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: to wit, we are going to *generate fake weather data* for a game.

We need you to provide a prototype of a program which artificially simulates the weather and outputs weather data in a standard format for our game to read.

Task

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

Location	Position	Local Time	Conditions	Temperature	Pressure	Humidity
Sydney	-33.86,151.21,39	2015-12-23 16:02:12	Rain	+12.5	1010.3	97
Melbourne	-37.83,144.98,7	2015-12-25 02:30:55	Snow	-5.3	998.4	55
Adelaide	-34.92,138.62,48	2016-01-04 23:05:37	Sunny	+39.4	1114.1	12

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:

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

where

- Location is an optional label describing one or more positions,
- Position is a comma-separated triple containing latitude, longitude, and elevation in metres above sea level,
- Local time is an ISO8601 date time,
- Conditions is either Snow, Rain, Sunny,
- Temperature is in °C,
- Pressure is in hPa, and
- Relative humidity is a %.

Your toy weather simulation should report data from a reasonable number of positions; 10±. The weather simulation will be used for games and does not need to be meteorogically accurate, it just needs to be emit weather data that looks plausible to a layperson.

So far we have assumed that our game takes place on Earth, leading to the use of latitude and longitude for co-ordinates and earth-like conditions. If you choose to assume that the game takes place elsewhere, please document any corresponding changes to the output format.

Implementation

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 simulation "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.

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!

Engineering
Analytics & Information
Commonwealth Bank