>> Function: Atmospheric Density Calculator

Description: This function computes the air density for each corresponding altitude

This function is from HW #1

(1) INPUTS:

- altitude: altitude in the atmosphere [m] or [ft]
- unit: input to decide whether to do the computations in SI or English units

(2) OUTPUTS:

- air density: the air density [kg/m^3] or [slug/ft^3]

```
function [air_density] = airDensity_cal(altitude, unit)
%%
% *EQUATIONS*
%
% (1) The Atmospheric Pressure at "Pause" State
%
% $$p = p1 * exp((-1)* g * (h - h1) / R / T)$
%
% (2) The Atmospheric Pressure at "Sphere" State
%
% $$p = p1 * (T / T1)^(-g / R / T_h)$
%
% where   $T_h = (T - T1) / (h - h1)$
%
% (3) The Temperature at Certain Altitude
%
% $$ T = T1 + T_h(h - h1)$
%
% (4) The Density of Atmosphere at Certain Altitude
%
% $$ d = p/R/T$$
%
% (5) The Speed of Sound
%
% $$a = sqrt(y*p/d)$
%
% where   $y = gamma = 1.4$


%%
% altitude_ft = 0:500:100000; % Altitude vector in feet (ft)
% altitude_m = 0:1:30480;     % Altitude vector in meters (m)

    % *PREPARATION*
    altitude_m = altitude;
    g_si = 9.81;                 % Gravitational acceleration (m/s^2)
    R_si = 287;                  % Gas constant (J/kg/K)
```

```matlab
    gamma = 1.4;                % Adiabatic Index or Isentropic Expansion Constant
    lapse_rate_m = [-6.5*power(10,-3), 3*power(10,-3), -4.5*power(10,-3), ...
    4*power(10,-3)];           % Temperature lapse rates (K/m)
    mark_height_m = [0, 11, 25, 47, 53, 79, 90, 105]*1000;
                               % Height at which the the state changes
                               % from "pause" to "sphere" or vice versa
                               % (m)
    initial_temp_m = [288.16, 216.66, 282.66, 165.66, 256.66];
                               % Initial temperatures (K) where the
                               % state changes from "pause" to "sphere"
                               % or vice versa
    % *Temperature*
    %
    % Finding the temperature by altitude (K)
    % Meters
    temp_m = tempCal(initial_temp_m, altitude_m, mark_height_m,lapse_rate_m);
    % *Pressure*
    pressure_m = pressureCal(g_si, R_si, temp_m, initial_temp_m,...
        altitude_m, mark_height_m, lapse_rate_m, "SI");
    % *Density*
    density_m = pressure_m ./ temp_m / R_si;

    % *PREPARATION*
    altitude_ft = altitude;
    g_eng = 32.174;             % Gravitational acceleration (ft/s^2)
    R_eng = 1716.27;            % Gas constant (ft^2/s^2R)
    gamma = 1.4;                % Adiabatic Index or Isentropic Expansion Constant
    lapse_rate_ft = lapse_rate_m / 3.28084 * 1.8;
                               % Temperature lapse rates (R/ft)
    mark_height_ft = mark_height_m * 3.28084;
                               % Height corresponding to mark_height_m
                               % in feet (ft)
    initial_temp_ft = [518.688, 389.988, 515.988, 298.188, 461.988];
                               % Initial temperatures (R) where the
                               % state changes from "pause" to "sphere"
                               % or vice versa
    % *Temperature*
    %
    % Finding the temperature by altitude (K)
    % Feet
    temp_ft = tempCal(initial_temp_ft, altitude_ft, mark_height_ft, lapse_rate_ft);
    % *Pressure*
    pressure_ft = pressureCal(g_eng, R_eng, temp_ft, initial_temp_ft,...
        altitude_ft, mark_height_ft, lapse_rate_ft, "ENG");
    % *Density*
    density_ft = pressure_ft ./ temp_ft / R_eng;
if unit == "SI"
    air_density = density_m;
else
    air_density = density_ft;
end
end
```