This project consists of 3 parts. Each one focuses on several main utilities of coding. **For This project you must upload everything that will be necessary for your code to run.** Code should have meaning full comments on them. A large focus of the project will be googling/ searching how to do things we have not covered in class. Each step is pretty detailed out and should guide you towards what you should be looking for. The nature of coding is that you will oftentimes not know something but that the solution exists on the internet already. In the future you will regularly come across things you do not know and it is good to build up a good level of critical thinking that will lead you to a solution. Additionally, it is not unreasonable to work problems out with others when you are stuck. Engineering is a very collaborative study and as such **the project can be done in groups of 3.**

Project 1: Lets Play a Game (30 %)

The goal of this project is to play with loops and conditionals. For this project you will be making a game of Tic-Tac-Toe.

Task 1: User input tic tac toe

1. Requirements:
   1. Board must be constructed out of a 3x3 matrix
   2. Must utilize a while loop
   3. Must utilize If and else conditionals
   4. Selected board position must be a user input from players 1 and players 2
   5. Display the board after each move is made
   6. A player can't select a board position that has already been taken. If they do, prompt them with a select again comment and let them select again.
   7. Game should stop when a win condition is met
   8. Game should be able to go to a tie
   9. Display which player won or if a tie was met

Task 2: tic tac toe simulation over 5000 iterations

* Requirements:
  1. Take the code from above and make slight modifications to it
  2. remove any input prompts
  3. Use a random integer generator to select the moves for each player
  4. Toss the code into a for loop that runs the game N times (for this instance N =5000 but I should be able to change it to any number)
  5. Modify the code as needed to properly run
  6. Keep log of each win, loss, and tie
  7. Generate a histogram of win, loss, and tied games

Project 1: Playing with real data (30 %)

**The goal of this project is to play with some real data. I have provided you with some real data from my research.** The data collected is from an instrument called a hot wire anemometer. This device heats up a small wire and maintains a constant temperature. When wind crosses the wire,it cools down due to forced convection. To compensate for the cooling, the sensor sends more energy to the wire to get it back up to temperature. The amount of energy added is then recorded with a data acquisition unit in terms of voltage needed to maintain temperature. A higher voltage corresponds to a higher wind speed. The data set given is a time trace from an experiment ran inside a wind tunnel. The wind inside the wind tunnel is dynamically controlled as a sum of sines but is unknown. We seek to determine that input. **Largely I will be walking you through step by step on how to solve this problem. Giving you intuition and logic along the way that should generally match how you would solve similar problems in the future.**

Task 1: Finding an input with only the output

1. Take the data from the excel sheet **“Dynamic\_wind.xlsx” and save the data column and the t column to variables voltage and t**
2. Since the data is currently in Voltage we need a way to relate the voltage to velocity ( wind speed). Previously a calibration was ran on the sensor and was found that the following average voltages cal\_volts=[1.1334761517302536, 1.2168838402316453, 1.2574484652661773, 1.28406653483074, 1.2945098977285117, 1.306963086494241, 1.3147604417015477, 1.3228523686035325, 1.3311519051962617, 1.3343771944501273]; (volts)Correspond to the average wind speeds cal\_speed=[0,.31,.53,.68,.73,.84,.90,.96,1.02,1.10];(m/s) **Use these two arrays and the polyfit() function to find a first order best fit line.** This will be our translation for our collected data to wind speed
3. **Use the output form polyfit(), and voltage in the function** **polyval** save this new array as V
4. Now we need to take a look at the time array we inputted.Looking at the data it appears to all be the same time for each step. In actuality it's not, this time structure is something called epoch time. Let's correct it to a readable data set. **Take the last value of t and subtract the first value from t save as T**. This will get you the time the whole experiment was ran in.
5. Now let's get an appropriate time stamp for each data point. Create a **linearly spaced array starting at 0 and going to T that has a length equal to the original time array call it TIME**.
6. **Plot Time vs V in figure(1)**
7. Now that we have a usable output signal we want to play with the data to see if we can find the input signal. Since we know the input is somehow governed by a sum of sines we can use a neat mathematical function called fourier transform (fft). This tells you dominant frequencies in a data set. What this means is if there is oscillation in your system it will tell you at what frequencies they occur. Now looking at the data there is not a lot we can see by eye for frequency so lets run an fft analysis on it. **Check out link :** [**https://stackoverflow.com/questions/10758315/understanding-matlab-fft-example**](https://stackoverflow.com/questions/10758315/understanding-matlab-fft-example)
8. Using the link above calculate the fft of our V data, take their example and eddit in our data.
   1. Sampling frequency can be calculated as **Fs = 1/(TIME(2)-TIME(1));**
   2. Sampling can be calculated as **Tf = 1/Fs;**
   3. Looking at the example L must be in milliseconds so let's find L to now be  **L = T\*1000;**
   4. Time vector is now **tf = (0:L-1)\*Tf;**
   5. This should be enough for you to finish off calculating the fft on your own.
9. Axis here should be frequency and power; the larger the power the more dominant the frequency. So if we see a spike at a given frequency we are doing well. However plotting the data as they do seems to not give us much information. In this instance I would then try plotting on a loglog scale. Remember log scale mage big changes small and small changes big and is like counting in powers of 10. Eg [.001 ,.01, .1, 1, 10, 100]. **Plot now on a loglog scale**
10. What do we see? Quite an odd looking figure. But this figure has meaning. For example the steep change in slope at the high frequency range indicates the wind data we collected was turbulent. We also now see 3 obvious peaks around the 10^-1 range; these peaks must mean our wind speed has periodic tendencies at these frequencies and would likely correspond to our signal. **Zoom in on the 10^-1 range and Identify the 3 peak’s coordinates toss 3 text labels pointing out those peaks.**
11. Now that we have those peak values we can interpret them. Each peak is a frequency. What function is based on frequency? Sine! Use those 3 frequencies to generate a sum of sines. **sig=sin(f1.\*TIME)+sin(f2.\*TIME)+sin(f3.\*TIME); where f1 = 1/peak1 ,f2=1/peak 2, and f3=1/peak3**
12. **Now make a figure that is a plot of the signal across 20 seconds**
13. This figure is a plot of a pretty close representation for the signal sent to the fans that generated the wind we collected.

