CHEG 4139 - Temperature Control Prelab assignment

We will be using this website:
https://apmonitor.com/pdc/index.php/Main/ArduinoTemperatureControl

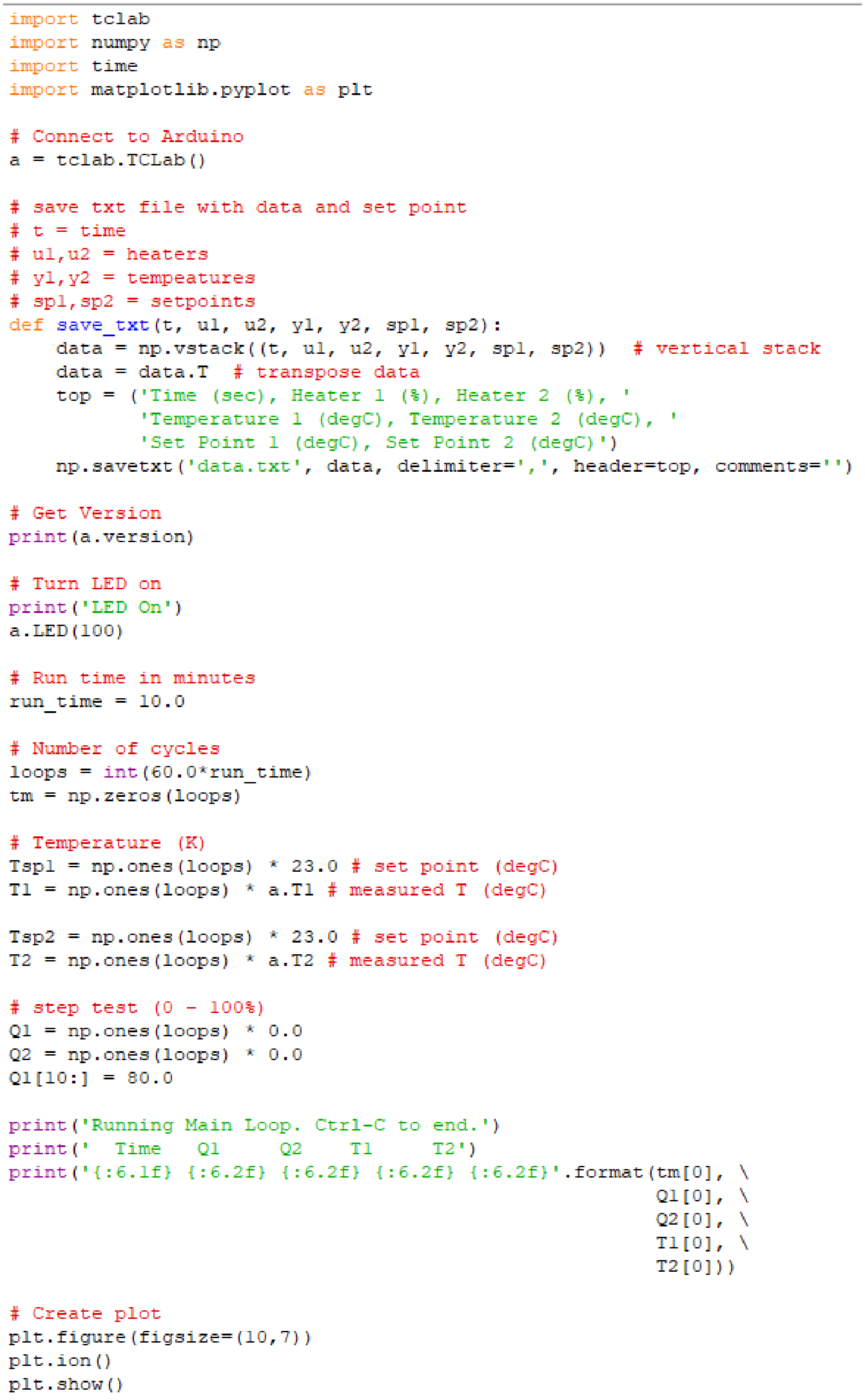
1) In order to prepare for your labs, it is important to familiarize yourself with your lab space. With this lab, you won't have a physical lab space, you'll have your code and Arduino board. When you are given code, it is always important to read through it and understand what is happening with it. Here is the Python code test_Heat that you will be using in this experiment. Use this to help answer the following questions.
    a. What function are the first 4 lines using?

    b. When you run this code, will it make a graph? If so, what does it display?

