1.1 Part b

Solve problem P by using the Matlab function linprog. Pay particular attention to developing, demonstrating, and explaining the correct set-up of linprog's arguments and return values.

The `linprog` function in MATLAB takes the following format

`[X,FVAL,EXITFLAG] = linprog(f, A, b, Aeq, beq, LB, UB)`

The following code snippet explains the process for determining each of the input values

```
% By default, linprog finds the minimum value of any function put into it.
% To make it a maximisation problem, the coeffecients must be inversed.
f = [-3; -2; -1];

% A, b represent the linear inequality constraints. A is the coefficients
% and b is the target problems.
A = [2 2 1;
     1 3 3];

b = [8;
     15];

% Aeq, beq represent the linear equality constraints. In this problem there
% are none so empty arrays are passed through.
Aeq = [];
beq = [];

% LB stands for lower bounds, which is outlined in the final constraint
LB = zeros(length(f), 1);

% UB stands for upper bounds, no upper bounds have been specified.
UB = [];

% The linprog function can now be applied
[X,FVAL,EXITFLAG] = linprog(f, A, b, Aeq, beq, LB, UB);
```

```
Optimal
solution found.
```

```
% Success, a solution has been derived.
fprintf('The objective value of the function is %d', FVAL)
```

```
The objective value of the function is -12
```

```
fprintf('The solutions are \n x1:%d \n x2:%d \n x3:%d', X(1), X(2), X(3))
```

```
The solutions are
 x1:4
 x2:0
 x3:0
```

1.4 Part d

Verify the complementary slackness theorem for the optimal solutions derived for P and D.

The complementary slackness theorem states that: *if x and y are corresponding feasible solutions to a Primal-Dual pair of Linear Programming problems then both are optimal if and only if*

$$(1.10) \qquad y^T(\mathrm{A}x - b) = 0$$

$$(1.11) \qquad x^T(A^T y - c) = 0$$

```
% Input solutions derived in earlier steps
y = [1.5; 0];
x = [4; 0; 0];

% C is the constraint values
c = [3; 2; 1];

% Apply equations 4 and 5 detailed above
equ110 = y'*(A*x - b);
equ111 = x'*(A'*y - c);

if equ110 == 0 && equ111 == 0
    disp('Optimal solutions have been found!!')
else
    disp('Optimal solutions have not been found')
end
```

```
Optimal solutions have been found!!
```