```
%%graphical

%step 1 define
A = [1 2;3 1;4 3;0 1;1 0]
B = [40;30;60;0;0]
C = [20 10]

%step 2 generate points
x1 = 0:max(B)

X21 = (B(1)-A(1,1)*x1)/A(1,2)
x22 = (B(2)-A(2,1)*x1)/A(2,2)
x23 = (B(3)-A(3,1)*x1)/A(3,2)

x21 = max(0,x21)
x22 = max(0,x22)
x23 = max(0,x23)

%step 3 plot
figure
plot(x1,x21,x1,x22,x1,x23)
xlabel('x1')
ylabel('x2')
title('Graph')
legend('x+2y \leq 40', '3x+y \geq 30', '4x+3y \geq 60', 'Location', 'Best')
grid on
hold off

%step 4 intersection
sol = []
for i = 1:size(A,1)
   A1 = A(i,:)
   B1 = B(i,:)
   for j = i+1:size(A,1)
      A2 = A(j,:)
      B2 = B(j,:)
      A3 = [A1; A2]
      B3 = [B1; B2]
      X = A3\B3 %or inv(A3)*B3
      sol = [sol; X'];
   end
end

disp('Intersection Points:');
disp(sol);

%step 5

x = sol(:,1)
y = sol(:,2)
H1 = find(x+2*y-40>0)
sol(H1,:) = []

x = sol(:,1)
y = sol(:,2)
H2 = find(3*x+y-30<0)
sol(H2,:) = []
```

```
x = sol(:,1)
y = sol(:,2)
H3 = find(4*x+3*y-60<0)
sol(H3,:) = []

%step 6 find max
for i=1:size(sol,1)
   obj(i,:) = sum(sol(i,:).*C)
end

A = max(obj)
P = find(obj==A)
s = sol(P,:)


% Basic sol

A = [1 1 2 3; 0 1 2 1];
B = [12; 8];
C = [1 2 -3 4];
n = 4;
m = 2;

if n > m
   ncm = nchoosek(n, m);
   p = nchoosek(1:n, m);
   sol = [];
   obj = [];

   for i = 1:ncm
      y = zeros(1, n);
      A1 = A(:, p(i,:));
      if rank(A1) == m
         X = A1 \ B;
         if all(X >= 0)
            y(p(i,:)) = X';
            sol = [sol; y];
            obj = [obj; C * y'];
         end
      end
   end

   if ~isempty(sol)
      [maxx, idx] = max(obj);
      opsol = sol(idx, :);

      disp('Feasible Solutions:');
      disp(sol);

      disp('Optimal Solution:');
      disp(opsol);

      disp('Maximum Objective Function Value:');
      disp(maxx);

      % Display points where the value is max
```

```
        disp('Points corresponding to maximum value:');
        disp(find(obj == maxx));
        disp('Solution(s) corresponding to maximum value:');
        disp(sol(obj == maxx, :));
    else
        disp('No feasible solution found.');
    end
else
    error('Not a valid condition.');
end
```