

The Weather

*When Alexander saw the breadth of his domain, he wept
for there were no more worlds to conquer.*

Having conquered the challenge of dispensing little bits of paper with your account balance on them, the Bank needs to move on to new ambitions in pursuit of our goal to improve the financial well-being of all our customers: to wit, we are going to *predict the weather* for them.

We need you to provide plausible example data which we can use to test our weather models.

Task

Create a toy model of the environment (taking into account things like atmosphere, topography, geography, oceanography, or similar) that evolves over time. Then take measurements at various locations (ie weather stations), and then have your program emit that data, as in the following:

Station	Local Time	Conditions	Temperature	Pressure	Humidity
Sydney	2015-12-23 16:02:12	Rain	+12.5	1010.3	97
Melbourne	2015-12-25 02:30:55	Snow	-5.3	998.4	55
Adelaide	2016-01-04 23:05:37	Sunny	+39.4	1114.1	12

where temperature is in °C, pressure in hPa, and relative humidity as a %. Obviously you can't give it to us as a table (ok, yes, you could feed us markdown, but let's not do that?) so instead submit your data to us in the following format

```
SYD|-33.86,151.21,39|2015-12-23T05:02:12Z|Rain|+12.5|1004.3|97  
MEL|-37.83,144.98,7|2015-12-24T15:30:55Z|Snow|-5.3|998.4|55  
ADL|-34.92,138.62,48|2016-01-03T12:35:37Z|Sunny|+39.4|1114.1|12
```

with a three letter IATA code used as a station label.

Implementation

Your test data should have a reasonable number of reporting stations; 10±. Their locations and distribution over a reasonably large (continental sized) area should be somewhat sensible.

You should supply your answer to us in the form of a Git repository. If you'd like to keep it in a private repo on GitHub, that's fine; just add us as collaborators. Or you can send us a blob of your code; use `git bundle`.

Work in whatever programming language you want to. You will need to tell us how to *build* and *run* your code, however. Ideally this will be nothing more than `stack build`, or `sbt test`, or `python`

`GenerateWeather.py`, or `./configure` && `make` && `make install` or whatever is blatantly idiomatic in your language of choice.

For geography, there's a high-resolution map at [visible earth](#); we can send you *elevation.bmp* with the elevation data in the red channel. Feel free to use a different source of geography to generate test data against if you like.

Expectations

The whole idea is to have some fun with this. It really shouldn't take more than about **6 hours** of your time. If you don't think you can finish in a couple evenings, pare the scope back, and do a good job of the part you choose to do. Feel free to **contact us** if you have any points you'd like to clarify.

This exercise is an opportunity for you to demonstrate that you can take some interesting algorithms and implement them in appropriately *tested*, reasonably *performant*, and — most importantly — *readable* code.

We don't expect you to learn everything about meteorology (beyond some basic definitions perhaps), or to get the model "right" (in a climatology sense). Make reasonable guesses based on your experience about how weather changes, and come up with a way to produce similar seeming behaviour.

Generating plausible test data is a common requirement when building systems; random fuzz is useful for checking boundaries, but something that looks realistic helps developers see their systems in action. That's all this assignment is.

Having had a go your submission is then the starting point for the next conversation we'll have together. Engineering is about responsibility for choices and technical elegance in the face of feasibility constraints. We want you to talk about which parts of the problem you chose to do and why, identify areas where what you did do might have fallen short, and where you'd go next from here.

Look forward to talking with you about your code. Good luck!

`\vspace{3em}`

Engineering
Analytics & Information
Commonwealth Bank