```
% MECH 6318 - HW 2
% Jonas Wagner
% 2021-09-07

clear
close all
```

Problem 1

```
disp('Problem 1 -----------------------------')
```

```
Problem 1 -----------------------------
```

```
% Wrote a seperate script for algorithm: simplex.m

% Problem 11.8 setup
f = [1;
    2;
    -7]
```

```
f = 3×1
     1
     2
    -7
```

```
A = [2, 1, 1;
    -1, 2,-1;
    1, 5, 5]
```

```
A = 3×3
     2     1     1
    -1     2    -1
     1     5     5
```

```
b = [15;
    7;
    25]
```

```
b = 3×1
    15
     7
    25
```

```
% Simplex Algorithm
[x_min, f_min, n_iter,T] = simplex(f,A,b)
```

```
Initial T =
     2     1     1     1     0     0     0    15
    -1     2    -1     0     1     0     0     7
     1     5     5     0     0     1     0    25
     1     2    -7     0     0     0     1     0

Final T =
     2     1     1     1     0     0     0    15
```

```
     1     3     0     1     1     0     0    22
    -9     0     0    -5     0     1     0   -50
    15     9     0     7     0     0     1   105
x_min = 3×1
     0
     0
    15
f_min = -105
n_iter = 1
T = 4×8
     2     1     1     1     0     0     0    15
     1     3     0     1     1     0     0    22
    -9     0     0    -5     0     1     0   -50
    15     9     0     7     0     0     1   105
```

Problem 2

```
disp('Problem 2 -----------------------------')
```

```
Problem 2 -----------------------------
```

```
% Problem setup
f = [3;
     0;
     1;
     0;
     2]
```

```
f = 5×1
     3
     0
     1
     0
     2
```

```
A = [];
b = [];
C = [1, 0, 1,-1, 1;
     0, 1,-2, 3, 2]
```

```
C = 2×5
     1     0     1    -1     1
     0     1    -2     3     2
```

```
d = [-1;
     -2]
```

```
d = 2×1
    -1
    -2
```

```
l = 0;
u = inf;

n = size(f,1);
m = size(C,1);
```

```
% Dual Simplex Method

% Positive right side...
A = diag(sign(d)) * C
```

```
A = 2×5
    -1     0    -1     1    -1
     0    -1     2    -3    -2
```

```
b = diag(sign(d)) * d
```

```
b = 2×1
     1
     2
```

```
% Fake State variables
Y = eye(m)
```

```
Y = 2×2
     1     0
     0     1
```

```
% Table Construct
T = [[A,            eye(m),     zeros(m,1), b];
     f',            zeros(1,m), 1,          0;
     zeros(1,n),    ones(1,m),  1,          0]
```

```
T = 4×9
    -1     0    -1     1    -1     1     0     0     1
     0    -1     2    -3    -2     0     1     0     2
     3     0     1     0     2     0     0     1     0
     0     0     0     0     0     1     1     1     0
```

```
for i = 1:m
    T(end,:) = T(end,:) - T(i,:);
end
T
```

```
T = 4×9
    -1     0    -1     1    -1     1     0     0     1
     0    -1     2    -3    -2     0     1     0     2
     3     0     1     0     2     0     0     1     0
     1     1    -1     2     3     0     0     1    -3
```

```
% Phase 1
min_col = 3; %1,1,-1,2,3,0,0,1
min_row = 3; %Ratios:inf,1,0

% Pivoting
new_T = zeros(size(T));
new_T(min_row,:) = T(min_row,:)/T(min_row,min_col);
for row = 1:size(T,1)
    if row ~= min_row
```

```
        new_T(row,:) = T(row,:) ...
            - T(row,min_col) * new_T(min_row,:);
    end
end
T = new_T
```

```
T = 4×9
     2     0     0     1     1     1     0     1     1
    -6    -1     0    -3    -6     0     1    -2     2
     3     0     1     0     2     0     0     1     0
     4     1     0     2     5     0     0     2    -3
```

```
% Optimal solution...
x3 = 0;
y1 = 1;
y2 = 2;

w = y1 + y2
```

```
w = 3
```

```
if w > 0
    disp('w > 0, infeasable')
end
```

```
w > 0, infeasable
```

```
% Confirmation:
disp('Confirmaiton with Linprog:')
```

```
Confirmaiton with Linprog
```

```
[~,~,exitflag]=linprog(f,[],[],C,d,zeros(n,1))
```

```
exitflag = -2
```

```
disp('exitflag = -2 => no feasable solution found')
```

```
exitflag = -2 => no feasable solution found
```