function [lambda\_vec, error\_train, error\_val] = ...

validationCurve(X, y, Xval, yval)

%VALIDATIONCURVE Generate the train and validation errors needed to

%plot a validation curve that we can use to select lambda

% [lambda\_vec, error\_train, error\_val] = ...

% VALIDATIONCURVE(X, y, Xval, yval) returns the train

% and validation errors (in error\_train, error\_val)

% for different values of lambda. You are given the training set (X,

% y) and validation set (Xval, yval).

%

% Selected values of lambda (you should not change this)

lambda\_vec = [0 0.001 0.003 0.01 0.03 0.1 0.3 1 3 10]';

% You need to return these variables correctly.

error\_train = zeros(length(lambda\_vec), 1);

error\_val = zeros(length(lambda\_vec), 1);

% ====================== YOUR CODE HERE ======================

% Instructions: Fill in this function to return training errors in

% error\_train and the validation errors in error\_val. The

% vector lambda\_vec contains the different lambda parameters

% to use for each calculation of the errors, i.e,

% error\_train(i), and error\_val(i) should give

% you the errors obtained after training with

% lambda = lambda\_vec(i)

%

% Note: You can loop over lambda\_vec with the following:

%

% for i = 1:length(lambda\_vec)

% lambda = lambda\_vec(i);

% % Compute train / val errors when training linear

% % regression with regularization parameter lambda

% % You should store the result in error\_train(i)

% % and error\_val(i)

% ....

%

% end

%

%

% =========================================================================

end