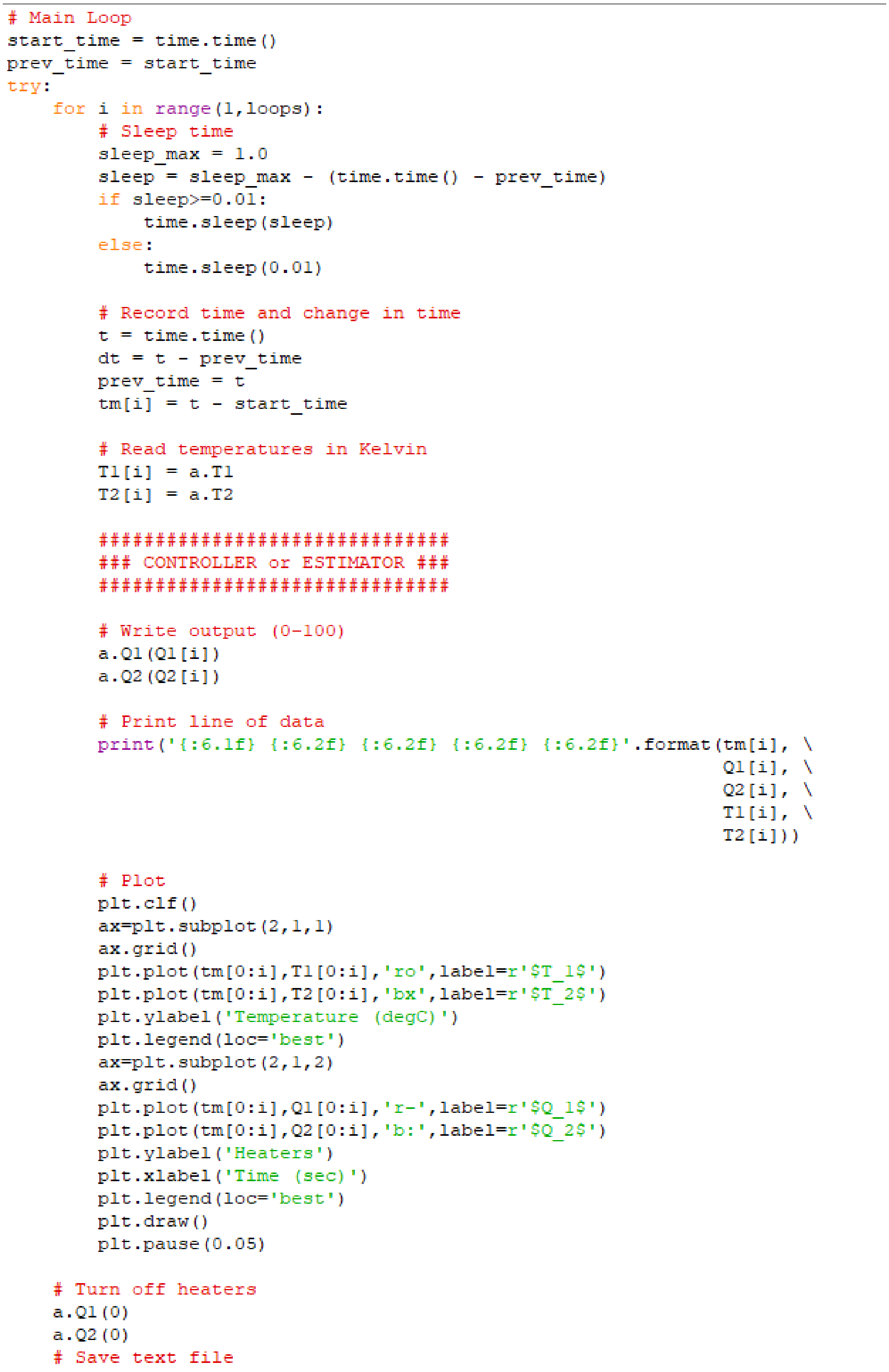
    c. How long will this code run for?

    d. Once this code starts, is there a way to stop it mid run?

