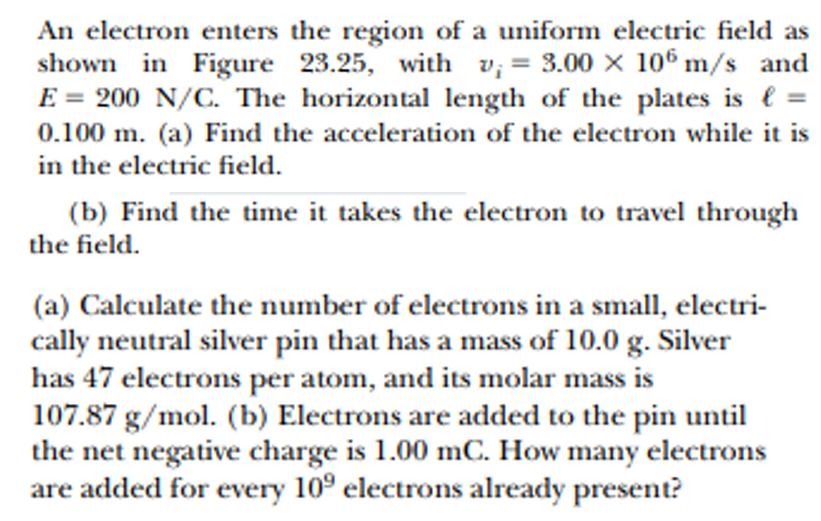
***NAME:YASH RAJ***

***ROLL NUMBER:24K-0737***

***QUESTION#1:*** ******

***PYTHON CODE:***

***q = 1.602e-19***

***m = 9.109e-31***

***E = 200***

***v\_i = 3.00e6***

***length = 0.100***

***F = q \* E***

***a = F / m***

***print(f"Acceleration of the electron: {a} m/s^2")***

***t = length / v\_i***

***print(f"Time to travel through the field: {t} seconds")***

***mass\_silver = 10.0***

***molar\_mass\_silver = 107.87***

***N\_A = 6.022e23***

***n\_moles = mass\_silver / molar\_mass\_silver***

***n\_atoms = n\_moles \* N\_A***

***n\_electrons\_pin = n\_atoms \* 47***

***print(f"Number of electrons in the silver pin: {n\_electrons\_pin}")***

***Q = 1.00e-3***

***n\_added\_electrons = Q / q***

***print(f"Number of added electrons: {n\_added\_electrons}")***

***additional\_electrons\_ratio = n\_added\_electrons / 1e9***

***print(f"Added electrons for every 10^9 electrons: {additional\_electrons\_ratio}")***

***OUTPUT:***

***Acceleration of the electron: 35174003732572.184 m/s^2***

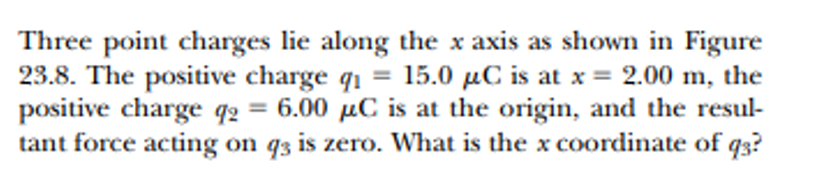
***Time to travel through the field: 3.3333333333333334e-08 seconds***

***Number of electrons in the silver pin: 2.623843515342542e+24***

***Number of added electrons: 6242197253433209.0***

***Added electrons for every 10^9 electrons: 6242197.253433209***

***QUESTION#2:***

******

***PYTHON CODE:***

***import sympy as sp***

***q1 = 15.0e-6***

***q2 = 6.0e-6***

***x1 = 2.0***

***k = 8.99e9***

***x = sp.Symbol('x')***

***F13 = k \* q1 / (x1 - x)\*\*2***

***F23 = k \* q2 / x\*\*2***

***equation = sp.Eq(F13, F23)***

***solution = sp.solve(equation, x)***

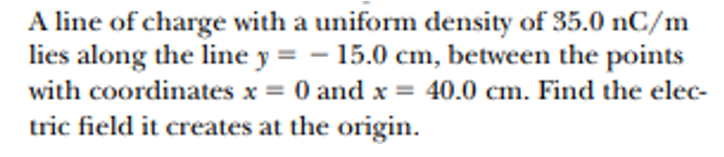
***x3 = [sol.evalf() for sol in solution]***

