Question 3

3a) The given skeleton code requires us to fill up m, k and c.

m would be the value of our last 5 student ID digits. Since my student ID is 32194471, the value of m would be 94471 and in a row vector form.k is the value that is given which is 10000 multiplied by the matrix and c would be 200 multiplied by the matrix.

The code that determines the matrices M, K and C is

$$M = \begin{bmatrix} m1 & 0 & 0 & 0 & 0 \\ 0 & m2 & 0 & 0 & 0 \\ 0 & 0 & m3 & 0 & 0 \\ 0 & 0 & 0 & m4 & 0 \\ 0 & 0 & 0 & 0 & m5 \end{bmatrix}$$

$$K = \begin{bmatrix} -k1-k2 & k2 & 0 & 0 & 0 \\ k2 & -k2-k3 & k3 & 0 & 0 \\ 0 & k3 & -k3-k4 & k4 & 0 \\ 0 & 0 & k4 & -k4-k5 & k5 \\ 0 & 0 & 0 & k5 & -k5 \end{bmatrix}$$

$$C = \begin{bmatrix} -c1-c2 & c2 & 0 & 0 & 0 \\ c2 & -c2-c3 & c3 & 0 & 0 \\ 0 & c3 & -c3-c4 & c4 & 0 \\ 0 & 0 & c4 & -c4-c5 & c5 \\ 0 & 0 & 0 & c5 & -c5 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 \\ M^{-1}K & M^{-1}C \end{bmatrix}$$

In MATLAB, the command will be

```
m = 100*[9,4,4,7,1];
k = 10000*ones(1,N);
c = 200*ones(1,N);
M = diag(m);
K = diag(-k-[k(2:end),0]) + diag(k(2:end),1) + diag(k(2:end),-1);
C = diag(-c-[c(2:end),0]) + diag(c(2:end),1) + diag(c(2:end),-1);
A = [zeros(N,N),eye(N,N); M^(-1)*K,M^(-1)*C];
```

3b) Referring back to Module 2 (Numerical Module) Euler's Method, we can solve the system of second order equations with the following method

$$x_i = x_{i-1} + t_{step} * x' (x_{i-1}, t_{i-1})$$

In MATLAB, the command will be

```
x(:,i) = x(:,i-1) + dt* RHS(x(:,i-1),t(i-1));
```

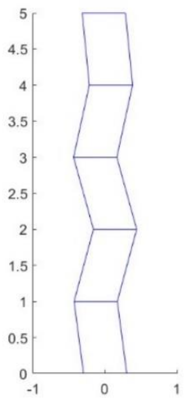
3c) As stated in the assignment sheet, the solution X(t) is the displacement of each floor relative to the ground which gives an indication of the extent to which the building is warped between each floor. Thus, our displacement will be

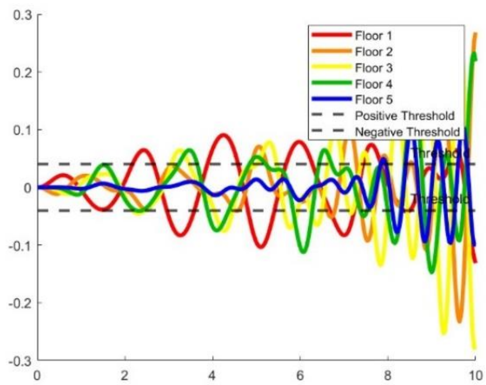
Displacement = [X1-X0; X2-X1; X3-X2; X4-X3; X5-X4]

In MATLAB, we can type the following command

```
displacement = x(1:N,:) - [zeros(1,Nt); x(1:N-1,:)];
```

Graphs obtained for part c





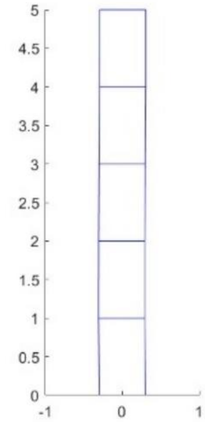
We can see from the graph above that the relative displacement of each floor will oscillate with increasing amplitude, crossing over the threshold many times, making the building structurally unsound. This would be due to the inaccuracies of Euler's Method. Hence, we have to use a better method in order to get a more reasonable solution

