```
#for questions 7-10

import numpy as np
import random

def createData(f, n):
    dataset = []
    for i in range(0, n):
        x = (random.uniform(-1, 1), random.uniform(-1, 1))
        y = 1 if (x[1] > f(x[0])) else -1
        dataset.append((x, y))

    return dataset

def setup(n):
    #Set up the target function f
    x = random.uniform(-1, 1), random.uniform(-1, 1)
    y = random.uniform(-1, 1), random.uniform(-1, 1)
    z = np.polyfit(x, y, 1)
    f = np.poly1d(z)

    #create data
    dataset = createData(f, n)

    return (f, dataset)

def pla(f, dataset):
    w = np.array([0, 0, 0])
    misclassified = dataset
    iterCount = 0
    while misclassified:
        iterCount += 1
        misclassified = []
        for point in dataset:
            x = np.array([1, point[0][0], point[0][1]])
            y = point[1]
            if np.sign(y) != np.sign(np.inner(w, x)):
                misclassified.append((x, y))
        if misclassified:
            p = random.choice(misclassified)
            w = w + p[1] * p[0]

    #calculate disagreement
    testdata = createData(f, 1000)
    disagreeCount = 0
    for point in testdata:
        x = np.array([1, point[0][0], point[0][1]])
        y = point[1]
        if np.sign(y) != np.sign(np.inner(w, x)):
            disagreeCount += 1
    return (iterCount, disagreeCount / 1000)

def runpla(runCount, n):
    totalIterCount = 0
    totalDisagreement = 0
    for i in range(0, runCount):
        f, dataset = setup(n)
        iterCount, disagreement = pla(f, dataset)
        totalIterCount += iterCount
        totalDisagreement += disagreement
    return (totalIterCount / runCount, totalDisagreement / runCount)

#print(runpla(1000, 10)) for answers to #7 and #8
#for #9-10:
print(runpla(1000, 100))
```