% Abdulmalik Almeheini
% MATH246 extracredit HW
% problems 6 and 8 from sec 9.3

clear
clo
warning off all
for alph = 0:0.2:1
  f = @(t, x) (1-alph)*[x(1)*(1 - x(1) - x(2));
                   x(2)*(3 - x(1) - 2*x(2))];
  x(2)*x(1) + alph*[1 - x(1) - x(2)];
  x(2)*x(1) + alph*[1/2 - (3/4)*x(1) - (1/4)*x(2)];
figure; hold on
for a = -2.25:0.25:1.75
  for b = -2.5:0.5:4
    [t, xa] = ode45(f, [0 10], [a b]);
    plot(xa(:,1), xa(:,2));
    [t, xa] = ode45(f, [0 -5], [a b]);
    plot(xa(:,1), xa(:,2));
  end
end
axis([-3 4 -3 4])
xlabel 'x'
ylabel 'y'
title 'trajectories of the systems in problem 6 and 8'
end
From the graphs above we can see how the systems in problem 6 change to become the systems in problem 8 as alpha changes from 0 to 1. The reason behind using alpha is to see how the systems behave as they change from a state to another (which the two states here are problems 6 and 8.)

First (stationary/critical point locations):

The first graph shows the trajectories of the systems when alpha = 0 which is exactly the trajectories of the systems in problem 6. As we can see from the first graph that the system when alpha = 0 has 4 critical points at (0,0), (1,0), (0,3/2), and (-1,2). And when we jump to see the systems when the alpha = 1 we can see that it still have 4 critical points, however, two of them are different and two are the same (0,0) and (1,0).

And to see how the systems behaved while changing from problem 6 to problem 8 we change the value of alpha from 0 to 1 with increment of 0.2. This changing in the value give us four more graphs that tell us the story behind these systems.

If we look at the systems trajectories when alpha is 0.2, 0.4 and 0.6, we can see the all of them share two of the critical points (0,0) and (1,0). However, they all have different values for the other two critical points. One thing we can notice in these different critical points is that as alpha goes from 0 to 0.6 they are changing every time to the same directions, for example, when alpha = 0 the systems started with (0,0), (1,0), (0,3/2) and (-1,2), and as alpha increased to 0.6 two of them changed, the critical point (0,3/2) changed to (0,50/33) then (0,20/13) then at alpha = 0.6 two of them share two of the critical points (0,0) and (1,0).

However, they all have different values for the other two critical points. One thing we can notice in these different critical points is that as alpha goes from 0 to 0.6 they are changing every time to the same directions, for example, when alpha = 0 the systems started with (0,0), (1,0), (0,3/2) and (-1,2), and as alpha increased to 0.6 two of them changed, the critical point (0,3/2) changed to (0,50/33) then (0,20/13) then at alpha = 0.6 two of them share two of the critical points (0,0) and (1,0).

However, when alpha changes to 0.8 and then to 1 something strange happens to the systems. This strange thing occurs to one of the critical points. We know that two of the critical points were moving and each one was moving at a specific direction however, when alpha changes to 0.8 the critical point (-11/2, 13/2) moves all the way from the second quadrant to the fourth quadrant to become (2, -1) and when alpha changes to 1 it jumps to the first quadrant to become (1/2, 1/2).

That's how these systems went from the first state to the second. And to illustrate these changes I solved for the critical points for the systems with each different value of alpha.

These are the critical/stationary points and corresponding solution for the systems as alpha goes from 0 to 1:

```
clear
clo
for alph = 0:0.2:1
syms x y
```
syms x y
S1 = x*(1 - x - y);
S2 = (1-alph)*y*(3 - x - 2*y) + alph*y*(1/2 - (3/4)*x - (1/4)*y);
[xc, yc] = solve([S1, S2, x, y]);
disp('Critical points:'); disp([xc yc])
A = jacobian([S1 S2], [x y]);
evals = eig(A);
end

Critical points:
[  0,   0]
[  1,   0]
[  0, 3/2]
[ -1,   2]

Critical points:
[  0,   0]
[  1, 50/33]
[ -17/14, 31/14]

Critical points:
[  0,   0]
[  1, 5/33]
[ -7/4, 11/4]

