```
clc
clear
file = 'weatherdataaligarh.xlsx';
data = readtable(file);

% Extracting data from columns
date_column = data{:, 2};
temperature_column = data{:, 3};
humidity_column = data{:, 4};
date_values = datetime(date_column, 'InputFormat', 'yyyy-MM-dd HH:mm:ss'); % Convert
date strings to datetime object

weather_temp = data{:, 3};
%Temp_inside = 8;  % inside temp in Celsius
Temp_inside = 1;  %onion inside temp in Celsius
%Temp_inside = 0;  % garlic inside temp in Celsius
U_wall = 0.275;  % in W/m^2/K
U_roof_insulated = 0.284;  % in W/m^2/K
A_wall = 44.38;   % total wall area in m^2
A_roof = 12.95;  % total roof area in m^2
%Cp = 3.43;   % specific heat for potato kJ/kg
Cp = 3.77;   % specific heat for onion kJ/kg
%Cp = 3.31;   % specific heat for garlic kJ/kg
H_g = 2500; %  latent heat of water  kJ/kg
M_commodity = 10^4;  % mass of commodity to be stored in cold storage
den_air = 1.225;  % Density in kg/m^3
Air_change = 4; % assumed 4 per day
v_room = 50;  % Volume of room in m^3
w_inside = 0.95 ; % inside humidity
cop_ref = 0.7; % COP of VARS
S_factor = 10;
Iirrad = 5;
N_con = 0.5; % efficiency of solar plate collector
P_vs = 3.165; % saturation pressure of air vapor in kPa
P_amb = 101.325; % atmospheric pressure in kPa

T_outside = data{:, 3};  % Temperature from Excel data
w_rel_data = data{:, 4};  %  Relative Humidity ratio from Excel data
w_rel_outside = w_rel_data / 100;
product_load = zeros(size(T_outside)); % Initialize array to store product_load
infiltration_load = zeros(size(T_outside)); % Initialize array to store infiltration
load
heat_transfer_wall = zeros(size(weather_temp)); % Initialize arrays to store the heat
transfer rates
heat_transfer_roof = zeros(size(weather_temp));

Q_inter = 2.3795; %   internal load in kW
Q_resp = 0.450; % respiration load in kW

for i = 1:length(weather_temp)
    % Calculate the temperature difference between the inside and outside
    temp_diff = weather_temp(i) - Temp_inside;

    % Calculate the heat transfer rate through the walls in kW
    heat_transfer_wall(i) = U_wall * A_wall * temp_diff / 1000;
```

```
    % Calculate the heat transfer rate if the roof is insulated in kW
    heat_transfer_roof(i) = U_roof_insulated * A_roof * temp_diff / 1000;

    % Calculate the infiltration load for each data point
    w_outside = 0.622 * w_rel_outside(i) * P_vs / (P_amb - w_rel_outside(i) * P_vs);
    Q_inf = den_air * Air_change * v_room * (1.005 * (T_outside(i) - Temp_inside) +↙
((w_outside / 100) - w_inside) * H_g) / (24 * 3600);
    infiltration_load(i) = abs(Q_inf) ;% in kW

    % Calculation of the average product load in kW
    Q5 = M_commodity * Cp * (T_outside(i) - Temp_inside);
    product_load(i) = Q5 / (3600 * 24 * 365);
end
total_heat_transfer = heat_transfer_wall + heat_transfer_roof;% Calculate the total↙
heat transfer as the sum of wall and roof heat transfer


% Calculate the total referigeration load
Total_ref_load = Q_inter + Q_resp + product_load + infiltration_load +↙
total_heat_transfer;
Total_ref_load=Total_ref_load*(1+S_factor/100);
figure;


plot(date_values, temperature_column, 'b', 'DisplayName', 'Temperature');% Plot↙
temperature and humidity data
hold on;
plot(date_values, humidity_column, 'r', 'DisplayName', 'Relative Humidity');
xlabel('Date');
ylabel('Temperature (°C) / Relative Humidity (%)');
title('Temperature and Relative Humidity Over Time');
legend('show');

%figure for the heat transfer rate vs. time and date, including the total
figure;
plot(date_values, heat_transfer_wall, 'b', 'DisplayName', 'Wall Heat Transfer');
hold on;
plot(date_values, heat_transfer_roof, 'r', 'DisplayName', 'Roof Heat Transfer');
plot(date_values, total_heat_transfer, 'g', 'DisplayName', 'Total Heat Transfer');
xlabel('Date');
ylabel('Heat Transfer Rate (KW)');
title('Heat Transfer Rate Over Time');
legend('show');
%figure for the infiltration load vs. time and date
figure;
plot(date_values, infiltration_load, 'b', 'DisplayName', 'Infiltration Load');
xlabel('Date');
ylabel('Infiltration Load (KW)');
title('Infiltration Load Over Time');
 legend('show');
 % figure for the product load vs. time and date
figure;
plot(date_values, product_load, 'r', 'DisplayName', 'Product Load');
xlabel('Date');
```

```
ylabel('Product Load (kW)');
title('Product Load Over Time');
legend('show');
% figure for the total reference load vs. time and date
figure;
plot(date_values, Total_ref_load, 'g', 'DisplayName', 'Total Refrigeration Load');
xlabel('Date');
ylabel('Total Refrigeration Load (kW)');
title('Total Refrigeration Load Over Time');
legend('show');
interval = 30;
a = length(Total_ref_load);
A = 0:interval:(interval * (a - 1));

% Check for NaN values in Total_ref_load
nan_indices = isnan(Total_ref_load);


% Remove NaN values from A and Total_ref_load
A = A(~nan_indices);help
Total_ref_load = Total_ref_load(~nan_indices);

% Calculate the trapezoidal rule approximation of the integral
integral_result = trapz(A, Total_ref_load);

fprintf('Energy required to run the cold storage for the whole year: %f kWhr\n',
integral_result/60);

E_gen_solar= integral_result*60/cop_ref/60/60;

fprintf('Total energy to be generated by solar to run the cold storage for whole year
: %f KWhr\n',E_gen_solar);
Area_solar=max(E_gen_solar)/(cop_ref*Iirrad*365*N_con);
fprintf('Area required for installing solar collector for 10  MT cold storage is %f
m^2\n',Area_solar);
```