The following parts make up this project. Each part, although similar and possibly using the same code, must be written as separate functions. Also, although there should be 1 main function for each part, it is possible to have many functions that are included and are used by the main function.

NOTE: Do each part, one at a time, in order. Make sure to comment and include H-1 and description for the “help” command

I.) A function must be written to accept the following inputs and outputs:
   a. a matrix (A), input, with dimensions m by n, where m>=n and n always equal to 2 for this project. (So two intersecting lines is the minimum number of lines to get at least 1 extreme point.)
   b. a vector (b), input, with dimensions m by 1, where is m is the number of rows of A (another way to think of m is the number of lines in this problem)
   c. a matrix (X), output, that holds all the extreme pts found.
   d. produce a plot, output, of the polygon formed by the extreme points

II.) A function must be written to accept the same inputs and outputs as in part I.) plus the following inputs.
   a. a vector (c) (where transpose(c)*x is the cost that you are trying to minimize or maximize). Where x=[x1;x2]
b. a string variable (MaxMin) where the only choices as the string input are ‘max’ or ‘min’.

c. a variable (p_star), output, that holds the min or max value (depending on the string MaxMin) of p=transpose(c)*x. The desired optimized variable (for example Revenue)

d. a variable (x_star), output, that holds the extreme pt that gave p_star

III.) EXTRA PART: This is your chance to shine and make something cool:

a. come up with an original idea to add to this program. It can be any of the following ideas but will be graded according to which one you have chose. For example i.) will be graded the highest but it will take a lot of work.
   i. Ideally, something GRAPHICAL that has to do with the optimization that you have learned or in relation to the polygon that is produced
   ii. Something GRAPHICAL that doesn’t have anything to do with optimization
   iii. Something that is NOT GRAPHICAL that has something to do with optimization

NOTE: I recommend that you start and finish parts I.) and II.) before getting too involved with part III.).

NOTE: Start this project as soon as possible.

**SIMPLEX METHOD:**

Assume that you have a current point (x), that could be the starting pt of [0;0] and the variables (A),(b), (A_I),(b_I).

Initially (A_I) and (b_I) can be found by comparing A*[0;0] to (b) and producing the first (I=[i j]) vector

1) decide which value of (I) should be removed (i or j?)
2) produce DeltaX from the equation A_I * DeltaX = [0;-1] if you are removing i from I (or [-1;0] if you are removing the j from I)
3) make the vectors (A*DeltaX) and (b-A*x) (x being the current point, which could be the starting pt)
4) produce Alpha = (b-A*x)/A*DeltaX for elemental values of DeltaX positive ONLY
5) choose the smallest value of Alpha and replace the removed (i or j) variable from (I) with the row variable that corresponds with the smallest Alpha
6) Now make a new (I) vector
7) Produce a new (A_I) and (b_I)
8) Calculation new point (x) from A_I * x = b_I and store it in X as an extreme pt

!!!!!!!!!!!!!!!!!REPEAT!!!!!!!!!!!!!!!!!!!!!!!