time at which data acqusition was started is outlined in the red bo
x.
#The first column then gives the time (in seconds) for that particular d
ata file.
#The second column is the pressure (kPa)
#The third column is the temperature (deg. Celsius)
#The fourth column is light intensity (lux)

Image("DataFileAnnotate.png", width=950, height=950)
```

Out[17]:

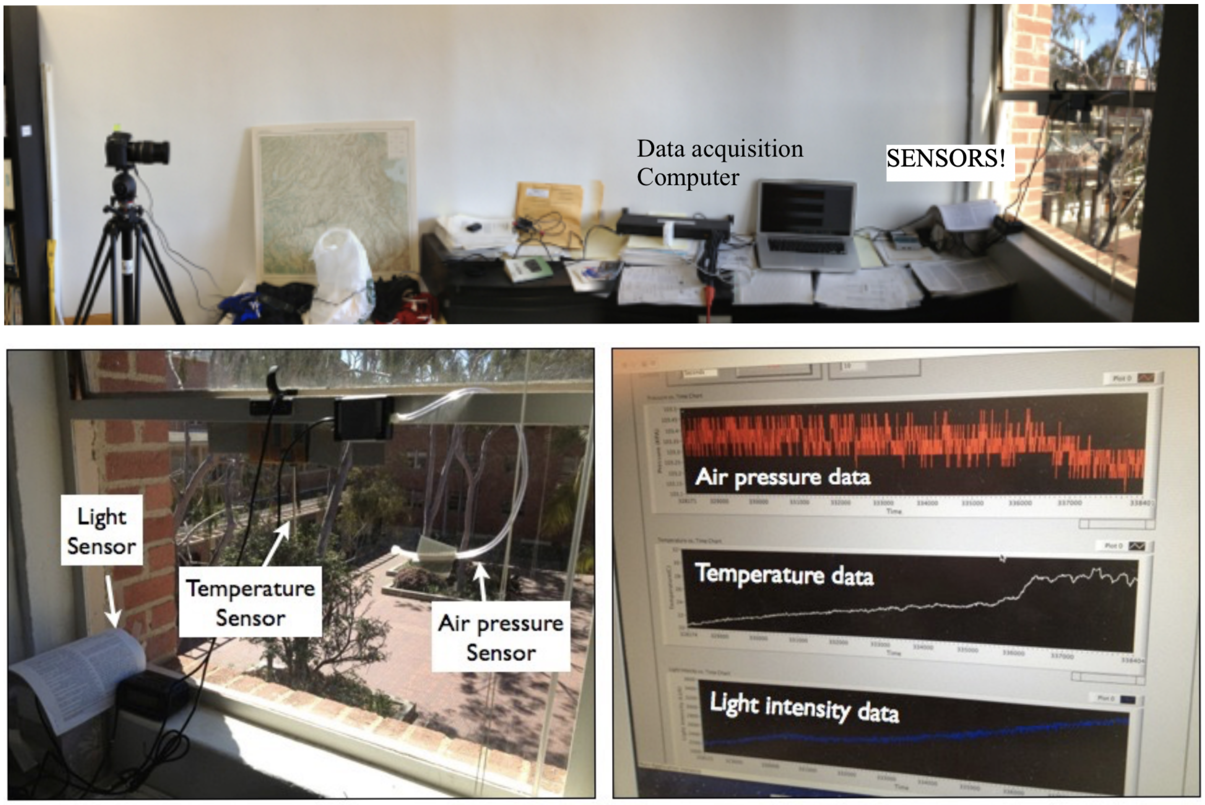


```
LabVIEW Measurement
Writer_Version	2
Reader_Version	2
Separator	Tab
Decimal_Separator	.
Multi_Headings	No
X_Columns	No
Time_Pref	Absolute
Operator	Tomas
Date	2013/04/15
Time	13:06:38.5044150352478027344
***End_of_Header***

