Documentation File for Ex3.py

For Question 1, the code accomplishes several tasks related to analysing solar panel power generation using Python. It addresses Question 1, where it computes the power generated by different types of solar panels based on solar intensity data for December. The program calculates the power for each hour, plots the results, saves the plot as 'ex3_question1.pdf', and determines the maximum power produced by each type of solar panel. First of all the program reads solar intensity data from a CSV file, removes headers, and converts data to floats for computation. Then, the program utilizes the provided formula (Equation 10) to compute solar power generated by each type of solar panel for every hour of December. After that, it employs Matplotlib for plotting power versus time for different solar panel types on a single figure. Even though the code seems complicated it successfully computes the power generated hourly by three different types of solar panels and plots the power generated by each panel type against time, facilitating visual comparison then it determines and prints the maximum power produced by each panel type and identifies the type generating the greatest power. As a future development The program might be extended to include statistical analysis or forecasting techniques for better insights into solar panel performance.

For Question 2, This code addresses Question 2, focusing on determining the percentage of hours in December when the total power generated by 10 solar panels (Type A) falls below the required 250 W for house heating. It calculates the total power generated by the 10 panels, assesses the percentage of hours where the total power is insufficient for heating, and prints this percentage to the screen. The techniques which are used is First, Scaling Power: Utilizes a function to scale the power output of a single solar panel to represent the total power generated by 10 panels. Then, Conditional Evaluation: Determines the percentage of hours meeting a specific condition (power < 250 W) based on power consumption data. the program computes the total power generated by 10 solar panels of Type A based on the power generated by a single panel correctly and accurately determines the percentage of hours in December when the total power falls below the 250 W threshold for house heating. Further Developments can be achieved by: (1) Accuracy of Condition: The condition 'power < 250' effectively identifies the hours when alternative power sources might be required for heating. (2) Potential Extensions: Future iterations could consider exploring different conditions or incorporating more sophisticated analysis techniques to assess power variability and potential heating inadequacies more comprehensively.

For Question 3, This code tackles Question 3, aiming to compute and visualize the average power produced by solar panel Type A during each month of the year. It processes solar intensity data for each month, calculates the power at each hour from 12 am to 11 pm, averages the results for each month, and generates a plot displaying the average power against the months. The techniques used is first of all, utilizes the previously defined function to convert solar intensity data into power data for solar panel Type A. then, implements a function to compute the average power consumption per hour for each month based on the generated power data. after that, employs Matplotlib to create a plot showing the average power per month. I do this techniques because Converts intensity data into power data to enable the calculation of average power per month for solar panel Type A. therefore, uses a function to compute the average power consumption per hour, facilitating the determination of average power per month. As a further development further analysis could involve statistical measures or comparisons with other solar panel types to assess seasonal power variations more comprehensively.

For Question 4 , This code addressed by calculating wind turbine power and rotation rates at one-hour intervals from 12 am to 11 pm in July. It generates a plot illustrating the rotation rate as a function of time and prints the wind power at 5 pm. The Techniques Used: Data Transformation: Converts wind speed data from miles per hour to meters per second for accurate computations. And Rotation Rate Calculation: Utilizes Equations (11) and (12) to compute rotation rates of a wind turbine based on wind speed data for July. In addition to Power Calculation: Computes wind power generated by the turbine using the rotation rates obtained. The Rotation Rate Plot: Creates a graphical representation showcasing the rotation rate variations of the wind turbine over time during July. On the other hand The Wind Power Calculation: Accurately calculates wind power at specific hours, such as 5 pm, based on the rotation rates computed. We also Converts wind speed data into appropriate units (m/s) to facilitate accurate power and rotation rate calculations, compute rotation rates and derive wind power output from these rates. As a further recommendation, the code could be expanded to include additional analyses, such as wind power comparisons for different months or detailed trends in wind power generation across various hours.

For Question 5, This code addressed by computing the total power generated by six solar panels of Type A and one wind turbine. It calculates the total power at one-hour intervals from 12 am on June 1st to 11 pm on December 31st. Additionally, it generates a plot illustrating the total solar power, wind power, and overall power at one-hour intervals on December 1st. Lastly, it assesses if the total power dropped below 250 W in December. Techniques Used: Converts solar intensity data into power data for solar panel Type A. Calculates wind power based on wind speed data. then computes the total power combining solar and wind sources at hourly intervals for each month. After that, utilizes Matplotlib to create a plot representing solar power, wind power, and total power on December 1st. Techniques used because converts intensity data into power data to facilitate power calculations for both solar panels and the wind turbine. Also, utilizes a function to compute the total power at one-hour intervals for a day or a month from combined solar and wind sources. The generated plot effectively displays the hourly power variation, emphasizing the consistent wind power and the sporadic solar power on

Documentation File for Ex3.py

December 1st. As a further development the code could be expanded to include additional analysis, such as more extensive power comparisons between wind and solar or detailed energy production trends across various months.