1. You have to choose the quantities of 3 different products to produce. The price you can get on the market for each product depends not only on the quantity of that product produced, but also on the amounts of the other products produced since the products are (imperfect) substitutes. In particular

\[ p_1 = 10 - q_1 - 0.5q_2, \]
\[ p_2 = 20 - q_2 - 0.5q_1 - q_3, \]
\[ p_3 = 30 - q_3 - 0.5q_2. \]

The total cost of production is

\[ C(q) = e^q_1 + e^{2q_2} + e^{3q_3}. \]

The profit is thus \( p't - C(q) \).

(a) Write out the objective function to minimize (minus profit). Verify that it is convex.

(b) Solve the problem numerically by writing matlab code to solve this problem using conjugate gradient with greedy steps. Turn in a print out of your code, and verify your answer is correct by checking the KKT conditions.

(c) Now suppose you have constraints \( q_2 \leq a \) and \( q_3 \leq b \) for some specified \( a \) and \( b \). Write a matlab function to minimize your objective subject to these constraints. You should be able to call your function from the command line as:

\[ [x,fval] = myminimize(a,b,x0,W) \]

where \( W \in \{[0,0],[0,1],[1,0],[1,1]\} \) is an indicator of your initial working set of active inequalities (1=on, 0 =off), \( x_0 \) is a feasible starting point, consistent with your working set, and output \( x \) is the solution and \( fval \) the optimal objective. Check the answer it gives you is correct for \( a = 2, b = 0.5 \) by verifying the KKT conditions.

2. A defender has 3 assets it wants to defend and they are labeled \( \{A,B,C\} \). The assets have values \( v_A, v_B, v_C \). The defender has 3 guards that it can assign to assets in any way she chooses, including all guards on 1 asset, or putting them on separate assets. Thus the following 10 strategies are possible:

\[ \{AAA, BBB, CCC, AAB, AAC, BBA, BBC, CCA, CCB, ABC\} \]

Where the letters indicate the assignment of guards. Since the guards are indistinguishable, only the number of guards assigned to each asset matter. Let \( x \) be a 10x1 column vector with \( x_i \) being the probability the defender chooses strategy \( i \). Since \( x \) contains probabilities of distinct choices, \( x \geq 0 \) and \( 1^T x = 1 \). Please use an indexing convention that is consistent with the ordering of strategies above.

An attacker chooses how to direct 2 operatives to the assets. Thus the defender picks from the following strategies:

\[ \{AA, BB, CC, AB, AC, BC\} \]

Let \( y \) be a 6x1 column vector with \( y_i \) being the probability the attacker chooses strategy \( i \). \( y \geq 0 \) and \( 1^T y = 1 \). Please use an indexing convention that is consistent with the ordering of strategies above.

The attacker gets the value of any asset for which it assigns more operatives than the defender assigns guards. If the number of operatives and guards is equal, the attacker has a 50/50 shot at taking that asset, so in expectation he gains half the value of the asset. If the attacker has fewer operatives on a target than there are guards, the attacker gets no value from that asset. The attacker’s total payoff is the sum of the gains from each target, while the defender’s loss is simply what the attacker gains. For example:
The attacker payoff can be expressed as 

$$y^T M x$$

where $M$ is a matrix with $M_{ij}$ being the payoff when the attacker plays strategy $i$ and the defender plays $j$. If we formulate the attacker’s problem of maximizing his minimum payoff we have:

$$\max \ s$$

s.t. $1_{6 \times 1}^T y = 1$

$$y^T M \geq s 1_{10 \times 1}^T$$

$$y \geq 0$$

Similarly, the defender’s problem of minimizing his maximum loss:

$$\min \ r$$

s.t. $1_{10 \times 1}^T x = 1$

$$M x \leq r 1_{6 \times 1}$$

$$x \geq 0$$

In class we will show that the latter problem is the dual of the former, and thus they produce the same value $r^* = s^*$.

(a) Write a Matlab script to specify $M$ numerically given a choice of $v_A, v_B, v_C$. It might help to use Matlab’s string functions to automate the checking of who wins the different assets. Define the string arrays:

S = {'AAA', 'BBB', 'CCC', 'AAB', 'AAC', 'BBA', 'BBC', 'CCA', 'CCB', 'ABC'};
T = {'AA', 'BB', 'CC', 'AB', 'AC', 'BC'};

To test whether the attacker wins a particular asset with a particular strategy pair $(i,j)$ you can use something like

```matlab
if count(T(i), 'A') > count(S(j), 'A')
    M(i,j) = M(i,j) + v_a;
elseif count(T(i), 'A') == count(S(j), 'A')
    M(i,j) = M(i,j) + v_a/2;
```

(b) Solve the game using one of the LPs above, in the case that all three targets have values of 1. You may use linprog in Matlab, but you will have to use care in defining the matrices that you pass to it in your function call. Provide the optimal value you get (known as the value of the game) and the strategy vectors.

(c) Re-use your code to solve the case when $v_A = 1, v_B = 2, v_C = 3$.

(d) Solve the case when $v_A = 1, v_B = 1, v_C = 50$. Does the attacker only attack target C?