Channels	4
Samples	1	1	1	1
Date	2013/04/19	2013/04/19	2013/04/19	2013/04/19
Time	13:06:42.0149250030517578125	13:06:42.0149250030517578125	13:06:42.0149250030517578125	13:06:42.0149250030517578125
X_Dimension	Time	Time	Time	Time
X0	3.4560351050996780E+5	3.4560351050996780E+5	3.4560351050996780E+5	3.4560351050996780E+5
Delta_X	1.000000	1.000000	1.000000	1.000000
***End_of_Header***
X_Value	Untitled	Untitled 1	Untitled 2	Untitled 3	Comment
	345600.000000	102.993303	31.473920	8137.895508
```

In [18]:
```
#Photograph of the Weather Station set up in Jon's office before he moved to his garage.
# In the bottom left you can see sensors just outside Jon's window.
# In the bottom right you can see real time data acquisition.
Image("WSD.png", width=700, height=700)
```

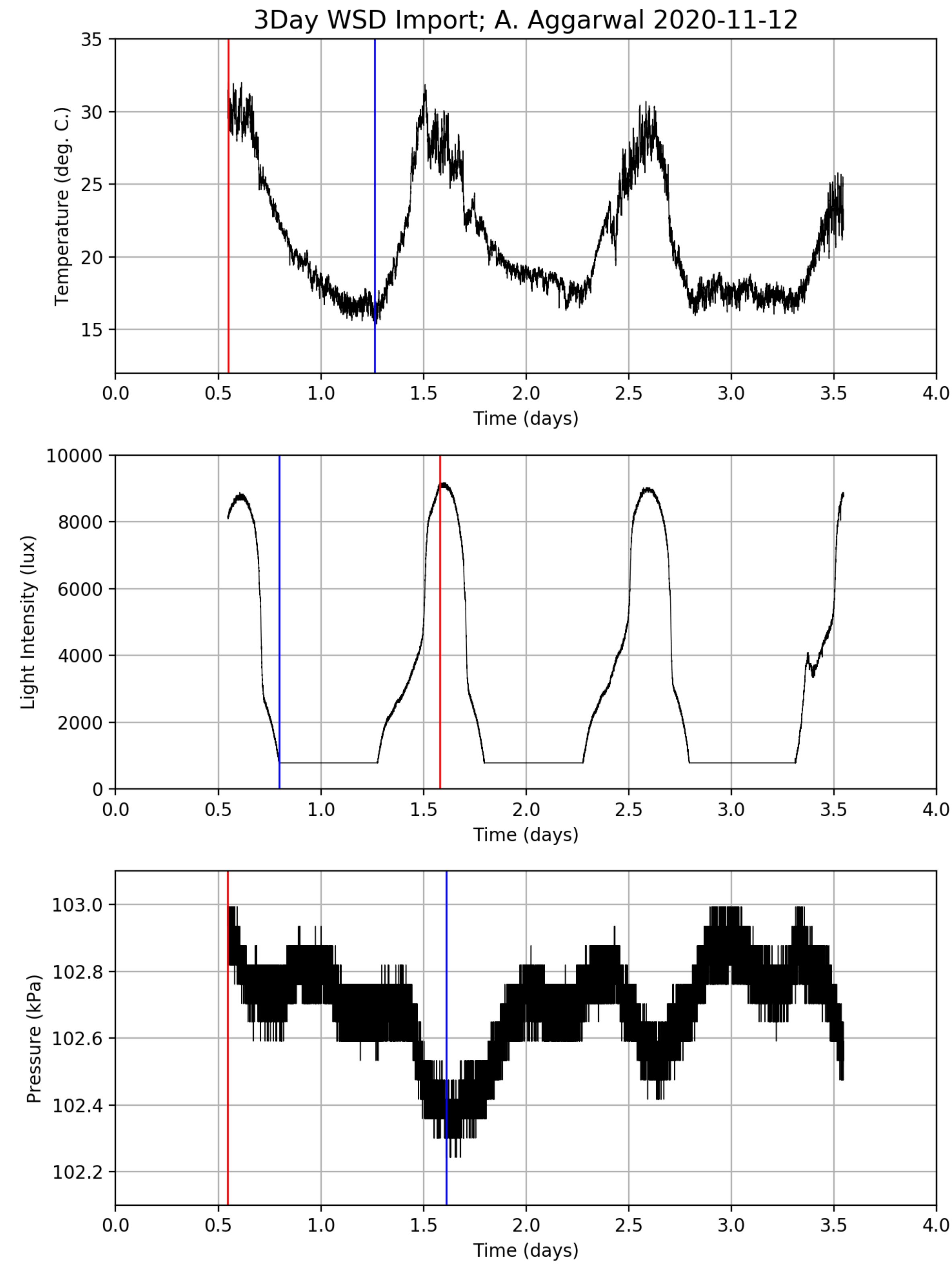
Out[18]:



In [19]:

```
#Example of the figure required for Problem 2
Image("Aggarwal_3DayWSD.png", width=700, height=700)
```

Out[19]:



We want your feedback!

For this week, we are conducting a short survey as a Mid-Quarter survey check in. Please click here (https://forms.gle/44D54P5PMhzZj6NE9) to fill out the survey.