```
"""
A simple wrapper for linear regression.  (c) 2015 Tucker Balch
Note, this is NOT a correct DTLearner; Replace with your own implementation.
"""

import numpy as np
import warnings

class DTLearner(object):

    def __init__(self, leaf_size=1, verbose = False):
        warnings.warn("\n\n  WARNING! THIS IS NOT A CORRECT DTLearner
IMPLEMENTATION! REPLACE WITH YOUR OWN CODE\n")
        pass # move along, these aren't the drones you're looking for

    def author(self):
        return 'tb34' # replace tb34 with your Georgia Tech username

    def addEvidence(self,dataX,dataY):
        """
        @summary: Add training data to learner
        @param dataX: X values of data to add
        @param dataY: the Y training values
        """

        # slap on 1s column so linear regression finds a constant term
        newdataX = np.ones([dataX.shape[0],dataX.shape[1]+1])
        newdataX[:,0:dataX.shape[1]]=dataX

        # build and save the model
        self.model_coefs, residuals, rank, s = np.linalg.lstsq(newdataX, dataY)

    def query(self,points):
        """
        @summary: Estimate a set of test points given the model we built.
        @param points: should be a numpy array with each row corresponding to a
specific query.
        @returns the estimated values according to the saved model.
        """
        return (self.model_coefs[:-1] * points).sum(axis = 1) + self.model_coefs[-
1]

if __name__=="__main__":
    print "the secret clue is 'zzyzx'"
```