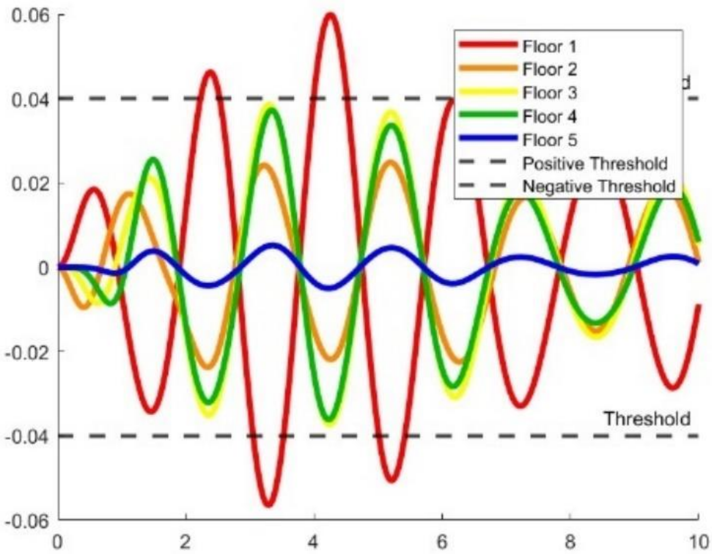
Code for part c

```
% Written by Tan Jin Chun (32194471)
% Last Modified: 12/5/22
% Q3c

clear all;close all;clc

%parameters
% Initializing my parameters (Instructions given in the assignment sheet)
% My student ID is 32194471
N = 5;
m = 100*[9,4,4,7,1]; % part (a) and later part (e)
k = 10000*ones(1,N); % part (a) and later part (e)
c = 200*ones(1,N);   % part (a) and later part (e)

%PS you may need to add more lines here to complete part (e). Please
%indicate these new lines with a comment.

% Using diag to create diagonal matrices
M = diag(m);                                                % part (a)
K = diag(-k-[k(2:end),0]) + diag(k(2:end),1) + diag(k(2:end),-1);% part (a)
C = diag(-c-[c(2:end),0]) + diag(c(2:end),1) + diag(c(2:end),-1);% part (a)

% Given Values
a = 0.3; omega = 3;
f = @(t) [a*cos(omega*t);zeros(N-1,1)];

% X'' = M^{-1}KX + M^(-1)CX'+F, X(0)=0 and X'(0)=0
% Rewrite this equation so that it is in the form x' = [X;X']' = Ax + F(t)
% by finding A (I have supplied F and the initial condition for you)
A = [zeros(N,N),eye(N,N); M^(-1)*K,M^(-1)*C];  % part (a)
F = @(t) [zeros(N,1); f(t)];

%initial condition
x0 = zeros(2*N,1);

%rate of change of state x
RHS = @(x,t) A*x + F(t);

%solver parameters and domain
Nt = 300;
tmax = 10;
t = linspace(0,tmax,Nt);
dt = t(2)-t(1);
x = zeros(2*N,Nt); %here we store the solution at each timestep in separate (Nt)
columns

 %set initial condition to solution in first column
 x(:,1) = x0;

 %iterate Eulers method for every timestep and later iterate Heun's method
 %for accuracy.
 for i = 2:Nt

    x(:,i) = x(:,i-1) + dt* RHS(x(:,i-1),t(i-1)); % part (b) and then part (d)
(two different codes, you will probably use more than one line for part (d))

 end
```

```
% No Changes Made to the code below except for "displacement" and
% rectification of xpos and also the addition of two yline to indicate the
% threshold
% Create animation of building --- you do not need to touch this.
xpos = x(1:N,:);
halfwidth = 0.3;
figure(1),
for i = 1:Nt
   cla, hold on
    plot(xpos(:,i)-halfwidth,[1:N],'b')
    plot(xpos(:,i)+halfwidth,[1:N],'b')
    plot([0, xpos(1,i)]+halfwidth,[0, 1],'b')
    plot([0, xpos(1,i)]-halfwidth,[0, 1],'b')
    for j = 1:N
        plot([xpos(j,i)+halfwidth, xpos(j,i)-halfwidth],[j, j],'b')
        plot([xpos(j,i)+halfwidth, xpos(j,i)-halfwidth],[j, j],'b')
    end

    axis equal
    xlim([-1 1])
    ylim([0 N])
drawnow
end


%Create figure showing the inter-floor displacement from each floor to the
%floor below it (in the case of Floor 1, it is the displacement of Floor 1
%compared to the ground).
figure (2)
cla
hold on
displacement = x(1:N,:) - [zeros(1,Nt); x(1:N-1,:)];%part c
for i = 1:N
    plot(t' ,displacement(i,:),'LineWidth',3)
end
newcolors = {'#F00','#F80','#FF0','#0B0','#00F'};
colororder(newcolors)

% Added Code
yline(0.04,'k--','Threshold', 'LineWidth',2);
yline(-0.04,'k--','Threshold','LineWidth',2);
legend('Floor 1','Floor 2','Floor 3','Floor 4','Floor 5','Positive
Threshold','Negative Threshold');
```

