Question 2

We suppose that there are p students in a class each of whom must be allocated to work in a group on exactly one of *q* problems. Each student i has expressed a preference order for each problem j, taking values 1, 2, . . . , *q*, where 1 denotes the student's top preference and *q* denotes their least favourite. Assume the preferences are provided in a *p*×*q* matrix C_{ij} . The binary decision variables x_{ij} indicate whether student *i* is matched to problem *j*, and can be unravelled as a vector *x*. We suppose that the objective is to minimise

$$\frac{1}{p}\sum_{i,j} C_{\text{ij}}x_{\text{ij}}$$

i.e., the preference order of the problem actually allocated to each student, averaged

over the class.

```matlab
% Define problem variables p, q, C
p = 60;
q = 14;

C = zeros(p, q);
for i = 1 : p
    preferences = 1 : q;
    preferences = preferences(randperm(length(preferences)));
    C(i, :) = preferences;
end
C
```

```
C = 60×14
     6     9    11    10     7    13     8     1     4     2    12    14     3 ···
     4    12     2    10    14     7     1     8     5     3    11    13     9
    13    10    11     3     5     8    14     6     2     7     1    12     9
    14    12     2     8    11     6     4     3    13     5     1     7     9
    12     4    13    14    11     5     8     3     9     1    10     7     2
     3    10     5    11     9     6     8     4    13     1    14     7    12
    14    13    10     9     7     5     2    12     3     6     8    11     1
    13     1     2     7    11     9     5     8     4    10     6    12     3
     2    11     3     8    10     1     6     4    13     9     7    12     5
     6    12    14     7    13     9    10     4     2    11     8     5     3
     ⋮
```

```matlab
% Unravel matrix C
f = reshape(C', [], 1);
```

a) Write down conditions on x that ensure each student is allocated to exactly one problem. Exhibit and explain Matlab code for use with intlinprog that builds the associated constraint matrix.

```matlab
f_length = length(f);
% Define student constraints
A_circ = zeros(1, f_length);
```

```
A_circ(1:q) = 1;
circ = functions.circulant(A_circ, 1);

Aeq = A_circ;
for i = 1:p-1
    rowIndex = q * i + 1;
    row = circshift(A_circ, rowIndex)
    row = circ(rowIndex, :);
    Aeq = [Aeq; row];
end

beq = ones(p, 1);
```