

Wind Power Density Program Requirements, Phase 1

Honors Math 3199

Overview

For this lab activity, you will write a program that uses one month of wind measurements (downloaded from a provided website), to calculate

- Average speed vs. time of day,
- Average speed vs. day of month
- Wind Power Density for each of the above

A second lab will use the calculated values in the design of a wind farm that is installed at your selected location. You will choose turbines for the farm, and will add a column to the output file generated by this lab.

To accomplish this, you will permit an operator (user) to select one of three options:

- Finish, exit the program
- Calculate average wind speed vs hour of day, for all of the days of the month (average of hour 1, day 1, hour 1 day 2,.... Hour 1 day 20....)
- Calculate average wind speed vs day of month, for all of the hours of each day (average of all speeds day 1, average of all speeds day 2,)

Output will be written to a text file (see later section "Output File Structure" for details). You will read the text file into Excel, and use Excel graphing utility to show the data.

For a second lab, you will then choose a set of wind turbines that will be installed at the location, and will provide calculated value of wind power that can be added to the electrical grid. Your system must include estimated costs, and locations of the turbines.

Your system must consider wind variations across a three-month span; i.e. run your program with three consecutive months of wind data, consolidate output files into a single large file.

The output file will be read into Excel, where you will display averages graphically.

Overview: Enrichment\Challenge

Enable your program to read three (or more) months of data, storing in a larger array. Process the data as per paragraph.

Requirements: Phase 1, Available Wind

The Bonneville Power Administration maintains a website that includes wind measurement data for many locations under their system. These are found at URL <http://transmission.bpa.gov/business/operations/wind/MetData.aspx>. You will pick a site, and then use three months of data, from either 2014 or 2013 for your program. You should download a text file version of measurements from your chosen site.

Your program must

- Include functions which strip header from the data file
- Recognize bad data (wind speed value is "NaN") and not use it in average calculations
- Store data in a 2-D array. Array indices are day (of month) and hour (of day). Stored values must be "float" data type. You can either read an hour of samples and average it, then store that in the 2-D array, or read the entire month into a very large array, and from it extract values to be used in averages.
- Permit console selection of data type: i.e. 0 = done, 1 = hourly average (i.e. 7:00 AM for entire month, 8:00 AM....), 2 = daily average (i.e. day 1, day 2...)
- Write header lines to output file
- Write selected data to an output file
- Calculate wind power density for each hour or day, write in the output file

NOTE: See section "Output File Structure" for more details.

Wind Power Density

The Wind Power Density is calculated using the following:

$$\frac{P}{A} = \frac{1}{2} \rho v^2 \quad \text{Equation 1}$$

where

P = Power (Watts)

A = swept area of blades = πr^2 , r = blade radius

ρ = atmospheric density (notice that this is listed in the input file, you may use a single value for the entire month(s))

v = calculated average wind speed

Notice that Equation 1 is a normalized value; it is available wind power. For your initial calculations, you do not need to know a turbine blade radius. Rather you are calculating the value of the ratio, which is the right hand side of Equation 1.

Actual wind power delivered by a turbine is a modification of Equation 1:

$$P = \frac{1}{2} c A \rho v^2 = \frac{1}{2} c \pi r^2 \rho v^2 \quad \text{Equation 2}$$

where

A = swept area of blades = πr^2 , r = blade radius

c = efficiency of specific turbine

During Phase II, you will choose turbines for a wind farm at your chosen location, and a number of turbines. For these turbines, you will need to know their blade size (radius in Equation 2) and the efficiency of the turbine. If you cannot find efficiency data for a specific

turbine, make a reasonable estimate using typical efficiency values you find on manufacturer websites. That effort is described in a second lab requirements document.

Output File Structure

Output file structure will include:

- Header
- Column titles
- (user selects hourly averages) 24 rows, each with the average speed for that hour, and wind power density for that speed (3 columns; hour number, average speed, power density)
- (user selects daily average) 30 or 31 rows, each with the average speed for that day and wind power density for that speed (3 columns; day number, average speed, power density)

You may run your program with three separate input files (month files), and manually consolidate output files into a single file, or generate a single output file from three input files.

You will read your C-output file into Excel, and then create two graphs; one showing average speed vs time\day, a second which shows wind power density.

For this activity, you must include two types of documentation:

- User
- Maintenance

User documentation must be sufficient so that a person who is not a member of our class, can operate your program and understand the output. Do not assume that the person knows "C" or has written programs before (i.e. might enter "a", or "5" to your prompt, or even just press the "ENTER" button).

Maintenance documentation must be sufficient so that you can understand how the code works, how to debug it, and how to enhance it, if you or a classmate returns to this code next year. Maintenance documentation consists of inline comments, optionally supplemented by a separate and a small maintenance guide. Inline comments should not be excessive, but should describe function inputs, outputs and operation; key data variables, data structures, algorithm progression. Note that data variables whose names have obvious meaning, do not require separate comments; for example a comment is required (and shown) for the first variable below, but no separate comment is required for the second

```
int x1; // x1 is hourly average value  
int houravg;
```

Similarly, variables which are only used for loop counting, do not need to be commented:

```
for (int i = 1; i <= 24; i++) {.....}
```

Extra Credit

Extra credit will be awarded if your program is operated by a parent, using only your “User Manual”, the input files, and output file. In this test scenario, you may import into Excel and create graphs (your parent does not need to create the Excel graphs). See Appendix page I for a signature form for parent operation of the code.

Enable user to select a site, or which three months. It is acceptable for you to limit the site selection to one of two, three, or four. It is also acceptable for you to limit the months (i.e. download 6 month files, allow the user to select 3 consecutive months).

Assessment

A program that uses only a single “main” function, with no other functions, will receive a maximum grade of 84 (C). A program of this complexity should have three or more functions for reasons of comprehension and maintainability.

You will receive a 100 point grade for this project. Points will be collected as follows:

- Documentation (maintenance comments\guide, users guide): 10%
- Code modularization: 25%
- Robustness of code (operates with improper user input): 10%
- Use of control and looping structures: 5%
- Output file accuracy: 5%
- Excel diagram: 5%
- Array definition, loading, accessing 10%
- Algorithm which calculates average 10%
- Use of C-features (tokens, formatting of floats, strings, etc) 20%
- Extra Credit: parent operation 5%
- Extra Credit: user selection of months or locations: 5%

Appendices

Appendix I Parent Test Affirmation

Your Honors Math 3199 Student has written a program in “C”, which analyzes three months of wind speed measurements (data provided by a power company in the Pacific North West). To accomplish this, your student has implemented “C” features such as:

- Functions
- 2-D Arrays
- Loops
- Variables
- Algorithm development
- Basic text file I/O (read, write)

Because documentation is an important part of any STEM effort, your student has written a small “User Guide”, which describes how to operate the code.

He\she will receive extra credit if you successfully operate the program, using only the User’s Guide and any online information provided by the program.

After you operate it, your student will import the program’s output data into Excel, and create a graph of the data. You will view this data as part of the extra credit assignment.

Code Operation Affirmation

Using only the Users Guide and console prompts, I operated the code and viewed output (program text file, Excel graph).

Station: _____ (may be filled in by student)

Months: _____ (may be filled in by student)

Signature _____

Date: _____