3d)





As we can see from the graph above, floor one has clearly exceeded the threshold of 0.04 multiple times. Thus, we can conclude that floor 1 is more likely to break and potentially should be reinforced.

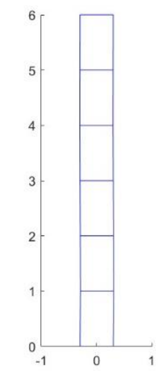
Code for 3d

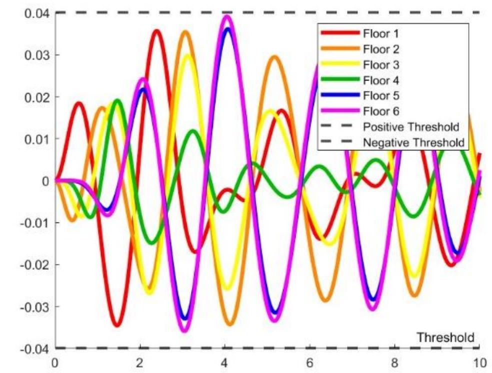
```
% Written by Tan Jin Chun (32194471)
% Last Modified: 12/5/22
% Q3d

clear all;close all;clc

%parameters
% Initializing my parameters (Instructions given in the assignment sheet)
% My student ID is 32194471
N = 5;
m = 100*[9,4,4,7,1]; % part (a) and later part (e)
k = 10000*ones(1,N); % part (a) and later part (e)
c = 200*ones(1,N);   % part (a) and later part (e)

%PS you may need to add more lines here to complete part (e). Please
%indicate these new lines with a comment.

% Using diag to create diagonal matrices
M = diag(m);                                                     % part (a)
K = diag(-k-[k(2:end),0]) + diag(k(2:end),1) + diag(k(2:end),-1);% part (a)
C = diag(-c-[c(2:end),0]) + diag(c(2:end),1) + diag(c(2:end),-1);% part (a)

% Given Values
a = 0.3; omega = 3;
f = @(t) [a*cos(omega*t);zeros(N-1,1)];

% X'' = M^{-1}KX + M^(-1)CX'+F, X(0)=0 and X'(0)=0
% Rewrite this equation so that it is in the form x' = [X;X']' = Ax + F(t)
% by finding A (I have supplied F and the initial condition for you)
A = [zeros(N,N),eye(N,N); M^(-1)*K,M^(-1)*C]; % part (a)
F = @(t) [zeros(N,1); f(t)];

%initial condition
x0 = zeros(2*N,1);

%rate of change of state x
RHS = @(x,t) A*x + F(t);

%solver parameters and domain
Nt = 300;
tmax = 10;
t = linspace(0,tmax,Nt);
dt = t(2)-t(1);
x = zeros(2*N,Nt); %here we store the solution at each timestep in separate (Nt)
columns

 %set initial condition to solution in first column
 x(:,1) = x0;

 %iterate Eulers method for every timestep and later iterate Heun's method
 %for accuracy.

 % Changed the code here from part(c) to Heun's Method
 for i = 2:Nt

     grad_A = RHS(x(:,i-1),t(i-1)); % part (d)
     newpred_x = x(:,i-1) + dt*grad_A;
```

```
    grad_B = RHS(newpred_x,t(i));
    x(:,i) = x(:,i-1) + dt/2 * (grad_A + grad_B); % part (b) and then part (d)
(two different codes, you will probably use more than one line for part (d))

 end

 % No Changes Made to the code below except for "displacement" and
 % rectification of xpos and also the addition of two yline to indicate the
 % threshold
 %Create animation of building --- you do not need to touch this.
 xpos = x(1:N,:);
 halfwidth = 0.3;
  figure(1),
 for i = 1:Nt
    cla, hold on
     plot(xpos(:,i)-halfwidth,[1:N],'b')
     plot(xpos(:,i)+halfwidth,[1:N],'b')
     plot([0, xpos(1,i)]+halfwidth,[0, 1],'b')
     plot([0, xpos(1,i)]-halfwidth,[0, 1],'b')
     for j = 1:N
         plot([xpos(j,i)+halfwidth, xpos(j,i)-halfwidth],[j, j],'b')
         plot([xpos(j,i)+halfwidth, xpos(j,i)-halfwidth],[j, j],'b')
     end

     axis equal
     xlim([-1 1])
     ylim([0 N])
 drawnow
 end


 %Create figure showing the inter-floor displacement from each floor to the
 %floor below it (in the case of Floor 1, it is the displacement of Floor 1
 %compared to the ground).
 figure (2)
 cla
 hold on
 displacement = x(1:N,:) - [zeros(1,Nt); x(1:N-1,:)]; %part c
 for i = 1:N
     plot(t' ,displacement(i,:),'LineWidth',3)
 end
 newcolors = {'#F00','#F80','#FF0','#0B0','#00F'};
 colororder(newcolors)

 % Added Code
 yline(0.04,'k--','Threshold', 'LineWidth',2);
 yline(-0.04,'k--','Threshold','LineWidth',2);
 legend('Floor 1','Floor 2','Floor 3','Floor 4','Floor 5','Positive
Threshold','Negative Threshold');
```

