Homework #2
Salvatore Zerbo

Problem 1

```
% Initial matrix
A = [1, 2, 3, 4; 3, 1, 3, 0; 1, 3, -3, -8];

% Create tableau
T = totbl(A);

% Exchange variables

T = ljx(T, 1, 1);
T = ljx(T, 2, 2);
T = ljx(T, 3, 3);

%% Now do again for A'
% Initial matrix
A = [1, 2, 3, 4; 3, 1, 3, 0; 1, 3, -3, -8]';

% Create tableau
T = totbl(A);

% Exchange variables
T = ljx(T, 1, 1);
T = ljx(T, 2, 2);
T = ljx(T, 3, 3);
```

Output =

	y_1	y_2	y_3	x_4
$x_1 =$	-0.3333	0.4167	0.0833	2.0000
$x_2 =$	0.3333	-0.1667	0.1667	0.0000
$x_3 =$	0.2222	-0.0278	-0.1389	-2.0000

	y_1	y_2	y_3
$x_1 =$	-0.3333	0.3333	0.2222
$x_2 =$	0.4167	-0.1667	-0.0278
$x_3 =$	0.0833	0.1667	-0.1389
y_4	-2.0000	0.0000	2.0000

All three of the rows in matrix A are linearly independent. There are also three columns of matrix A that are linearly independent. The one linearly dependent column has the equation $y_4 = -2y_1 + 2y_3$.

Problem 2

```
% Initial matrix
A = [1, 1, 1; 1, -1, -1; 1, -1, 1];
b = [1; 1; 3];

% Create tableau
T = totbl(A, b);

% Exchange Variables
T = ljx(T, 1, 1);
T = ljx(T, 2, 2);
T = ljx(T, 3, 3);
```

Output =

	y_1	y_2	y_3	1
$x_1 =$	0.5000	0.5000	0.0000	1.0000
$x_2 =$	0.5000	0.0000	-0.5000	-1.0000
$x_3 =$	-0.0000	-0.5000	0.5000	1.0000

Problem 3

```
%% Part 1
% Initial matrix
A = [2, -1, 1, 1; -1, 2, -1, -2; 4, 1, 1, -1];
a = [1; 1; 5];

% Create tableau
T = totbl(A, a);

% Exchange Variables
T = ljx(T, 1, 1);
T = ljx(T, 2, 2);
```

Output =

	y_1	y_2	x_3	x_4	1
$x_1 =$	0.6667	0.3333	-0.3333	0.0000	1.0000
$x_2 =$	0.3333	0.6667	0.3333	1.0000	1.0000
$y_3 =$	3.0000	2.0000	0.0000	0.0000	0.0000

Since the final column corresponding to y_3 is equal to 0 and x_3 and x_4 are independent, there are infinitely many solutions. x_3 and x_4 can arbitrarily be chosen to characterize the solution set by the following equations:

$$x_1 = -\frac{1}{3}x_3 + 1$$
$$x_2 = \frac{1}{3}x_3 + x_4 + 1$$

```
%% Part 2
% Initial matrix
B = [1, -1, 1, 2; 1, 1, 0, -1; 1, -3, 2, 5];
b = [2; 1; 1];

% Create tableau
T = totbl(B, b);

% Exchange Variables
T = ljx(T, 1, 1);
T = ljx(T, 2, 2);
```

Output =

	y_1	y_2	x_3	x_4	1
$x_1 =$	0.5000	0.5000	-0.5000	-0.5000	1.5000
$x_2 =$	-0.5000	0.5000	0.5000	1.5000	-0.5000
$y_3 =$	2.0000	-1.0000	0.0000	0.0000	2.0000

Since the pivot elements corresponding to x_3 and x_4 are 0, we are not able to continue with moving y_3. Also, since the element in the final column corresponding to y_3 is non-zero, this means that the system has no solutions. The linear relationship is as follows:

$$y_3 = 2y_1 - y_2 - 2$$

```
%% Part 3
% Initial matrix
C = [1, -1, 1; 2, 1, 1; -1, -1, 2; 1, 1, -1];
c = [3; 2; 2; -1];

% Create tableau
T = totbl(C, c);

% Exchange Variables
T = ljx(T, 1, 1);
T = ljx(T, 2, 2);
T = ljx(T, 3, 3);
```

