```
% 19ucc023
% Mohit Akhouri
% Experiment 7 - Observation 3

% In this code , we take the input of audio file and divide the whole
% audio file into blocks of size 1x256 . Now we apply DCT on each
% individual block and Apply the COMPRESSION ALGORITHM

% COMPRESSION ALGORITHM : In this we decide the threshold ( minimum
% threshold and maximum threshold ) , we reject the coefficient values
% which are between the threshold values and accept the rest. This is
 how
% we achieve compression.

% Finally we plot the reconstructed wave and also write it to an audio
 file

clc;
clear all;
close all;

[x,fs] = audioread('input.wav'); % Reading of audio file 'input.wav'

% Plot of Original Sound wave 'input.wav'
figure;
plot(x);
xlabel('time(t) ->');
ylabel('x(t) ->');
title('19ucc023 - Mohit Akhouri','Plot of Original Sound wave
 input.wav');
grid on;

sound(x,fs); % To hear the original sound wave via speakers

N = size(x,1); % Row size of the input audio signal

index = ceil(N/256); % calculating the index for division of blocks

% We adjust the input audio signal 'x(t)' so as it is DIVISIBLE by 256

% We pad the extra spaces by zeros
x = [x ; zeros(256*index - N,1) ];

N = size(x,1); % Re-calculation of size after adjustment of audio
 signal x(t)

threshold1 = 0.09; % Maximum threshold
threshold2 = -0.09; % Minimum threshold

% Main loop algorithm for the compression of audio signal starts here
for i=1:256:N
```

```
    x_block = x(i:i+255); % Dividing input audio signal into 1x256
 blocks
    x_block_dct = myCompression(x_block); % Taking DCT of 1x256 block

    % Loop for the checking of coefficients
    % Those coefficients are rejected , which are between the
 threshold
    % values , rest are accepted
    for j=1:256
        if (x_block_dct(j) >= threshold2 && x_block_dct(j) <=
 threshold1)
            x_block_dct(j) = 0;
        end
    end

    x_block_idct = myDeCompression(x_block_dct); % Inverse DCT of the
 Compressed Audio block
    x(i:i+255) = x_block_idct; % Storing the IDCT into reconstructed
 wave variable x
end

% Plot of the reconstructed Compressed wave after applying threshold
 values
% for compression
figure;
plot(x);
xlabel('time(t) ->');
ylabel('x_{reconstructed}(t) ->');
title('19ucc023 - Mohit Akhouri','Plot of Reconstructed Sound Wave
 when - removing coefficients between -0.09 and 0.09');
grid on;

audiowrite('Obs3_outputcompression.wav',x,fs); % Writing Compressed
 audio signal to a file
sound(x,fs); % To hear the compressed audio signal from speakers
```

Published with MATLAB® R2020b