```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Mar 18 15:14:07 2020

COIS 2310H Assignment 3 Question 4 (Module)

@author: davidstothers
"""

import numpy as np


global g
g=9.81

def pendulum_EC(time,mass=1.0,Length=1.0,theta0=1.0,omega0=0.0,forceq=0.0,drivef
    """
    Uses the Euler-Cromer method to approximate the motion of a pendulum.

    time: numpy array of time locations.
    mass: mass of the pendulum.
    Length: length of the pendulum.
    theta0: the initial angle of the pendulum.
    omega0: the initial speed of the pendulum.
    forceq: the constant determining the damping force.
    drivefreq: the frequency of the driving force.
    driveamp: the amplitude of the driving force.

    Returns arrays for position and speed.
    """
    timelength=np.size(time)

    theta=np.arange(0,timelength,dtype=float)
    theta[0]=theta0
    omega=np.arange(0,timelength,dtype=float)
    omega[0]=omega0

    dt=np.arange(0,timelength,dtype=float)

    for i in np.arange(0,timelength-1):
        dt[i] = time[i+1]-time[i]
        omega[i+1]=omega[i]+(Fnet(mass,Length,theta[i],omega[i],forceq,drivefreq
        theta[i+1]=theta[i]+omega[i+1]*dt[i]

        while(theta[i+1]>=np.pi):
            theta[i+1]-=2.0*np.pi
        while(theta[i+1]<-1.0*np.pi):
            theta[i+1]+=2.0*np.pi
```

```python
    return theta,omega

def Fnet(m,L,theta,omega,q,dfreq,damp,t):
    Fdamp = -1.0*m*q*omega
    Fg = -1.0*np.sin(1.0*theta)*m*g/L
    Fdrive = damp*np.sin(dfreq*t)

    Fnet = 1.0*(Fg+Fdamp+Fdrive)
    return Fnet

def lyapunov(time,theta1,theta2):
    dtheta=np.abs(theta2-theta1)
    ly = np.log(dtheta)*1.0/time

    return ly
```