```
%%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
```

ALTERNATIVE

```
clc
clear all
format short
%% Phase:1- Input the parameters
c=[5 3];
A=[1 1;5 2;2 8]
b=[2;10;12]

%% Phase:2- Plotting the Graph

P=max(b);
x1=0:1:P;
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);

plot(x1,x21,'r',x1,x22,'g',x1,x23,'b');
title('x1 vs x2');
xlabel('value of x1');
ylabel('value of x2');
legend('x1+x2=2','5x1+2x2=10','2x1+8x2=12');

%% Phase:3- Concepts on axes

cx1=find(x1==0);
c1=find(x21==0);
line1=[x1([c1 cx1]);x21([c1 cx1])];

c2=find(x22==0);
line2=[x1([c2 cx1]);x22([c2 cx1])];
```

```
c3=find(x23==0);
line3=[x1([c3 cx1]);x23([c3 cx1])];

corpt=unique([line1,line2,line3]','rows');
%% Phase:4- Intersection points of constraints

pt=[0;0];
for i=1:size(A,1)
   for j=i+1:size(A,1)
      A1=A([i j],:);
      B1=b([i j]);
      x=A1\B1;
      pt=[pt x];
   end
end
ptt=pt';

%% Phase:5- All corner pts

allpt=[ptt;corpt];
points=unique(allpt,'rows');

%% Phase:6- Feasible Region

for i= 1:size(points,1)
   const1(i)=A(1,1)*points(i,1)+A(1,2)*points(i,2)-b(1);
   const2(i)=A(2,1)*points(i,1)+A(2,2)*points(i,2)-b(2);
   const3(i)=A(3,1)*points(i,1)+A(3,2)*points(i,2)-b(3);
end

s1=find(const1>0);
s2=find(const2>0);
s3=find(const3>0);
s=unique([s1,s2,s3]);
points(s,:)=[];

%% Phase:7 - compute objectuive values

value= points * c';
table=[points value];
[obj,index]=max(value);
x1=points(index,1);
x2=points(index,2);
fprintf('Objective Value is %f at (%f,%f)',obj,x1,x2);
```