

# Computação em Larga Escala

(ano letivo 2024'25)

## 1 - Weather Stations

The goal of this exercise is to load a file with weather station data and then print for each station the average temperature, minimum and maximum temperature. For the exercise you will be using C++ and CMake to build the program.

### 1.1 - Project Structure

```
weather-stations/
├── CMakeLists.txt      # CMake configuration file
├── Makefile            # Make build configuration
├── src/
│   ├── main.cpp        # Main program file that reads and processes measurements
│   ├── cities.cpp       # Contains database of cities with their mean temperatures
│   └── create-samples.cpp # Program to generate sample measurement data
├── build/              # Build output directory (generated)
│   ├── cle-ws          # Main program executable
│   └── cle-samples     # Sample generator executable
```

### 1.2 - Build

Run the following command to build the program:

```
make
```

Run the following command to run the program:

```
./build/cle-ws <input_file>
```

### 1.3 - Input Data Format

The program reads a text file containing weather measurements. Each line in the file follows this format:

- Station name and temperature value separated by semicolon (;)
- Example: Tokyo;15.4

Input value ranges:

- Station name: UTF-8 string (1-100 bytes), containing neither semicolon (;) nor newline characters
- Temperature: double between -99.9 and 99.9 (inclusive), always with one decimal place
- Maximum of 10,000 unique station names
- Lines end with n character on all platforms

### 1.4 - Requirements

1. Modify the `main.cpp` file to:
  - Read the input file (provided as command line argument or default to “measurements.txt”)
  - For each station, calculate:
    - Average temperature
    - Minimum temperature
    - Maximum temperature
  - Print results sorted alphabetically by station name

2. Expected output format for each station:

```
London: avg=10.8 min=10.2 max=11.3  
Tokyo: avg=15.8 min=15.4 max=16.1
```

## 1.5 - Sample Data Generation

A program to generate sample data is provided ( `cle-samples` ). Usage:

```
./build/cle-samples NUMBER_OF_SAMPLES
```

This will create a file named `samples-NUMBER_OF_SAMPLES.txt` with random temperature measurements based on real city data.

As the number of samples increases, the program will take longer to run and the output file will be larger.

The ultimate goal is to process 1 Billion samples as fast as possible. However, you should start by trying to process at least 1 Million samples, then increase the number of samples until you see the program is taking too long to run.

## 1.6 - Example

For input file:

```
Tokyo;15.4  
London;10.2  
Tokyo;16.1  
London;11.3
```

Expected output:

```
London: avg=10.8 min=10.2 max=11.3  
Tokyo: avg=15.8 min=15.4 max=16.1
```

## 1.7 - Tips

- Consider using appropriate data structures to store and process the measurements
- Remember to handle file reading errors appropriately
- Pay attention to floating-point number formatting in the output
- For ordering the output, consider using a map or a sorted container