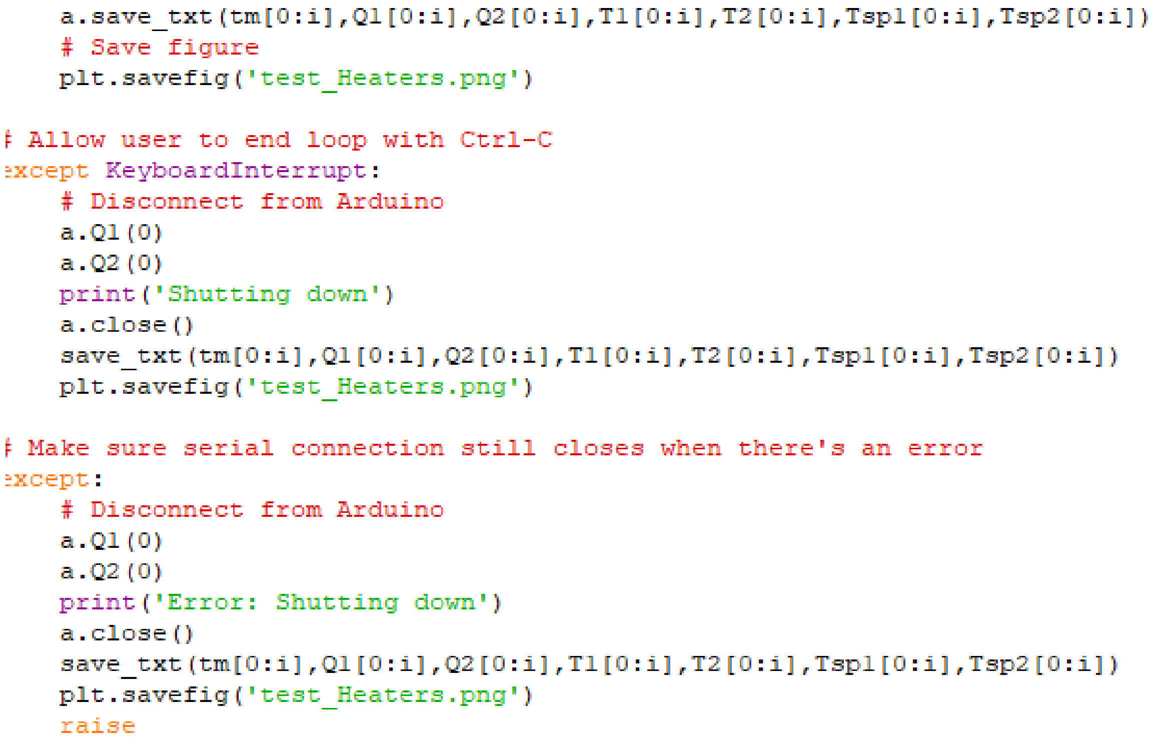
    e. What does this code do?



```python
import tclab
import numpy as np
import time
import matplotlib.pyplot as plt

# Connect to Arduino
a = tclab.TCLab()

# save txt file with data and set point
# t = time
# u1,u2 = heaters
# y1,y2 = tempeatures
# sp1,sp2 = setpoints
def save_txt(t, u1, u2, y1, y2, sp1, sp2):
    data = np.vstack((t, u1, u2, y1, y2, sp1, sp2))  # vertical stack
    data = data.T  # transpose data
    top = ('Time (sec), Heater 1 (%), Heater 2 (%), '
           'Temperature 1 (degC), Temperature 2 (degC), '
           'Set Point 1 (degC), Set Point 2 (degC)')
    np.savetxt('data.txt', data, delimiter=',', header=top, comments='')

# Get Version
print(a.version)

# Turn LED on
print('LED On')
a.LED(100)

# Run time in minutes
run_time = 10.0

# Number of cycles
loops = int(60.0*run_time)
tm = np.zeros(loops)

# Temperature (K)
Tsp1 = np.ones(loops) * 23.0 # set point (degC)
T1 = np.ones(loops) * a.T1 # measured T (degC)

Tsp2 = np.ones(loops) * 23.0 # set point (degC)
T2 = np.ones(loops) * a.T2 # measured T (degC)

# step test (0 - 100%)
Q1 = np.ones(loops) * 0.0
Q2 = np.ones(loops) * 0.0
Q1[10:] = 80.0

print('Running Main Loop. Ctrl-C to end.')
print('  Time   Q1     Q2    T1     T2')
print('{:6.1f} {:6.2f} {:6.2f} {:6.2f} {:6.2f}'.format(tm[0], \
                                                       Q1[0], \
                                                       Q2[0], \
                                                       T1[0], \
                                                       T2[0]))

# Create plot
plt.figure(figsize=(10,7))
plt.ion()
plt.show()
```



```python
# Main Loop
start_time = time.time()
prev_time = start_time
try:
    for i in range(1,loops):
        # Sleep time
        sleep_max = 1.0
        sleep = sleep_max - (time.time() - prev_time)
        if sleep>=0.01:
            time.sleep(sleep)
        else:
            time.sleep(0.01)

        # Record time and change in time
        t = time.time()
        dt = t - prev_time
        prev_time = t
        tm[i] = t - start_time

        # Read temperatures in Kelvin
        T1[i] = a.T1
        T2[i] = a.T2

        ###############################
        ### CONTROLLER or ESTIMATOR ###
        ###############################

        # Write output (0-100)