Project 3: Avengers Assemble (40 %)

The goal of this project is to exercise your ability to search for existing functions and make your own. Taking the script of The Avengers movie you will create a code that when given a Superhero name it shows a distribution of their N most used words. For all functions given, rename them with your Initials attached to the end. Eg. get\_words\_AL

Task 1: create a function that reads a .txt file and makes a cell array of every word

1. Requirements:
   1. I’ll get you started. Download function get\_words and fill in the blank
   2. Input must be a string
   3. Output will be a single array of all the words in the text
   4. You must open and close each text file
   5. Googling “matlab function takes words from a text file” leads the link: <https://www.mathworks.com/help/matlab/ref/textscan.html> use this function to complete Task 1

Task 2: create a function that takes your word array and removes all delimiters and makes all words lowercase

* Requirements:
  1. Take the function remove\_delim and fill in the blanks
  2. Google around a bit and find a function that removes a partial string from the whole string(delimiters) eg. Hello! to Hello
  3. Iterate the step above for each string in your cell array.
  4. Do steps 2 and 3 for [".",",","?","!"]
  5. Lastly, make every string lowercase

Task 3: create a function that removes the top most commonly used word in the english language and finds the length of the resulting array

* Requirements:
  1. Take the function clean\_words and fill in the blanks
  2. Google “remove specific entries in a cell array matlab” and play with the search results
  3. Use <https://www.espressoenglish.net/the-100-most-common-words-in-english/> to find a list of the top most common words
  4. Use step 2 and 3 to remove the top 20 words from the array
  5. Set the filtered data and the Length of the resulting array

Task 4: create a function that aggregates all the repeated words and finds the count of each category.

* Requirements:
  1. Take the function categorize\_words and fill in the blanks
  2. Google around a bit to find some functions that can be used to categorize and count strings in a cell array
  3. Have your outputs be the categories which will later be used for the x axis labels of a bar graph and the count for each word eg. dataclean={dog,hello, hello,dog,car,dog} → barx={hello, car, dog} and count =[2,1,3]

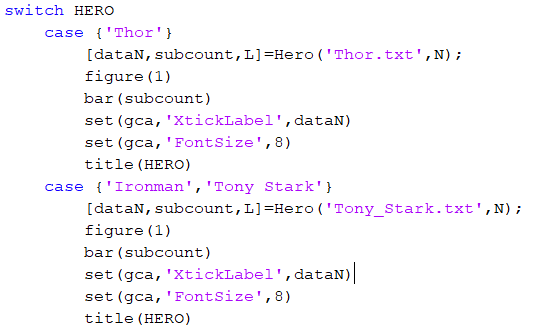
Task 5: I have given you a function that uses the output from task 4 and finds the top N used words and finds each word’s count. Give it a proper description for when calling help.

* Requirements:
  1. Take my function(Top\_N) that is fully completed and add a useful descriptor to it

Task 6: Generate a function that uses all 5 previous functions in order

* Requirements:
  1. Create a function that uses all 5 previous functions in order
  2. Inputs should be the name of a .txt file and the N number of top used words.
  3. Outputs should be a cell array of the top N number words, each of those words counts (outputs form Top\_N) and Length from clean\_words

Task 7: Plotting top words used in the movie

* Requirements:
  1. Create a script that uses a user input to specify the inputs into your function from task 6.
  2. Inputs should be a string and a number corresponding to the hero's name and how many words you want to see. To test use Ironman and 10
  3. Create a switch case structure that takes in the Heroes name. And then uses your function from task 6 to plot a bar graph where the y axis is the count and the x axis is the N most used words by that character.
  4. Each case should be able to except the super hero name or their secret Identity
  5. Eg. 
  6. Lastly, when prompted with the hero name and given the string ‘Avenger Assemble’ a 2x3 sub plot with all 6 heroes should be created with appropriate titles.