```python
# -*- coding: utf-8 -*-
"""
Created on Wed May 21 15:00:59 2025

@author: ugims
"""


def calcular_todos_hq():
    import numpy as np
    from scipy.interpolate import interp1d

    # Table Data
    T_values = [100, 150, 200, 250, 300, 350, 400, 450, 500, 550,
                600, 650, 700, 750, 800, 850, 900, 950, 1000, 1050, 1100]
    nu_values = [2.00, 4.426, 7.590, 11.44, 15.89, 20.92, 26.41, 32.39, 38.79, 45.57,
                 52.69, 60.21, 68.10, 76.37, 84.93, 93.80, 102.9, 112.2, 121.6, 131.2, 141.8]
    alpha_values = [2.54, 5.84, 10.3, 15.9, 22.5, 29.9, 38.3, 47.2, 56.7, 66.7,
                    76.9, 87.3, 98.0, 109, 120, 131, 143, 155, 165, 177, 195]
    Pr_values = [0.786, 0.758, 0.737, 0.720, 0.707, 0.700, 0.690, 0.686, 0.684, 0.683,
                 0.685, 0.690, 0.695, 0.702, 0.709, 0.716, 0.711, 0.707, 0.705, 0.702, 0.700]
    k_values = [9.34, 13.8, 18.1, 22.3, 26.3, 30.0, 33.8, 37.3, 40.7, 43.9,
                46.9, 49.7, 52.4, 54.9, 57.3, 59.6, 62.0, 64.3, 66.5, 69.0, 71.5]

    # Interpolators
    nu_interp = interp1d(T_values, nu_values, kind='linear')
    alpha_interp = interp1d(T_values, alpha_values, kind='linear')
    pr_interp = interp1d(T_values, Pr_values, kind='linear')
    k_interp = interp1d(T_values, k_values, kind='linear')

    altura_mm = float(input("Plate height (mm): "))
    largura_mm = float(input("Plate width (mm): "))
    comprimento_mm = float(input("Plate depth (mm): "))
    # Convertion into meters
    altura = altura_mm / 1000
    largura = largura_mm / 1000
    comprimento = comprimento_mm / 1000

    Ts_C = float(input("Surface temperature (B°C): "))
    Tamb_C = float(input("Ambient temperature (B°C): "))

    Ts = Ts_C + 273.15
    Tamb = Tamb_C + 273.15
    Tfilm = (Ts + Tamb) / 2
    delta_T = Ts - Tamb
    g = 9.81
    beta = 1 / Tfilm

    # Properties interpolations
    nu = nu_interp(Tfilm) * 1e-6
    alpha = alpha_interp(Tfilm) * 1e-6
    Pr = pr_interp(Tfilm)
    k = k_interp(Tfilm) * 1e-3

    # lengths and areas
    A = largura * comprimento
    P = 2 * (largura + comprimento)
    L_horizontal = A / P

    # Specific vertical length
    L_vertical = altura

    def calcular_ral_nul_hq(L, tipo):
        RaL = (g * beta * delta_T * L**3) / (nu * alpha)
        if tipo == "vertical":
            NuL = 0.68 + (0.670 * RaL**(1/4)) / ((1 + (0.492 / Pr)**(9/16))**(4/9))
        elif tipo == "horizontal_sup":
            if RaL <= 1e7:
                NuL = 0.54 * RaL**(1/4)
            else:
                NuL = 0.15 * RaL**(1/3)
        elif tipo == "horizontal_inf":
            NuL = 0.27 * RaL**(1/4)
        else:
            return None, None, None
        hq = NuL * k / L
        return RaL, NuL, hq

    # Calculation for specific cases
    Ra_v, Nu_v, hq_v = calcular_ral_nul_hq(L_vertical, "vertical")
    Ra_hs, Nu_hs, hq_hs = calcular_ral_nul_hq(L_horizontal, "horizontal_sup")
    Ra_hi, Nu_hi, hq_hi = calcular_ral_nul_hq(L_horizontal, "horizontal_inf")

    # Show results
    print("\n--- General Results ---")
    print(f"T_film: {Tfilm:.2f} K")
    print(f"OS = {nu:.2e} mBI/s | O± = {alpha:.2e} mBI/s | Pr = {Pr:.4f} | k = {k:.4f} W/mB·K")

    print("\n[1] Vertical:")
    print(f"  L = {L_vertical:.4f} m")
    print(f"  Ra_L = {Ra_v:.2e} | Nu_L = {Nu_v:.2f} | h_q = {hq_v:.2f} W/mBIB·K")

    print("\n[2] Horizontal - Upper hot surface (or lower cold):")
    print(f"  L = {L_horizontal:.4f} m")
    print(f"  Ra_L = {Ra_hs:.2e} | Nu_L = {Nu_hs:.2f} | h_q = {hq_hs:.2f} W/mBIB·K")

    print("\n[3] Horizontal - Lower hot surface (or upper cold):")
    print(f"  L = {L_horizontal:.4f} m")
    print(f"  Ra_L = {Ra_hi:.2e} | Nu_L = {Nu_hi:.2f} | h_q = {hq_hi:.2f} W/mBIB·K")

# Main Function
calcular_todos_hq()
```