SET UP INSTRUCTIONS:

1. Power your BeagleBone Black (BBB) by plugging to your computer with provided usb cable.
2. To work with Linux embedded devices e.g. BBB, you will NOT install an IDE. There are two ways of development on BBB mentioned as follows. It's up to you which one to choose.
    a. Login to BBB as it runs an entire linux operating system. Write code, compile and run on BBB.
    b. Develop code and compile on a computer, then transfer compiled binary to BBB and run on it.
3. To login to BBB, you need to establish a connection from your computer to BBB through a serial console using **ssh** protocol. You can use different serial clients depending upon your operating system e.g.use Terminal for MAC, and Putty for Windows. Refer to the following link for details, https://www.howtogeek.com/311287/how-to-connect-to-an-ssh-server-from-windows-macos-or-linux/
4. Initiate an ssh connection to your beaglebone from your computer using the default IP address assigned to BBB's usb port,
    a. You will use a different ssh client depending upon your operating system
    b. IP address of BBB is either 192.168.7.2 or 192.168.6.2 depending upon your operating system.
    c. Username: debian, Password: temppwd
    d. To login to BBB, use command,
        i. > ssh debian@192.168.X.2 (X is either 6 or 7)
        ii. Enter the password and you are logged in
5. Explore the filesystem on BBB and get accustomed to it using frequently used linux commands such as "ls" and "cd". Some useful commands here, https://www.tecmint.com/useful-linux-commands-for-newbies/
    a. Learn the following about your BBB,
        i. Kernel version (command: uname -a)
        ii. Network configuration (command: ifconfig)
        iii. CPU information (path: /proc/cpuinfo)
        iv. GPIO (path: /sys/class/gpio)
        v. Ethernet Hardware Clock (path: /sys/class/ptp/ptp0)
        vi. File transfer (psftp, scp) - (Transfer files to and from BBB)

LAB 3:

1. Write a code in Python and name it “firstname_lastname_PWM”
    a. This code should generate a Pulse Width Modulated (PWM) signal on a GPIO pin (P8_13)
    b. You should be able to change the period of PWM
    c. You should be able to change duty cycle of PWM

    HINT*: You can make use of already provided python libraries in BBB*
2. Keep this PWM python code running in one window so PWM signal is available on P8_13
3. Use a male-male jumper wire to feed this PWM signal from P8_13 to P8_9

    HINT: We have provided a BBB pinout for clarity on pin connection.
4. Write another code in Python and name it “firstname_lastname_trigger”
    a. Configure P8_9 as an input pin
    b. Configure to interrupt when a PWM rising edge is detected at this pin
    c. Upon detecting a rising edge at this pin, timestamp and save it in a local file, and wait for the next rising edge
    d. Timestamp using the clock that provides sub-millisecond resolution
    e. Keep on repeating step c and running the experiment for a duration of 20 minutes i.e. timestamp every rising edge of PWM for 20 minutes
    f. Calculate Jitter using the equation below,
        i. Jitter = Actual period of one PWM cycle - Measured period of one PWM cycle

    HINT: *Actual period is 1second for 1Hz PWM, 10 millisec for 100 Hz PWM signal; whereas, Measured period comes from subtracting consecutive timestamps in Step c.*

        ii. Calculate mean and standard deviation of jitter plot
    g. Plot Jitter Distribution from step f. i.
        i. Use any plotting tool you are comfortable with such as python, Matlab or Excel.

    HINT: *Most software have predefined libraries that provide plotting functions*
5. Run step 3 by varying the frequency of the PWM signal to: 1Hz, 100 Hz, 10,000Hz

REPORT FORMAT:

Format your reports using the following instructions,

1. Provide three jitter distributions plots for 1Hz, 100Hz, and 10,000Hz PWM signals
2. Report mean and standard deviation of each distribution
3. Explain your results quantitatively,
    a. What factors contribute to this jitter.
    b. What are different ways to minimize this jitter

SUBMISSION:

Submit your labs on Gradescope.

1. You will submit **three** files,
    a. Python code for generating PWM, “firstname_lastname_PWM”
    b. Python code for triggering, “firstname_lastname_trigger”
    c. Report with all the graphs and results
2. Gradescope will allow you to submit multiple files