Critical points:
[  0,   0]
[  1, 3/2]
[  2, -1]

Critical points:
[  0,   0]
[  0, 2]
[  1, 0]
[  1/2, 1/2]

Second: now we solve for the eigen values for the systems to show how the stability of these points change as alpha change

clear
clo

% when alpha = 0
syms y
alph = 0;
S1 = x*(1 - x - y);
S2 = (1-alph)*y*(3 - x - 2*y) + alph*y*(1/2 - (3/4)*x - (1/4)*y);
[xo, yo] = solve([S1, S2, x, y]);
disp('Critical points:'); disp([xo yo]);
A = jacobian([S1 S2], [x y]);
evals = eig(A);
disp('Eigenvalues at (0,0):');
disp(double(subs(evals, {x, y}, {0, 0})));
disp('Eigenvalues at (1,0):');
disp(double(subs(evals, {x, y}, {1, 0})));
disp('Eigenvalues at (0,3/2):');
disp(double(subs(evals, {x, y}, {0, 3/2})));
disp('Eigenvalues at (-1,2):');
disp(double(subs(evals, {x, y}, {-1, 2})));
Eigenvalues at (1,0);
  -1
  2

Eigenvalues at (0,3/2);
  -3.0000
  -0.5000

Eigenvalues at (-1,2);
  -3.5616
  0.5616

% From the eigen values we can conclude that at the point (0,0) a we have
% a node source b/c both eigen values are real and positive values so its unstable, and at the
% points (1,0) and (-1,2) we have saddles, which they are also unstable
% b/c one of the eigen values is negative and the other is positive
% and finally at the point (0,3/2) we have a node sink (stable) b/c both eigen values are
% real and negative.

clear
clc

% When alpha = 0.2
syms y
alph = 0.2;
S1 = x*(1 - x - y);
S2 = (1-alph)*y*(3 - x - 2*y) + alph*y*(1/2 - (3/4)*x - (1/4)*y);
[xc, yc] = solve(S1, S2, x, y);
disp('Critical points:'); disp([xc yc]);
A = jacobian([S1 S2], [x y]);
evals = eig(A);
disp('Eigenvalues at (0,0):');
disp(double(subs(evals, {x, y}, {0, 0})));
disp('Eigenvalues at (1,0):');
disp(double(subs(evals, {x, y}, {1, 0})));
disp('Eigenvalues at (0,50/33):');
disp(double(subs(evals, {x, y}, {0, 50/33})));
disp('Eigenvalues at (-17/14,31/14):');
disp(double(subs(evals, {x, y}, {-17/14, 31/14})));
% when alpha = 0.4
syms x y
alph = 0.4;
S1 = x*(1 - x - y);  
S2 = (1-alph)*y*(3 - x - 2*y) + alph*y*(1/2 - (3/4)*x - (1/4)*y);
[xc, yc] = solve(S1, S2, x, y);
disp('Critical points:'); disp([xc yc]);
A = jacobian([S1 S2], [x y]);
evals = eig(A);
disp('Eigenvalues at (0,0):');
disp(double(subs(evals, {x, y}, {0, 0})))
disp('Eigenvalues at (1,0):');
disp(double(subs(evals, {x, y}, {1, 0})))
disp('Eigenvalues at (0,20/13):');
disp(double(subs(evals, {x, y}, {0, 20/13})))
disp('Eigenvalues at (-7/4,11/4):');
disp(double(subs(evals, {x, y}, {-7/4, 11/4})))

Critical points:
[ 0, 0]
[ 1, 0]
[ 0, 20/13]
[ -7/4, 11/4]
evals =
3/2 - (9*y)/5 - ((121*x^2)/100 + (2*x*y)/25 + (11*x)/5 + (64*y^2)/25 - (16*y)/5 + 1/2)/(1/2) - (29*x)/20
((121*x^2)/100 + (2*x*y)/25 + (11*x)/5 + (64*y^2)/25 - (16*y)/5 + 1/2)/(1/2) - (29*x)/20 + 3/2
Eigenvalues at {0,0}:
1
2
Eigenvalues at {1,0}:
-1.0000
1.0000
Eigenvalues at {0,20/13}:
-2.0000
-0.5385
Eigenvalues at {-7/4,11/4}:
-2.5731
0.7481

