Lab 11
Weather stats

10 April 2018

This lab will give you continued practice working with structs and with vectors and with vectors of structs (and writing functions for them, and additionally will involve reading in data rather than hand-building the vectors and structs in a test suite.

1. To get started on this part of the lab, copy some files into your directory. The task of the week will involve processing weather data, so first, you'll copy a set of weather data, found in

   ```
   /home/shared/160/weather-big.txt
   /home/shared/160/weather-small.txt
   ```

   into your own working directory for this lab.

2. Look at (but don't change) the contents of the files, then continue reading for their description.

These are data about the weather for the month of August 2009 in Galesburg, IL (where I worked at the time). Each line of the file contains information about the date and time it represents, as well as the temperature (in degrees Fahrenheit) and the wind speed (in miles per hour). This is a sample line illustrating the format:

```
08 02 2009      18 00   76      12
```

The date is first, then the time: this line represents August 2, 2009, at 18:00 (6pm). The temperature was 76°F, and the wind was blowing at 12mph. The "big" file contains measurements for every hour that month; the "small" file contains two days. (You're also encouraged to create your own files, even smaller, for testing purposes, but make sure they follow the same format.)

3. Based on the description above, and using the examples of `Card` and `Location` (and the struct you wrote for last week's lab), define a

`struct` called `Weather` that is capable of holding all the data on each line of our data file. This definition should go in a file `Weather.h` (which will also be where the related function headers will eventually go).

4. In a file called `run_weather_stats.cpp`, make sure to `#include` the file you just created, and then define a `main` that can read in all the pieces from one line and make a `Weather` value that bundles them up. (This struct will have seven fields, bigger than we've seen before.) (Look at the files in `/home/shared/160/0409/` to see what we defined in class, for a pattern to follow.)

5. Verify that it compiles. Have you started a readme yet to put the compiling and testing commands into? You should start a readme to put the compiling and testing commands into.

6. Wrap that in a loop, as we did in class Monday, that keeps reading until we run out of input, and builds a vector of `Weather` objects.

7. Add a function header to `Weather.h` that will determine whether the temperature in a given vector of weather values ever dipped below 55 degrees.

8. Create a file `Weather.cpp` with a stub for the function you designed in the previous step.

9. Create a file `test_Weather.u` and write tests for the function you've designed. Note that the vectors you use for your test cases only need to have a handful of values to effectively test the function, but they *should* be valid and plausible weather values.

10. Go back to the `run_weather_stats.cpp` file and after the vector of weather values is read in, have it print the result of a call to the function you've been writing. (Note that `cout` prints `bool` values as 1 and 0 by default, which is fine by me if it doesn't bother you.)

11. Finish writing the function.