Output =

	y_1	y_2	y_3	1
$x_1 =$	0.4286	0.1429	-0.2857	1.0000
$x_2 =$	-0.7143	0.4286	0.1429	-1.0000
$x_3 =$	-0.1429	0.2857	0.4286	1.0000
$y_4 =$	-0.1429	0.2857	-0.5714	-0.0000

We have run out of room to move y_4 up, but since the corresponding element in the last column is zero, the system has a final solution of $x_1 = 1$, $x_2 = -1$, $x_1 = 1$. y_4 is linearly dependent on the other y's and follows the relationship:

$$y_4 = -\frac{1}{7}y_1 + \frac{2}{7}y_2 - \frac{4}{7}y_3$$

Problem 4

1) A = $\begin{bmatrix} 2 & 5 \\ 1 & 2 \end{bmatrix}$ has only one solution regardless of b. This is because there are exactly 2 columns and 2 rows, or in other words, there are 2 equations and 2 unknowns. This system is well-determined.

2) A = $\begin{bmatrix} 1 & -1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 2 & -1 \end{bmatrix}$ has either no solution or infinitely many solutions depending on b. Since there is a non-pivot column, then the value in the last column corresponding to the final y left will determine whether the system has no solution

(it is non-zero) or infinitely many solutions (it is zero).

3) A = $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ has either one solution or no solution depending on b. Since the system is over-determined, the constraints from b could make it impossible for the variable to satisfy both equations. It is possible, depending on b, to make the constraints able to be satisfied by the one variable.

4) A = $\begin{bmatrix} 1 & 3 & 2 \\ 3 & 1 & 1 \end{bmatrix}$ has infinitely many solutions regardless of b. This is because the system is under-determined, meaning that you can pick some of the variables arbitrarily and then find the remaining dependent variables to satisfy the constraints.