***print(f"x coordinate of q3: {x3}")***

***OUTPUT:***

***x coordinate of q3: [0.0, 3.0]***

***QUESTION#3:***

******

***PYTHON CODE:***

***import numpy as np***

***k = 8.99e9***

***lambda\_charge = 35.0e-9***

***y = -0.15***

***x\_start = 0.0***

***x\_end = 0.4***

***def electric\_field\_x(x):***

***r = np.sqrt(x\*\*2 + y\*\*2)***

***dE = k \* lambda\_charge / r\*\*2***

***dE\_x = dE \* (x / r)***

***return dE\_x***

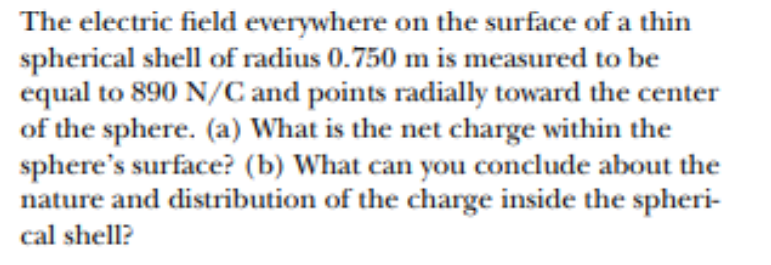
***E\_x = np.trapz([electric\_field\_x(x) for x in np.linspace(x\_start, x\_end, 1000)], np.linspace(x\_start, x\_end, 1000))***

***print(f"The electric field at the origin due to the line of charge is {E\_x:.2e} N/C in the positive x-direction.")***

***OUTPUT:***

***The electric field at the origin due to the line of charge is 1.36e+03 N/C in the positive x-direction***

***QUESTION#4:***

******

***PYTHON CODE:***

***# Given values***

***E = 890 # Electric field in N/C***

***r = 0.750 # Radius in meters***

***k = 8.99e9 # Coulomb's constant in N\*m^2/C^2***

***# Calculate the net charge Q***

***Q = (E \* r\*\*2) / k***

***# Output the result***

***print(f"The net charge within the sphere's surface is {Q:.2e} C.")***

***# Part (b) analysis***

***if E < 0:***

***charge\_nature = "negative"***

***else:***

***charge\_nature = "positive"***

***print(f"The electric field points radially toward the center, indicating the charge is {charge\_nature}. The charge is uniformly distributed over the shell.")***

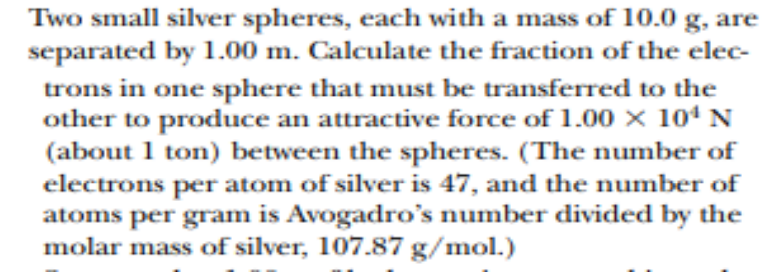
***OUTPUT:***

***The net charge within the sphere’s surface is 5.7e-08 C.***

***The electric field points radially toward the center, indicating the charge is positive.***

***The charge is uniformly distributed over the shell.***

***QUESTION#5:***

******

***PYTHON CODE:***

***import scipy.constants as const***

***import sympy as sp***

***mass\_sphere = 10.0e-3***

***distance = 1.0***

***force = 1.00e4***

***electrons\_per\_atom = 47***

***molar\_mass\_silver = 107.87***

***avogadro\_number = const.Avogadro***

***atoms\_per\_gram = avogadro\_number / molar\_mass\_silver***

***atoms\_in\_sphere = atoms\_per\_gram \* (mass\_sphere \* 1000)***

***total\_electrons = electrons\_per\_atom \* atoms\_in\_sphere***

***k = const.k\_e***

***q = sp.Symbol('q')***

***equation = sp.Eq(k \* (q \* q) / (distance \*\* 2), force)***

***solution = sp.solve(equation, q)***

***charge\_needed = abs(solution[0])***

***electron\_charge = const.e***

***fraction\_electrons\_transferred = charge\_needed / (total\_electrons \* electron\_charge)***

***print(f"Fraction of electrons transferred: {fraction\_electrons\_transferred:.2e}")***

***OUTPUT:***

***Fraction of electrons transferred: 2.50e-13.***