Data dictionary

| var\_name | Data type | Belongs to | Example | description |
| --- | --- | --- | --- | --- |
| speed | float | WindlogType struct | 6 | data read from input file stored in Windlogtype struct |
| solar\_radiation | float | WindlogType struct | 565 |
| air\_temperature | float | WindlogType struct | 20.25 |
| d | date | date WindlogType struct |  |
| t | time | time WindlogType struct |  |
| StructWind | WindlogType | ReadFile function |  | WindlogType struct is created to temporary hold vars read from file and pass to Vector object |
| monthParam | int | month\_to \_string function | 10 | parameter that take month in int to process in function |
| watt\_per\_min | float | Wh\_to\_kWh function | 512 | parameter that take solar radiation and do unit conversion |
| speed\_kmh | float | Kmh\_to\_Ms function | 5 | parameter that take speed and do unit conversion |
| windlog | Vector<WindlogType> | Vector<template> |  | this is the data structure that the program keep |
| month\_input | int | Process\_data function | 1 or 0 | parameter indicate if there will be month input |
| year\_input | int | Process\_data function | 1965 or 0 | parameter indicate if there will be year input and hold year input when function calculating |
| s | int | Process\_data function | 1 or 0 | parameter indicate if calculation involve speed |
| sr | int | Process\_data function | 1 or 0 | parameter indicate if calculation involve solar radiation |
| t |  | Process\_data function | 1 or 0 | parameter indicate if calculation involve ambient air temperature |
| file\_output |  | Process\_data function | true or false | parameter indicate if func output to file |
| found |  | Process\_data function | true or false | var indicate if search requirement |
| len | int | Vector<template> | 5600 | current rows in data |
| list | Vector<WindlogType> | Vector<template> |  | data structure of program |
| maxLen | int | Vector<template> | 12000 | capacity of the data structure |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |