```
#!/usr/bin/python3

import numpy as np
import matplotlib.pyplot as plt

fig, (ax1, ax2) = plt.subplots(2)

# PROBLEM 1
a = 2
w = np.pi

def F(x, N):
    return (a / (np.pi * w) * sum([((
        np.sin(np.pi * n) - np.pi * n * np.cos(np.pi * n)) / np.power(n, 2.0)
                                    ) * np.sin(n * w * x) for n in range(1, N+1)]))
def Real(x):
    while x < -1:
        x += 2
    while x > 1:
        x -= 2
    return a*w/(2*np.pi)*x

X = np.linspace(-2, 2, 500)
ax1.plot(X, [Real(x) for x in X], 'k', label="Analytical")
ax1.plot(X, [F(x, 1) for x in X], 'k--', label="N=1")
ax1.plot(X, [F(x, 5) for x in X], 'k:', label="N=5")
ax1.plot(X, [F(x, 20) for x in X], 'k-.', label="N=20")
ax1.legend()

# PROBLEM 2
def Force(t):
    return 0.1*np.cos(0.8*t)+0.1*np.sin(0.8*t)

def euler_cromer(x, v, a, dt, ti, tf):
    X = [x]
    V = [v]
    A = [a]
    for t in np.arange(ti, tf, dt):
        V.append(V[-1] + A[-1] * dt)
        X.append(X[-1] + V[-1] * dt)
        A.append(-X[-1]-0.5*V[-1]+Force(t))
    return X[:-1], V[:-1], A[:-1]

def diff_eq(t):
    return 0.185823*(np.cos(0.8*t-0.8379)+np.sin(0.8*t-0.8379))

T = np.arange(0, 30, 0.1)
eX, eV, eA = euler_cromer(1, 0, 0, 0.1, 0, 30)
ax2.plot(T, eX, 'k-', label="Euler Cromer")
ax2.plot(T, [diff_eq(t) for t in T], 'k--', label="Superposition")
ax2.legend()

fig.set_size_inches(6,9)
plt.savefig("hw3.ps", papertype="a4")
# plt.show()
# plt.show()
```