```python
# -*- coding: utf-8 -*-

"""
Christian DiPietrantonio
ME 5110: Advanced Thermodynamics
Homework 04
09/16/2025

"""

import matplotlib.pyplot as plt
import numpy as np


# -------------------------
#PROBLEM 01
# -------------------------

# -------------------------
# INFINITE WELL
# -------------------------

#CONSTANTS AND CONVERSION FACTORS
h = 6.626*10**-34               #Planck's constant (Units: J-s)
h_bar = h/(2*np.pi)             #Reduced Planck constant (Units: J-s)
m = 9.1093897*10**-31           #Mass of an electron from "Molecular Thermodynamics" (Units: kg)
eV = 1.602177*10**-19           #Joule to eV conversion factor from "Molecular Thermodynamics"


#WELL DEFINITION
L = 10**-9                      #Length of well (Units: m)
n_values = [1, 2, 3]            #List of Quantum Numbers

x = np.linspace(0, L, 100000)     #x values from 0 to L

#WAVE FUNCTIONS
plt.figure("Wave Functions for an Infinite Well", figsize=(8,5))

#Plot Wave Function for each Energy Level
for n in n_values:
    psi     = np.sqrt(2/L)*np.sin((n*np.pi*x)/L)
    plt.plot(x*10**9, psi, label = fr'$\psi(x), n={n}$')


plt.axvline(0, color='k', linestyle=':')          #Plot vertical line at x = 0
plt.axvline(L*10**9, color='k', linestyle=':')    #Plot vertical line at x = L
plt.title('Infinite 1D Well: Wavefunctions')
plt.xlabel('x (nm)')
plt.ylabel(r'$\psi(x)$')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.tight_layout()
plt.show()


#PROBABILITY DENSITIES
plt.figure("Probability Densities for an Infinite Well", figsize=(8,5))
```

```
#Plot Probability Density for each Enegy Level (and shade under the curve)
for n in n_values:
    psi     = np.sqrt(2/L)*np.sin((n*np.pi*x)/L)
    prob    = psi**2
    plt.plot(x*10**9, prob, label = fr'$|\psi(x)|^2, n={n}$')
    plt.fill_between(x*10**9, 0, prob, alpha=0.2)


plt.axvline(0, color='k', linestyle=':')          #Plot vertical line at x = 0
plt.axvline(L*10**9, color='k', linestyle=':')    #Plot vertical line at x = L
plt.title('Infinite 1D Well: Probability Densities')
plt.xlabel('x (nm)')
plt.ylabel(r'$|\psi(x)|^2$')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.tight_layout()
plt.show()


#ENERGY LEVELS
#Create list of Energy levels
E_list = []
for n in n_values:
    E_n = ((n**2)*(h**2))/(8*m*L**2)
    E_list.append(E_n/eV)

E = np.array(E_list)

plt.figure("Energy Levels for an Infinite Well", figsize=(8,5))

#Plot horizontal line for each Energy Level
for n in range(len(n_values)):
    plt.hlines(E[n], 0, L*10**9, colors=f'C{n}', label=fr'$E_{n_values[n]}, n={n_values[n]}$')


plt.xlabel('x (nm)')
plt.ylabel('Energy (eV)')
plt.title('Infinite 1D Well: Energy Levels')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.tight_layout()
plt.show()


#MOMENTUM VALUES
#Create list of Momentum Values
p_list = []
for n in n_values:
    p_n = (n*h)/(2*L)
    p_list.append(p_n)

p = np.array(p_list)

plt.figure("Momentum Values for an Infinite Well", figsize=(8,5))

#Plot horizontal line for each Momentum Value
for n in range(len(n_values)):
    plt.hlines(p[n], 0, L*10**9, colors=f'C{n}', label=fr'$p_{n_values[n]}, n={n_values[n]}$')


plt.xlabel('x (nm)')
plt.ylabel(r'Momentum $(\frac{kg \cdot m}{s})$')
plt.title('Infinite 1D Well: Momentum Values')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.tight_layout()
plt.show()
```