```
#Problem 7.2
#-------------------------Comments------------------------
"""
In this task we assume that since Beagles mass is so small
ralative to Mars, that Mars' velocity and position after
the initial values are negligible.
We are only looking at Beagles position and velocity,
using the velocity to calculate the position for the next
iteration.
To simplify the program i vectorized everything into the
vector u, witch contains both position and velocity,
iterating over it in the while loop, calculating the force
until the distance from origo is less than the radius of Mars,
which means that it has landed (or crashed).
"""
#-------------------------Imports---------------------------

from scitools.std import *
from numpy.linalg import norm

#-------------------------Defining variables-----------------

G = 6.67*10**-11          #Gravitational constant
k = 0.00016               #Friction constant
radius = 3400000          #Radius of Mars in meters
Mars_mass = 6.4*10**23    #[kg]
Beagle_mass = 100.0       #[kg]

u0 = asarray([[-298000 - radius, 0.0], [0.0, -4000]])
                          #position (x,y), Velocity (x,y)

u = zeros((70000, 2, 2)) #Empty array for values
u[0] = u0                 #Adding initial values to u
n = 70000
i=0
land = 'no'               #Beagle has not landed

#-------------------------Force function---------------------

def Force(u):             #Both forces (Athmospheric friction and Gravitation)
 r = u[0]
 v = u[1]
 a = -k*v/Beagle_mass - G*Mars_mass*r/(norm(r)**3)
 return asarray([v, a])

#--------------------------While loop-----------------------

while land=='no' and i<(n-1):
    h = 100               # Calculating position and velocity (MidEuler)
    u[i+1] = u[i] + h*Force(u[i] + h/2*Force(u[i])) # Timestep not needed
    r = u[i+1]            #Distance from (0,0)
    if norm(r) < radius: #As explained; if it kicks in, we have landed
        land = 'yes'
        break
    i = i+1
```

```
x = u[:,0][:,0]
y = u[:,0][:,1]

#-------------------------Plot command----------------------
t=linspace(0,2*pi, 70000) # Making an array for my simple Mars plot.
marsx=array(radius*cos(t))
marsy=array(radius*sin(t))


plot(x,y, '-b', marsx,marsy,'r',axis='equal') #Mars is the red circle (Not an
 orbit)
savefig('PlotOblig1.png')

"""
Problem 7.3
It looks as if Beagle lands on mars close to where it started after orbiting
mars 2 times.
"""
```