Evaluation document

**Data structure design choice:**

* BST is provided with delete that will only set the node data to 0 or “ “
* Data from sensor is loaded into vector, index month and year to BST linked by STL Map<int, int>
* Deletion of BST with date class will show deleted node data as 0. There is also no re-order of the nodes in BST to fill up the empty node. Therefore, the program design so that BST node should not be deleted.
* BST are aggregation to Vector means that BST will not be deleted if Vector is removed. BST stores date object of its own not pointer to Vector date object.
* BST loaded with date class that is inserted only for every month and year.
* The STL map stores key as int in the form of: yyyymmdd so that year has the highest order of significant. Map data is pointed to the location in Vector.
* Map linked the 1st date of month, year to Vector.at(i) to quick access the data instead of searching through the whole data set.
* When use a option to search data for result, the search will use index from map to limit the search to only the needed range

**Changes to the assignment 1:**

Since the program keep all data as each 10 minutes and not averaged it into each day, this program will keep that design since more data can be extract from storing full data.

Unit class is added to keep all sensor data. This mean that if next changes require the program to store more data from the file, it would only need to change the unit class instead of changing struct and calculations. Using class Unit is also for data hiding.

Unit test function find string, unit test BST template with delete that turn node data into 0 or “0” as default without re-order node order, unit test Map, unit test Unit class

Program open file .txt and store the data as a list of string. Then use file name in string to open the .csv file and read file as in assignment 1. The program also changes how it handles N/A and empty rows.

**How the program works:**

The file data is read into singleLine string that store and check each string data, then loads it into class Date, Time, Unit and then load into Sruct that push into Vector, then BST and Map

**Limitation**:

When a program meet a unexpected data in the file (such as “None” or empty or “ “ or “” instead of “N/A”) it will break the read file process without reading further rows beneath the file.

File output WindTempSolar keep only the last query data.

Time to read files can take very long (up to 7 minutes for 8 data files), 20MB for each file

No value check for date and time entered (e.g. date 135, month 546, year 12, hour 200, min -02)

No AVL for BST making this just the same efficiency of linked list.

Repetition of date, time will just add up the sensor data (no checking for repeated time)

# Testing

Test function readMetIndex.txt

ReadMetIndex

Test read multiple file with new data file

Handled “N/A” and “” line

Other unexpected string in data file will end loop for the ReadFile function

Test unit class

Test date class

Test getDateAsInt()

Test time class

Test function SearchString and checkNA

check NA return true then var that read file = 0 instead of read that spot in singleLine

SearchString nolonger used in the program.

Test IndexMonthYear

Insert into BST and STL map<int,int>

Test BST class

Test BST with int

Test BST with Date class

Test STL map class

Test map with <string, int>

Test map with <int, DateTime Struct>

Test map with <DateTime Stuct, int> and Comparison operator overloading for DateTime Struct

Test searchBSTTIme

Get a year then look for the 1st  occurrence in Map. Return the rows location in Vector.

Test ProcessData()

Search from BST a mth or year

# Output