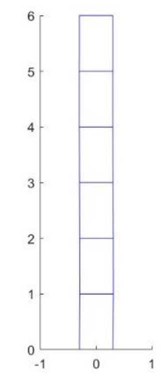
3e)

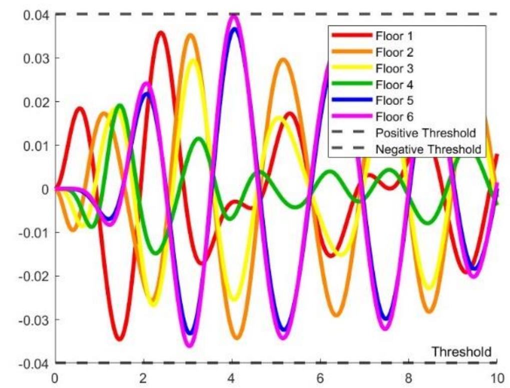
M6 just smaller than this cutoff. M6 has a value of 70000



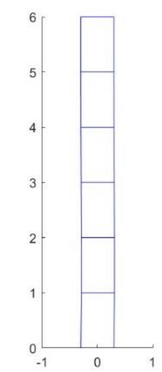


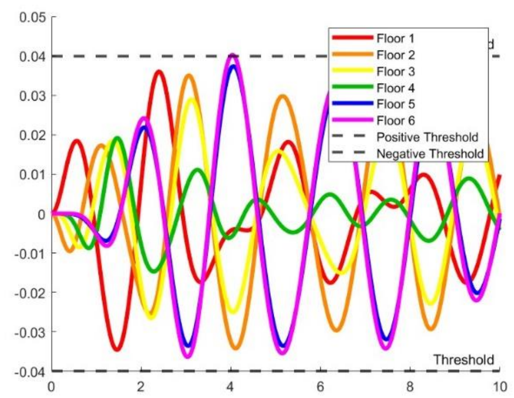
M6 at the cutoff. M6 has a value of 60000





M6 just above the cutoff. M6 has a value of 50000





Based on the visual observations of the graph, we can conclude that the smallest value of M6 will be 70000 units (just below the threshold). We can also see that when the pendulum has a mass of 50000 units, it will just be larger than the cutoff. In a nutshell, when the pendulum is added, the building can be saved if and only if the right weight of the pendulum is placed on the building. As we can see from the graphs, if the pendulum is added, then floor 1 can remain stable and thus maintain the structural integrity of the building. Hence, we can conclude that the pendulum will help to improve the structural integrity of the building (provided the right weight of the pendulum is chosen and the distribution of weight across the building is proper) for my given case.

