```
import numpy as np

indata = np.loadtxt(r"C:\Users\Zhenc_000\Documents\Caltech\CS156\Set6\indta.txt")
outdata = np.loadtxt(r"C:\Users\Zhenc_000\Documents\Caltech\CS156\Set6\outdta.txt")

def linreg(x,y): #function to perform linear regression
    return np.dot(np.linalg.pinv(x),y)

def regreg(x,y,l): #function to perform linear regression with regulation
    return np.dot(np.dot(np.linalg.inv(np.dot(np.transpose(x),x) +
    l*np.identity(8)),np.transpose(x)),y)

def error(w,z,y): #find the in-sample or out of sample error
    g = np.sign(np.dot(z,np.transpose(w)))
    return ((len(y)-np.sum(np.dot(y,g)))/2)/len(y)

in_x1 = indata[:,0]
in_x2 = indata[:,1]
in_y = indata[:,2]
out_x1 = outdata[:,0]
out_x2 = outdata[:,1]
out_y = outdata[:,2]

in_z = np.zeros([len(in_x1),8]) #transform x to z
in_z[:,0] = 1
in_z[:,1] = in_x1
in_z[:,2] = in_x2
in_z[:,3] = np.multiply(in_x1,in_x1)
in_z[:,4] = np.multiply(in_x2,in_x2)
in_z[:,5] = np.multiply(in_x1,in_x2)
in_z[:,6] = np.absolute(in_x1-in_x2)
in_z[:,7] = np.absolute(in_x1+in_x2)

out_z = np.zeros([len(out_x1),8]) #transform x to z
out_z[:,0] = 1
out_z[:,1] = out_x1
out_z[:,2] = out_x2
out_z[:,3] = np.multiply(out_x1,out_x1)
out_z[:,4] = np.multiply(out_x2,out_x2)
out_z[:,5] = np.multiply(out_x1,out_x2)
out_z[:,6] = np.absolute(out_x1-out_x2)
out_z[:,7] = np.absolute(out_x1+out_x2)

w_lin = linreg(in_z,in_y) #normal linear regression
w_reg = regreg(in_z,in_y,10**-3) #linear regression with regulation k = 10^-3
w_reg2 = regreg(in_z,in_y,10**3) #linear regression with k = 10^3
w_reg3 = regreg(in_z,in_y,10**-1) #linear regression with k = 10^-1

print(error(w_lin,in_z,in_y),error(w_lin,out_z,out_y)) #problem 2
print(error(w_reg,in_z,in_y),error(w_reg,out_z,out_y)) #problem 3
print(error(w_reg2,in_z,in_y),error(w_reg2,out_z,out_y)) #problem 4
print(error(w_reg3,in_z,in_y),error(w_reg3,out_z,out_y)) #problem 5 & 6
```