```
% Write variables in the order:
% xA1 xB1 xC1 xD1 ...
% xA2 xB2 xC2 xD2 ...
% xA3 xB3 xC3 xD3 ...
% xA4 xB4 xC4 xD4 ...
% xA5 xB5 xC5 xD5 ...
% xA6 xB6 xC6 xD6

% 4*6 variables -> 24 columns
% 1 time constaint for each A, B, C, and D
% 1 demand constraint for each 1,2,3,4,5,6
% 1 non-negative constraint for each of the 24 variables
% 4 + 6 + 24 constraints -> 34 rows
A = zeros(34,24);

% set time constraints
A(1,1:4:24) = 1;                    % sum(A) <= 24
A(2,2:4:24) = 1;                    % sum(B) <= 24
A(3,3:4:24) = 1;                    % sum(C) <= 24
A(4,4:4:24) = 1;                    % sum(D) <= 24

% set demand constraints
A(5,1:4) = -[523 721 639 0];        % sum(P1) >= 21000
A(6,5:8) = -[419 615 492 0];        % sum(P2) >= 5000
A(7,9:12) = -[563 709 515 0];       % sum(P3) >= 2000
A(8,13:16) = -[315 0 0 330];        % sum(P4) >= 1200
A(9,17:20) = -[292 0 0 407];        % sum(P5) >= 1400
A(10,21:24) = -[312 314 309 291]; % sum(P6) >= 4300

% set non-negative constraints
A(34-24+1:34,:) = -eye(24);         % x_ij >= 0

% print A to check that it is correct
A

b = [24 24 24 24 ...
    -21000 -5000 -2000 -1200 -1400 -4300 ...
    0 0 0 0 ...
    0 0 0 0 ...
    0 0 0 0 ...
    0 0 0 0 ...
    0 0 0 0 ...
    0 0 0 0];

c = [2.8*523  2.8*721 2.8*639 2.8*0 ...
     2.5*419  2.5*615 2.5*492 2.5*0 ...
     1.9*563  1.9*709 1.9*515 1.9*0 ...
     2.9*315  2.9*0   2.9*0   2.9*330 ...
     5.75*292 5.75*0  5.75*0  5.75*407 ...
```

```
     0.9*312  0.9*314 0.9*309 0.9*291];

sol1 = linprog(-c, A, b);

% print the solution and the profit
sol1
c*sol1
```