% also when we see the aigen values of the systems when alpha = 0.4
% we see that thier stability havent changed, where
% (0,0),(1,0) and (-7/4,11/4) are unstabel and (0,20/13) is stable ,so
% thier locations only have changed.

clear
clo

% when alpha = 0.6
syms x y
alph = 0.6;
S1 = x*(1 - x - y);  
S2 = (1-alph)*y*(3 - x - 2*y) + alph*y*(1/2 - (3/4)*x - (1/4)*y);
[xc, yc] = solve(S1, S2, x, y);
disp('Critical points:'); disp([xc yc]);
A = jacobian([S1 S2], [x y]);
evals = eig(A);
disp('Eigenvalues at (0,0):');
disp(double(subs(evals, {x, y}, {0, 0})))
disp('Eigenvalues at (1,0):');
disp(double(subs(evals, {x, y}, {1, 0})))
disp('Eigenvalues at (0,30/19):');
disp(double(subs(evals, {x, y}, {0, 30/19})))
disp('Eigenvalues at (-11/2,13/2):');
disp(double(subs(evals, {x, y}, {-11/2,13/2})))

Critical points:
[ 0, 0]
[ 1, 0]
[ 0, 30/19]
[ -11/2, 13/2]
evals =
5/4 - (29*y)/20 - ((529*x^2)/400 + (133*x*y)/100 + (23*x)/20 + (81*y^2)/100 - (9*y)/10 + 1/4)^(1/2)/2 - (57*x)/40

Eigenvalues at \( (0,0) \):
1.0000
1.5000

Eigenvalues at \( (1,0) \):
-1.0000
0.6500

Eigenvalues at \( (0,30/19) \):
-1.5000
-0.5789

Eigenvalues at \( (-11/2,13/2) \):
-2.2582
1.5832

% also when alpha = 0.6 the critical points stability dont change.

clear
clc
% when alpha = 0.8
syms x y
alph = 0.8;
S1 = x*(1 - x - y);
S2 = (1-alph)*y*(3 - x - 2*y) + alph*y*(1/2 - (3/4)*x - (1/4)*y);
[xc, yc] = solve(S1, S2, x, y);
disp('Critical points:'); disp([xc yc]);
A = jacobian([S1 S2], [x y]);
evals = eig(A)
disp('Eigenvalues at (0,0):');
disp(double(subs(evals, {x, y}, {0, 0})))
disp('Eigenvalues at (1,0):');
disp(double(subs(evals, {x, y}, {1, 0})))
disp('Eigenvalues at (0,5/3):');
disp(double(subs(evals, {x, y}, {0, 5/3})))
disp('Eigenvalues at (2,-1):');
disp(double(subs(evals, {x, y}, {2, -1})))

Critical points:
[ 0, 0]
[ 1, 0]
[ 0, 5/3]
[ 2, -1]

evals =
1 - (11*y)/10 - ((36*x^2)/25 + (68*x*y)/25 + y^2/25)^(1/2)/2 - (7*x)/5
((36*x^2)/25 + (68*x*y)/25 + y^2/25)^(1/2)/2 - (11*y)/10 - (7*x)/5 + 1

Eigenvalues at \( (0,0) \):
1
1

Eigenvalues at \( (1,0) \):
-1.0000
0.2000

Eigenvalues at \( (0,5/3) \):
-1.0000
-0.6667

Eigenvalues at \( (2,-1) \):
-1.0000
-0.4000

% But when we look at the eigen values of the systems when alpha = 0.8 we % notice that the stability of the critical point that changes it location % from the second quadrant to the fourth qudrant also changes, which it % became stable and switch to nodel sink. while nothing happens to the % stability of the other points.
% and when we look at the eigen values of the systems when alpha=1 we also 
% see that these changes continue happening on some of the points. first, 
% we know that the point that changed its location from the second to the 
% fourth quadrant moves to the first quadrant when alpha = 1, however, what 
% we see here that its changes it stability while its moving so it return 
% to be unstable again. and another thing is the point (1,0) that havent 
% changed at all while alpha changed from 0 to 0.8 finally becomes stable 
% when alpha = 1 and i think this changes occurs b/c of the point that 
% comes close to it.