```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct  5 21:28:19 2016

Authors: Andrew Alferman and Nathan Schorn

This is the Python code used to obtain the answer to problem 3.c)
"""

#Importing Commands
import numpy as np
import matplotlib.pyplot as plt

#Time Variables
timescale = 0.001
runtime = 250.0
#The variables below are modified as part of the problem.
p1_1=0.001
p2_1=0.1
p1_2=0.01
p2_2=1.0

#Creating a function to plot a trajectory
def trajectory(p1,p2,timescale,runtime,case):
    #Initialize the variables and plotting vectors
    x=0.0
    y=0.0
    time = 0.0
    theta=np.radians(45.0)
    v_x=np.cos(theta)
    v_y=np.sin(theta)
    y_max = 0.0
    a=[]
    b=[]
    #Given Information
    d=0.1
    m=1.0
    rho=10.0
    #Find the coefficient of drag
    c_d=(8.0*p1*m)/(np.pi*rho*d**3)
    #Computation of the trajectory is accomplished in a single while loop
    while time<=runtime:
        #Append the plotting vectors with the current x and y coordinates
        a.append(x)
        b.append(y)
        #Move the ball based on the velocity of the previous timestep
        x+=v_x*timescale
        y+=v_y*timescale
        #Calculate the velocity and angle computed in the previous timest
        v=np.sqrt(v_x**2+v_y**2)
        theta=np.arctan(v_y/v_x)
        #Accelerate the ball using the formula calculated in step 3.c)
```

```
            v_x+=-1*p1*v**2*np.cos(theta)*timescale
            v_y+=-1*(p2+(p1*v**2*np.sin(theta)))*timescale
            #Advance the time one timescale
            time+=timescale
            #Determine the maximum height and output stats of the trajectory
            if y>=y_max:
                y_max=y
            elif y <= 0:
                print("Ball {} impacts the ground under the following conditi
                      .format(case))
                print(" Time: {}.".format(time))
                print(" Distance: {}.".format(x))
                print(" Speed of impact: {}.".format(v))
                print(" Maximum height: {}.".format(y_max))
                print(" Coefficient of drag: {}.".format(c_d))
                print("----------")
                return [a,b]
                break

#Create all of the trajectory data
case1=trajectory(p1_1,p2_1,timescale,runtime,"1")
case2=trajectory(p1_2,p2_1,timescale,runtime,"2")
case3=trajectory(p1_1,p2_2,timescale,runtime,"3")
case4=trajectory(p1_2,p2_2,timescale,runtime,"4")
#Plot out the result
plt.plot(case1[0],case1[1])
plt.plot(case2[0],case2[1])
plt.plot(case3[0],case3[1])
plt.plot(case4[0],case4[1])
plt.xlabel("Distance")
plt.ylabel("Height")
plt.show()
```