```
# for question 12

import numpy as np
from sklearn import svm

def transform(matx):
    matz = np.zeros(shape=(len(matx), 2))
    for i in range(len(matx)):
        x1 = matx[i, 0]
        x2 = matx[i, 1]
        z1 = x2 ** 2 - 2 * x1 - 1
        z2 = x1 ** 2 - 2 * x2 + 1
        matz[i] = [z1, z2]
    return matz

matx = np.matrix([[1, 0], [0, 1], [0, -1], [-1, 0], [0, 2], [0, -2], [-2, 0]])
vecty = np.matrix([[-1], [-1], [-1], [1], [1], [1], [1]])
matz = transform(matx)

clf = svm.SVC(kernel='poly', degree=2, C=10000000)
clf.fit(matz, vecty)
print(clf.dual_coef_)
```