```python
#!/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.d)
"""

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

#Time Variables
timescale = 0.01
runtime = 5000.0

#Given Information
d=0.1
m=1.0
g=9.8
rho=10.0

#The variables below are modified as part of the problem.
p1_1=0.001
p1_2=0.01
p1_v=[p1_1,p1_2]

#Creating a function to plot a trajectory
def trajectory(p1,timescale,runtime,v_0,g,d):
    #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
    v_0i=1/v_0
    a=[]
    b=[]
    #Computation of the trajectory is accomplished in a single while loo
    while time<=runtime:
        #Append the plotting vectors with the current x and y coordinate
        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 times
        theta=np.arctan(v_y/v_x)
        v=np.sqrt(v_x**2+v_y**2)
```

```
        #Accelerate the ball using the formula calculated in step 3.c)
        v_x+=-1*p1*v**2*timescale
        v_y+=-1*(g*(v_0i**2)+p1*v**2)*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:
            return x
            break

#Find the coefficient of drag
for l in p1_v:
    c_d=(8.0*l*m)/(np.pi*rho*d**3)
    print ("Coefficient of drag for P1 = {}: {}".format(l,c_d))

#Create all of the trajectory data
x_s=[]
v_v=[]
for i in p1_v:
    for j in range(5,26):
        x_s.append(trajectory(i,timescale,runtime,j,g,d))
        v_v.append(j)
    plt.plot(v_v,x_s)
    x_s=[]
    v_v=[]


#Plot out the result
plt.xlabel("Initial Velocity")
plt.ylabel("Distance")
plt.show()
```