```
% ECE 213, C2
% Aidan Chin
% 5/21/24

%this code's purpose is the create a transfer function for the RLC circuit
%and find its impulse response using inverse laplace transform, and outputs
%its plot

% ------ initialization --------

clc % clear terminal
clf % clear all figures
clear % remove all variables from the workspace

% --------- User inputs ---------
R = input('Enter the resistance R (in ohms): ');
L = input('Enter the inductance L (in mH): ') * 1e-3; % convert mH to H
C = input('Enter the capacitance C (in uF): ') * 1e-6; % convert uF to F


% --------- Given ----------
a = R / (2*L); % damnping ratio
w0 = 1 / sqrt(L*C); % initial omega
%This RLC circuit can be written in the s domain by the impulse response
%H(s) = 1 / ((Ls^2 + Rs + 1/C)). With the user inputs R, L, and C, we can
%again rewrite the function in terms of alpha and omega_0 by the
%conversions a = R / 2L and W0 = 1 / sqrt(LC). Now, the transfer function
%can be written as (s^2 + 2as + W0^2). Since alpha and omega are linked to
%the user inputs, the roots will help determine the damping in h(t)

% Time vector
t0 = 0; %start time
tf = .12; %final time, 120ms
n = 1000; %number of steps in t
t = linspace(t0, tf, n + 1); % create array of time

% Determine the damping and compute h(t)
if a > w0 % Overdamped
    s1 = -a + sqrt(a^2 - w0^2);
    s2 = -a - sqrt(a^2 - w0^2);
    h1 = @(t) exp(s1*t); %positive root
    h2 = @(t) exp(s2*t); % negative root
    h = @(t) (s2*exp(s1*t) - s1*exp(s2*t)) / (s2 - s1); %sum of terms
    titleText = {'Aidan chin: 213 C2', ...
        sprintf('Overdamped: a > w_0, R = %g Ω, L = %g mH, C = %g µF', ...
        R, L*1e3, C*1e6)};
elseif a == w0  % Critically damped
    s1 = -a;
    h1 = @(t) exp(s1*t); %positive root
    h2 = @(t) t.*exp(s1*t); %negative root
    h = @(t) (1 + s1*t).*exp(s1*t); % sum of terms
    titleText = {'Aidan chin: 213 C2', ...
        sprintf(['Critically Damped: a = w_0, R = %g Ω, L = %g mH, ' ...
        'C = %g µF'], R, L*1e3, C*1e6)};
else % Underdamped
    omega = sqrt(w0^2 - a^2);
    h1 = @(t) cos(omega*t).*exp(-a*t); %positive root
    h2 = @(t) sin(omega*t).*exp(-a*t); % negative root
    h = @(t) exp(-a*t).*(cos(omega*t) + (a/omega)*sin(omega*t));
    % sum of terms
    titleText = {'Aidan chin: 213 C2', ...
        sprintf(['Underdamped: a < w_0, R = %g Ω, L = %g mH, ' ...
        'C = %g µF'], R, L*1e3, C*1e6)};
end

% Plotting
figure;
plot(t*1e3, h1(t), ':', 'DisplayName', 'h1(t)');
hold on;
plot(t*1e3, h2(t), ':', 'DisplayName', 'h2(t)');
plot(t*1e3, h(t), 'LineWidth', 2, 'DisplayName', 'h(t) (Sum)');
hold off;

% Formatting
title(titleText, 'FontSize',12);
xlabel('Time (ms)','FontSize',15);
ylabel('h(t)','FontSize',15);
legend('show','FontSize',15);
grid on;
```