You are provided data from historical records of sensors. Data is logged at intervals of 10 minutes. Sample data in comma-separated value text files is made available for this assignment. Each file contains a year's worth of data for multiple sensors. Data for each date-time recording are on separate rows. Within each row, a comma separates the value for each sensor.

Write a program to read in **all** files in the "Data" folder, storing these data:

WAST column = Date & Time

S column = Wind Speed

SR column = Solar Radiation

T column = Ambient Temperature

Other columns can be ignored.

Note: Data column "S" , "SR" , "T" with "N/A" values should be ignored by the program as well.

You are to create these minimum required .cpp & .h file for these classes:

Date: to store date data.

Time: to store time data.

Sensor: to store Wind Speed, Solar Radiation & Ambient Temperature data.

Design and then write an **Object Oriented program** in C++ that meets the specifications shown below. You should provide a suitable menu with an exit option in your main program. When designing the output, imagine yourself as the user of the program. You want the user interaction to be concise but user friendly on the command line. **Do not use GUI interaction.**

You must use/modify the provided template BST.h file and implement for the program usage. You MUST also implement the use of STL map.

The use of template BST and STL map is *mandatory*

You may use other data structures together with the STL map and template BST, but the main design should utilize the STL map and template BST.

You will need to explain the rationale to use these data structures in *your* particular way. The data structures can be used in many ways and so we want to know the thinking behind your choice of usage. Write down what you are thinking. There may not be a perfect answer to how to combine the required data structures as we are looking at what are your thinking processes when you use the data structures. You will need to highlight issues with your chosen approach as well.

You should be careful that you do not have data structure classes that do I/O. You may want to have dedicated I/O classes instead or let the main program deal with I/O. Make sure you modularise your main program.

You may use std::string and string stream classes in your program instead of using C like strings.
You may use iostream and file handling classes and objects in C++.
STL data structures and algorithms can be used.

Sample output formats shown below use made up data for the year 1905.

Menu options are:

Menu Option 1.

The average wind speed and average ambient air temperature for a specified month and year. (print on screen only)

Example output format if there is data:

```
January 1905: 5.5 km/h, 25.5 degrees C
```

Example output format if there is no data:

```
March 1905: No Data
```

Menu Option 2.

Average wind speed and average ambient air temperature for each month of a specified year. (print on screen only)

Example output format is:

```
1905
January: 5.5 km/h, 25.5 degrees C
February: 4.5 km/h, 27.5 degrees C
March: No Data
…
```

Menu Option 3.

Total solar radiation in kWh/m2 for each month of a specified year. (print on screen only)

Example output format is:

```
1905
January: 196.4 kWh/m2
February: 200.3 kWh/m2
March: No Data
…
```

Menu Option 4.

Average wind speed (km/h), average ambient air temperature and total solar radiation in kWh/m2 for each month of a specified year. (write to a file called “WindTempSolar.csv”)

Output Format:

Year Month,Average Wind Speed, Average Ambient Temperature, Solar Radiation

Example output format is:

```
1905
January,5.5,25.5,196.4
February,4.5,27.5,200.3
…
```

Year is printed on the first line and the subsequent lines list the month and the average wind speed, average ambient air temperature and the total solar radiation for each month. The values are comma separated.

For menu item 4: If data is not available for any month, do not output the month. In the example, March 1905 has no data. Nothing is output for March. If the entire year's data is not available, output just the year on the first line and the message "No Data" on the second line.

Menu Option 5.

Given a Date in the form d/m/yyyy, show the times for the highest solar radiation for that Date. Output to screen using this sample format: (There can be one or more time values that are output for the same date as the highest solar radiation value may be recorded more than once. The time values could be consecutive.)

```
Date: 11/1/2019
High solar radiation for the day: 1046 W/m2
Time:
10:40
11:30
```

The date, month and/or year are specified by the user. Your program asks for these on the command line and the user types in the required values and presses the "Enter" key.

Menu Option 6.

Exit the program.

Processing:

Your program must first load the data from multiple files. After loading the data, a menu is displayed to the user.