```
function optimal_stepsize = optimise_step_mse(pyramid,h,X,ratios)
%% This function finds the optimal stepsize for mse
% Input:  pyramid is a cells object containing the unquantised X_lists and
% Y_lists
%         ratios is an array containing the mse relative step sizes

% Output: optimal_stepsize is an array of length n+1 where n is the size of the
 %        pyramid. each number in the array indicates the optimal stepsize
 %        for that layer


% Author: Andy Cai CRSID ajc327
% Date : 12/05/2020

    errors = [];
    x = linspace(5,30,26);

    for i = 5: 30
        quant_steps = round(i*ratios);
        quantised_pyramidi = quantpyramid(pyramid, quant_steps);
        decodedi= pyndec(quantised_pyramidi,h);
        standard_devi = std(X(:)-decodedi(:));
        errors = [errors, standard_devi];
    end
    X_quantised = quantise(X,17);
    orig_error = std(X(:)-X_quantised(:));

    closeness = abs(errors - orig_error);
    [my_min, my_index] = min(closeness);
    optimal_stepsize = round(x(my_index)*ratios);
    optimal_stepsize = optimal_stepsize(1:length(pyramid{2})+1);



%     plot(x, errors);
%     p = polyfit(x, errors, 1);
%     x1 = linspace(5,30,200);
%     p1 = polyfit(errors, x,1);
%
%     my_fit = polyval(p, x1);
%     hold on;
%     optimal_stepsize = polyval(p1,orig_error);
%     plot(optimal_stepsize,orig_error,'o');
%
%
%     plot(x1, my_fit);
%     my_text = strcat('\leftarrow optimal step size at ', string(optimal_stepsize));
%     text(optimal_stepsize,orig_error, '\leftarrow optimal step size');
%     hold off;

    return
```

```
end
```