Code for 3e

```
% Written by Tan Jin Chun (32194471)
% Last Modified: 12/5/22
% Q3e

clear all;close all;clc

%parameters
% Initializing my parameters (Instructions given in the assignment sheet)
% My student ID is 32194471
N = 6; % Add a new floor (for pendulum)
m = 100*[9,4,4,7,1,70]; % part (a) and later part (e) % Play around with the value
of m6 (10000 def val)
k = 10000*ones(1,N); % part (a) and later part (e)
c = [200*ones(1,N-1),20];   % part (a) and later part (e) % c6 is 20 not 200

%PS you may need to add more lines here to complete part (e). Please
%indicate these new lines with a comment.

% Using diag to create diagonal matrices
M = diag(m);                                                       % part (a)
K = diag(-k-[k(2:end),0]) + diag(k(2:end),1) + diag(k(2:end),-1);% part (a)
C = diag(-c-[c(2:end),0]) + diag(c(2:end),1) + diag(c(2:end),-1);% part (a)

% Given Values
a = 0.3; omega = 3;
f = @(t) [a*cos(omega*t);zeros(N-1,1)];

% X'' = M^{-1}KX + M^(-1)CX'+F, X(0)=0 and X'(0)=0
% Rewrite this equation so that it is in the form x' = [X;X']' = Ax + F(t)
% by finding A (I have supplied F and the initial condition for you)
A = [zeros(N,N),eye(N,N); M^(-1)*K,M^(-1)*C];  % part (a)
F = @(t) [zeros(N,1); f(t)];

%initial condition
x0 = zeros(2*N,1);

%rate of change of state x
RHS = @(x,t) A*x + F(t);

%solver parameters and domain
Nt = 300;
tmax = 10;
t = linspace(0,tmax,Nt);
dt = t(2)-t(1);
x = zeros(2*N,Nt); %here we store the solution at each timestep in separate (Nt)
columns

 %set initial condition to solution in first column
 x(:,1) = x0;

 %iterate Eulers method for every timestep and later iterate Heun's method
 %for accuracy.
 for i = 2:Nt

     grad_A = RHS(x(:,i-1),t(i-1)); % part (d)
     newpred_x = x(:,i-1) + dt*grad_A;
     grad_B = RHS(newpred_x,t(i));
```

```
    x(:,i) = x(:,i-1) + dt/2 * (grad_A + grad_B); % part (b) and then part (d)
(two different codes, you will probably use more than one line for part (d))

 end

 % No Changes Made to the code below except for "displacement" and
 % rectification of xpos and also the addition of two yline to indicate the
 % threshold
 % Create animation of building --- you do not need to touch this.
 xpos = x(1:N,:);
 halfwidth = 0.3;
 figure(1),
 for i = 1:Nt
    cla, hold on
     plot(xpos(:,i)-halfwidth,[1:N],'b')
     plot(xpos(:,i)+halfwidth,[1:N],'b')
     plot([0, xpos(1,i)]+halfwidth,[0, 1],'b')
     plot([0, xpos(1,i)]-halfwidth,[0, 1],'b')
     for j = 1:N
         plot([xpos(j,i)+halfwidth, xpos(j,i)-halfwidth],[j, j],'b')
         plot([xpos(j,i)+halfwidth, xpos(j,i)-halfwidth],[j, j],'b')
     end

     axis equal
     xlim([-1 1])
     ylim([0 N])
 drawnow
 end


 %Create figure showing the inter-floor displacement from each floor to the
 %floor below it (in the case of Floor 1, it is the displacement of Floor 1
 %compared to the ground).
 figure (2)
 cla
 hold on
 displacement = x(1:N,:) - [zeros(1,Nt); x(1:N-1,:)];%part c
 for i = 1:N
     plot(t' ,displacement(i,:),'LineWidth',3)
 end
 newcolors = {'#F00','#F80','#FF0','#0B0','#00F','#FF00FF'};
 colororder(newcolors)

 % Added Code (Added Floor 6 -> Pendulum)
 yline(0.04,'k--','Threshold', 'LineWidth',2);
 yline(-0.04,'k--','Threshold','LineWidth',2);
 legend('Floor 1','Floor 2','Floor 3','Floor 4','Floor 5','Floor 6','Positive
Threshold','Negative Threshold');
```