```
%% Part 1
clc
clear
close all %clearing out to start the script
%Load the data file ENGR131_Lab4_Data.m into the workspace
load('ENGR131_Lab4_Data.mat')

for i = 1:length(w) %outer loop %will loop for as many characters in w
    T = linspace(0, 15/w(i), 100); %create a time vector
    for j = 1:length(DC) %inner loop
        H = CalcPosition(T, DC(1,j));
        PlottingFunction(i, j, T, H, w(1,i));
    end
end

%% Part 2
load ("ENGR131_Lab4_CatMap.mat") %load CatMap file (binary matrix)

CatMarks = CatMap;

%create a new figure
figure(1)
set(gcf,'color','#f80')
hold on
% T = time , H = height , w = frequency
% i =  # of subplots, j = 1=5 (line color)
Catmap()

function CatMap()
for x = 2:9
    for y = 2:9
        if CatMap(x,y) == 1
           CellsTouching = 0;
           %fprintf('x: %.1x ', x) %trying to see if it recognizes a 1
           %fprintf('y: %.1x \n', y)
           for k = [-1,1] % variable for -1 and 1 (to add to x and y)
                   if CatMap(x+k,y) == 1
                       fprintf('original cell: x: %.1x y: %.1x \n', x, y) %debug
                       fprintf('touching cell: x: %.1x y: %.1x \n', x+k, y)
                       CellsTouching = CellsTouching + 1;
                   end
                   if CatMap(x,y+k) == 1
                       fprintf('original cell: x: %.1x y: %.1x \n', x, y) %debug
                       fprintf('touching cell: x: %.1x y: %.1x \n', x, y+k)
                       CellsTouching = CellsTouching + 1;
                   end
                   switch CellsTouching
                   case 0
                   CatMarks(x,y) = 2;
                   case 1
                   CatMarks(x,y) = 2;
                   case 2
                   CatMarks(x,y) = 1;
           end
           end
        end
    end
end
end
```

```
%%
function PlottingFunction(i,j,T,H,w)
    DC = evalin('base','DC;'); %allows it to see DC
    switch j % to understand what switch is, right click then hit help
        case 1
            LineColor='b--';
        case 2
            LineColor='m-';
        case 3
            LineColor='k:';
        case 4
            LineColor='g-';
        case 5
            LineColor='c-.';
        otherwise
            LineColor='r;';
    end
    figure(1)
    subplot(2,3,i)
    hold on
    plot(T,H,LineColor)
    xlabel('Time (seconds)')
    ylabel('Height (meters)')
    title('Height as a function of Damping'); %semicolon does not print result of
the operation, suppresses the output from getting printed to the command window
    if i == 5
        legend(['DC1=',num2str(DC(1,1))],['DC2=',num2str(DC(1,2))],
['DC3=',num2str(DC(1,3))],['DC4=',num2str(DC(1,4))]);
    end
    axis([0 T(end) -1 1])
    hold off
end

function Height = CalcPosition(Time, Damping)
    % calculating Height based on the inputs time and damping coefficient
    a = 1; % defining variables for values of a and w
    w = 6;
    Height = exp(-Damping.*Time).*a.*cos(w.*Time); % calculate height
end
```