```
function [resultFeatures] = getFeatures(exampleID, plt)
    % Only plot the digits if plt == 1

    % Features:
    % 1 - Total energy
    % 2 - Standard deviation
    % 3 - Max amplitude
    % 4 - Zero crossing rate
    % 5 - Duration
    resultFeatures = zeros(10, 5);

    % Find the duration of the longest audio
    maxRows = 0;
    for i = 0:9
        [y, ~] = audioread(sprintf("samples/%d_16_%d.wav", i, exampleID));
        [rows, ~] = size(y);

        if(maxRows < rows)
            maxRows = rows;
        end
    end

    for i = 0:9
        % y is the audio signal
        % Fs is the sampling frequency
        [y, Fs] = audioread(sprintf("samples/%d_16_%d.wav", i, exampleID));

        % Store the standard deviation and the max amplitude before
normalizng the amplitude
        resultFeatures(i + 1, 2) = std(y);
        resultFeatures(i + 1, 3) = max(y);


        % Normalize the signal based on the maximum amplitude
        y = y / max(abs(y));

        % [rows, cols] are the original dimensions of y (there are #rows
samples)
        [rows, ~] = size(y);

        % Split the audio signal in small frames in order to extract certain
features
        % Frame size in seconds
        frameSize = 0.01;
        % Get number of samples per frame
        frameSamples = round(frameSize * Fs);
        % Calculate number of frames
        numFrames = floor(rows / frameSamples);

        % frameEnergy is an array with the energy value of every frame
```

```
        frameEnergy = getFrameEnergy(y, frameSamples, numFrames);

        energyThreshold = 0.5;

        % Find first index of the first frame with energy above threshold
        startFrame = find(frameEnergy > energyThreshold, 1);
        % Get the TIME index of the first frame with energy above threshold
        startSample = (startFrame - 1) * frameSamples + 1;

        % Find the index of the last frame with energy above threshold
        endFrame = find(frameEnergy > energyThreshold, 1, 'last');
        % Get the TIME index of the last frame with energy above threshold
        endSample = endFrame * frameSamples;

        % Store the audio duration without silences in seconds
        resultFeatures(i + 1, 5) = (endSample - startSample) / Fs;

        % Trim y in order to remove low energy values
        y = y(startSample:end);
        % Add silence to the end of y
        [curRows, ~] = size(y);
        concatY = zeros(maxRows - curRows, 1);
        y = [y; concatY];

        % Store total energy and pitch
        resultFeatures(i + 1, 1) = sum(abs(y) .^ 2);
        resultFeatures(i + 1, 4) = zerocrossrate(y);

        if plt == 1
            % Ts is the sampling period
            Ts = 1 / Fs;
            % t is the time vector
            t = (0:maxRows-1);
            % t is the time vector in seconds (starts at the first frame
that exceeds the energy threshold)
            t = (t .* Ts);
            subplot(5, 2, i + 1);
            plot(t, y');
            xlabel('Time (s)');
            ylabel('Amplitude');
            label = sprintf("%d", i);
            title(label);
        end
    end
end
```