```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math

def main():
    tonzi = pd.read_csv('TonziSample.csv')
    vaira = pd.read_csv('VairaSample.csv')

    #Problem 1A: Plot Rn
    plt.plot(tonzi['TimeOfDay'], tonzi['Rn'], '-o', label =
'Tonzi Ranch')
    plt.plot(vaira['TimeOfDay'], vaira['Rn'], '-o', label =
'Vaira Ranch')
    plt.legend()
    plt.xlabel('Time of day')
    plt.ylabel('Net Radiation, $R_n$ ($W/m^2$)')
    plt.savefig('problem1a_rn.png')

    # Problem 2
    print('\n\nPROBLEM 2\n')
    Rn = 280 #W/m^2
    rho_water = 1000 #kg/m3
    rho_air = 1.2 #kg/m3
    T_air = 20 #c
    u = 1 # m/s
    RH = 65 #%
    h = 2 #m, height of met station
    gamma = 0.67 #mb C (psychrometric constant)
    g = 25 #W/m2 (ground heat flux)

    # Problem 2a: Calculate VPD

    def clausius_clap(T_air):
        """
        Returns e_sat given T_air in C.
        """
        e_star = 6.1094*math.exp((17.625*T_air)/(T_air +
243.04))
        return e_star

    def calc_vpd(T_air, RH):
        """
        Returns VPD in units of hPa
        given T_air in C and RH in decimal or percent.
        """
        # Convert to percent if given whole number
```

```
        if RH % 1 == 0: RH /= 100
        e_star = clausius_clap(T_air)
        return e_star * (1-RH)

    vpd = calc_vpd(T_air, RH)
    print(f"Problem 2a: VPD (hPa) = {round(vpd,2)}")


    # Problem 2b: Calculate aerodynamic resistance for each
plant
    gs_1 = 30
    gs_2 = 60
    h_1 = .30 #m
    h_2 = 1 #m

    def calc_aero_resistance(h, u):
        # h in m, u in m/s
        log = (2 - 0.7*h) / (0.1*h)
        denom = 0.41 * 0.41 * u
        return math.log2(log) / denom

    ra_1 = calc_aero_resistance(h_1, u)
    ra_2 = calc_aero_resistance(h_2, u)

    print(f"Ra (s/m) Option 1: {ra_1}")
    print(f"Ra (s/m) Option 2: {ra_2}")

    # Problem 2c: Calculate ET

    def calc_s(T_air):
        # Slope of Clausius-Clapyeron
        num = 17.625 * (T_air + 243.04) - 17.625 * T_air
        denom = (T_air + 243.04)**2
        return num/denom

    def calc_penman_monteith(Rn, G, T_air, h, u, RH, gamma,
rho_air, rho_water, r_s):
        r_a = calc_aero_resistance(h, u)
        vpd = calc_vpd(T_air, RH)
        s = calc_s(T_air)
        num = s * (Rn-G) + ((rho_air * 1005) / r_a)*vpd
        denom = s + gamma * (1 + (r_s/r_a))
        latent_heat = num/denom
        et = latent_heat / 2.45
        return et
```

```
    et_1 = calc_penman_monteith(Rn, g, T_air, h_1, u, RH, gamma,
rho_air, rho_water, gs_1)
    et_2 = calc_penman_monteith(Rn, g, T_air, h_2, u, RH, gamma,
rho_air, rho_water, gs_2)

    print(et_1)
    print(et_2)
    lam = 2.54e6
    sec_per_day = 86400

    print(f"Daily ET, option 1: {et_1 / lam * sec_per_day} mm/
day")
    print(f"Daily ET, option 2: {et_2 / lam * sec_per_day} mm/
day")


    ## Problem 4
    print('\n\nPROBLEM 4\n')
    t_inside = 20.6 #c
    rh_inside = 0.75 #%
    t_outside = -6.7 #c
    rh_outside = 0.8 #%
    rho_air = 1.2 #kg
    pressure = 1000 #mb

    # A) RH inside hosptal if air is brought
    # from outside and heated to temp,
    # but not humidified

    # RH = e/e*
    e_star_outside = clausius_clap(t_outside)
    e_outside = rh_outside * e_star_outside
    e_star_inside = clausius_clap(t_inside)
    print(f"E_star {e_star_inside, e_star_outside}")
    rh_inside = e_outside / e_star_inside
    print(f"RH in hospital if air not humidified:
{round(rh_inside*100)}%")

    # B) Hospital with 1500 m3 volume.
    # Humidifier vaporizes 4 liters/hr water.
    # How many hours should it be running to increase RH to 75%

    e_target = 0.75 * e_star_inside
    e_actual = rh_inside * e_star_inside
    print(f"Target vapor pressure (e) for 75% RH:
{round(e_target,2)} hPa")
    print(f"Unhumidified e: {round(e_actual,2)} hPa")
```

```
    print(f"Difference in vapor pressure: {round(e_target -
e_actual,2)} hPa")

    def mass_of_water(e):
        mass_air = 1.2 # kg
        P = 1000 # hPa
        mass_water = (e/P) / mass_air
        return mass_water

    water_target = mass_of_water(e_target) * 1500
    water_actual = mass_of_water(e_actual) * 1500
    print(water_target, water_actual)

    water_needed = (mass_of_water(e_target) -
mass_of_water(e_actual)) * 1500
    print(f"Litres of water needed: {round(water_needed)}")
    print(f"Hours of humidifier operation: {water_needed/4}")


if __name__ == '__main__':
    main()
```