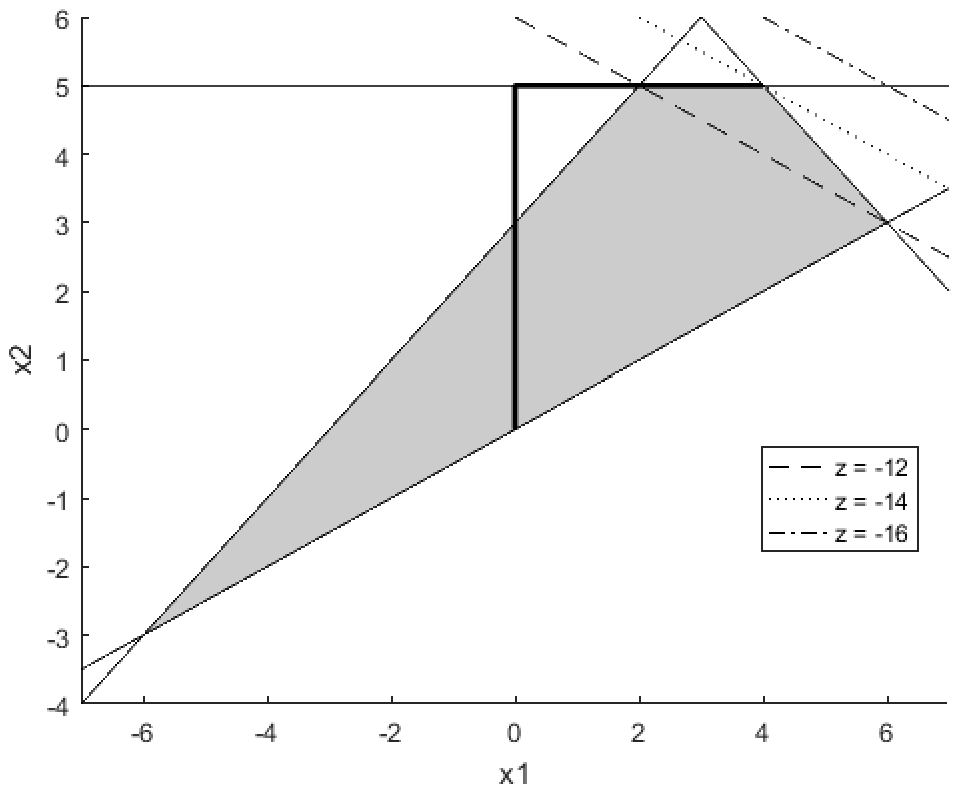
Problem 5

i and ii) Writing out the constrains Ax $\geq$ b yields:

$$\begin{aligned} 0 * x_1 + -1 * x_2 &\geq -5 \\ -1 * x_1 + -1 * x_2 &\geq -9 \\ -1 * x_1 + 2 * x_2 &\geq 0 \\ 1 * x_1 + -1 * x_2 &\geq -3 \end{aligned}$$

Rewriting these:

$$\begin{aligned} x_2 &\leq 5 \\ x_2 &\leq -x_1 + 9 \\ x_2 &\geq \frac{1}{2} x_1 \\ x_2 &\leq x_1 + 3 \end{aligned}$$



The shaded region is the feasible region, the solid lines are the constraints on the system, the thick solid line is the simplex method trace, and the non-solid lines are the contours at specified z-values. Looking at the plot, the solution is graphically determined to be at $x_1 = 4$ and $x_2 = 5$ and yields $z = -14$.

iii)

```
% Initial system
A = [0, -1; -1, -1; -1, 2; 1, -1];
b = [-5; -9; 0; 3];
p = [-1; -2];

% Create tableau
T = totbl(A, b, p);

% Exchange variables
T = ljx(T, 1, 2);
T = ljx(T, 2, 1);

% Data for simplex jumps
simplex_x = [0, 1, 4];
simplex_y = [0, 5, 5];

% Create range of inputs to sample
r = linspace(-7, 7, 1000);

% Z contours
p = [-1; -2];
y5 = (-1 / 2) * (r - 12);
y6 = (-1 / 2) * (r - 14);
y7 = (-1 / 2) * (r - 16);

% Constraint equations
y1 = 5 + 0 * r;
y2 = -r + 9;
y3 = r / 2;
y4 = r + 3;

% Conditions to fill in region
[x, y] = meshgrid(r);   % Get 2-D mesh for x and y based on r
cond1 = (-y >= -5);
cond2 = (-x - y >= -9);
cond3 = (-x + 2*y >= 0);
cond4 = (x - y >= -3);
conditions = cond1 & cond2 & cond3 & cond4;

% Plot
hold on;
colors = zeros(size(x)) + cond1 + cond2 + cond3 + cond4;
plot(x(colors == 4), y(colors == 4), 'color', [0, 0, 0]+0.8);
plot(r, y1, 'k');
plot(r, y2, 'k');
plot(r, y3, 'k');
plot(r, y4, 'k');
h1 = plot(r, y5, 'k--');
h2 = plot(r, y6, 'k:');
h3 = plot(r, y7, 'k-.');
plot(simplex_x, simplex_y, 'k', 'LineWidth', 2)
legend([h1, h2, h3], {'z = -12', 'z = -14', 'z = -16'}, 'Location', 'best')

xlabel('x1')
ylabel('x2')
xlim([-7, 7])
ylim([-4, 6])
```

Output =

	x_4	x_3	1
$x_2 =$	-0.0000	-1.0000	5.0000
$x_1 =$	-1.0000	1.0000	4.0000
$x_5 =$	1.0000	-3.0000	6.0000
$x_6 =$	-1.0000	2.0000	-4.0000
z =	1.0000	1.0000	-14.0000

We can see that the solution in the above table matches the graphical solution from part ii.