        a.Q1(Q1[i])
        a.Q2(Q2[i])

        # Print line of data
        print('{:6.1f} {:6.2f} {:6.2f} {:6.2f} {:6.2f}'.format(tm[i], \
                                                               Q1[i], \
                                                               Q2[i], \
                                                               T1[i], \
                                                               T2[i]))

        # Plot
        plt.clf()
        ax=plt.subplot(2,1,1)
        ax.grid()
        plt.plot(tm[0:i],T1[0:i],'ro',label=r'$T_1$')
        plt.plot(tm[0:i],T2[0:i],'bx',label=r'$T_2$')
        plt.ylabel('Temperature (degC)')
        plt.legend(loc='best')
        ax=plt.subplot(2,1,2)
        ax.grid()
        plt.plot(tm[0:i],Q1[0:i],'r-',label=r'$Q_1$')
        plt.plot(tm[0:i],Q2[0:i],'b:',label=r'$Q_2$')
        plt.ylabel('Heaters')
        plt.xlabel('Time (sec)')
        plt.legend(loc='best')
        plt.draw()
        plt.pause(0.05)

    # Turn off heaters
    a.Q1(0)
    a.Q2(0)
    # Save text file
```



```
    a.save_txt(tm[0:i],Q1[0:i],Q2[0:i],T1[0:i],T2[0:i],Tsp1[0:i],Tsp2[0:i])
    # Save figure
    plt.savefig('test_Heaters.png')

# Allow user to end loop with Ctrl-C
except KeyboardInterrupt:
    # Disconnect from Arduino
    a.Q1(0)
    a.Q2(0)
    print('Shutting down')
    a.close()
    save_txt(tm[0:i],Q1[0:i],Q2[0:i],T1[0:i],T2[0:i],Tsp1[0:i],Tsp2[0:i])
    plt.savefig('test_Heaters.png')

# Make sure serial connection still closes when there's an error
except:
    # Disconnect from Arduino
    a.Q1(0)
    a.Q2(0)
    print('Error: Shutting down')
    a.close()
    save_txt(tm[0:i],Q1[0:i],Q2[0:i],T1[0:i],T2[0:i],Tsp1[0:i],Tsp2[0:i])
    plt.savefig('test_Heaters.png')
    raise
```

2) This lab will also involve process control. Using the APmonitor website as a resource (https://apmonitor.com/pdc/index.php/Main/ProportionalIntegralDerivative) please write out the PID formula, identify the P, the I, and the D sections, and in a sentence or two, describe what each section calculates.

3) What are the major learning objectives of this experiment? (Hint: Gaining experience with large scale equipment is not a learning objective)

4) This experiment does not take place in a traditional laboratory. Despite that, what might be potential hazards that you could encounter?