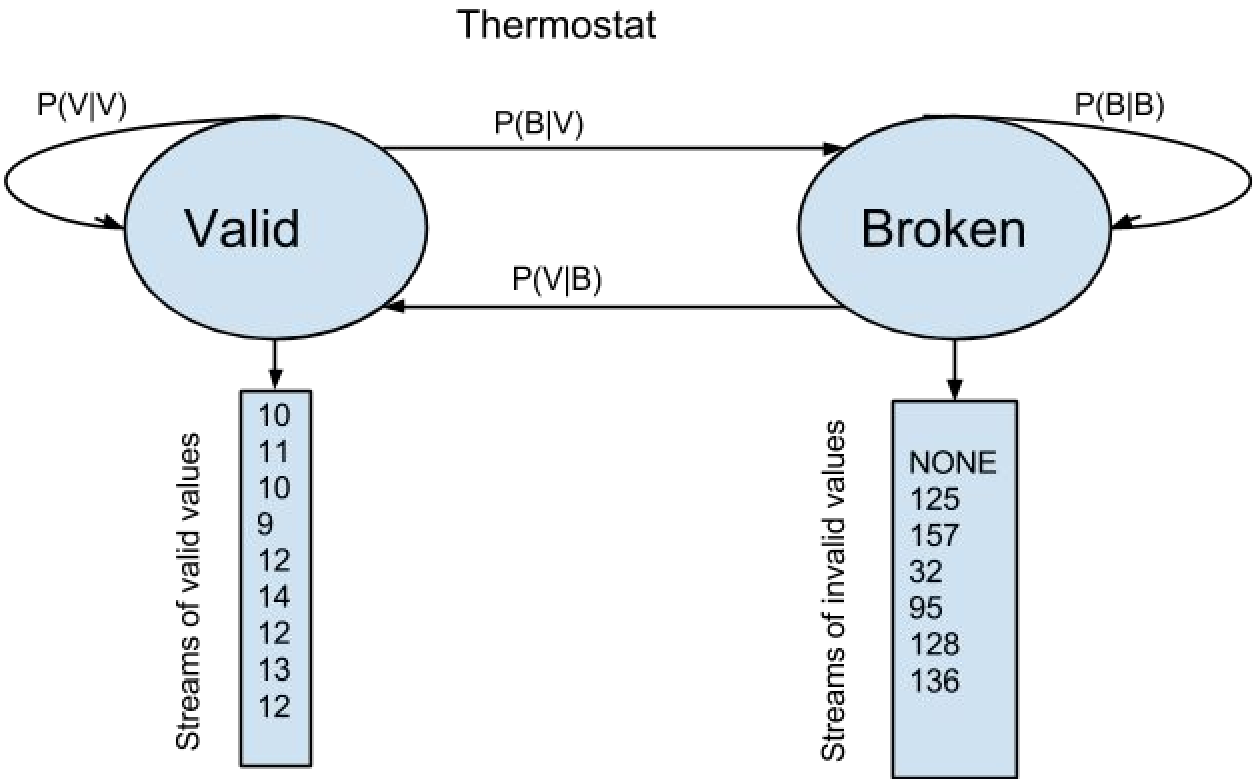
1. There’s a thermostat
2. Thermostat could be in 2 states (valid or broken).
3. Given the state of thermostat with some probability it can change it or stay at the same state.
4. There’s a lot of such thermostats in the system (parameter to the program, but assume a number like 100-1000000).
5. Each thermostat produces a stream of values.
6. Temperature could be stationary or dynamic (as you want) (bonus for dynamic).
7. At each point of time, based on the state thermostat could produce next values:

- If the thermostat is functional:
    - it produces a valid temperature value (from some distribution)
- If the thermostat is broken:
    - it can produce NONE value (hardware/network failure, etc.)
    - it can produce an invalid temperature value (anomaly)



TASK

For all software engineers:

1) Implement ThermostatSimulator class/function that encodes all the above information
2) Implement **Stream merging** for all of the thermostats in the system (for big data positions could be implemented on top of Spark Streaming/Storm/Samza/Akka/whatever or whatever streaming technology you like):

```
Input:
   Thermostat1: [1,2,3……...infinite stream of values]
   Thermostat2: [2,3,4……...infinite stream of values]
   Thermostat3: [3,5,6……...infinite stream of values]
   ………..
   ThermostatN: [100,101,102……...infinite stream of values]

Output:
   [[1,2,3,....,100],[2,3,5,...,101],[3,4,6,....,102],……...infinite stream of values]
```

3) Implement a **moving average** of averages from previous task:

```
    Input:
     [[1,2,3,....,100],[2,3,5,...,101],[3,4,6,....,102],……...infinite stream of values]
     window = 2

Output:
    [avg(avg([1,2,3,....,100]),avg([2,3,5,...,101])),
     avg(avg([2,3,5,...,101]), avg([3,4,6,....,102])),
      ..infinite stream of values]
```

For frontend/backend software engineers:

1) Implement a web monitoring tool for this streams (moving charts, etc.)

For big data engineers/data scientists:

1) Implement a thermostat machine learning model to detect anomalies in the streams or to predict the temperature value based on a metrics history. Also assume that temperature distribution could change over time (dynamic temperature). Assume that at broken state thermostat could produce some time correct or close to correct temperature values.

You can use any languages and libraries of your choice. You can add any custom metrics to Thermostat symulator, that would help you to solve a problem.

You should send zipped source code, instructions how to compile/run it. For a complex environment you can create docker image or virtual machine with pre installed application.