



CY-MÉTÉO

Gr6-MI

Grisez Alexandre, Dupontroupe Xavier and Laborde Camille

Planning and distribution of tasks

Dates	Xavier	Camille	Alexandre
From 09/12/22 to 15/01/23	<ul style="list-style-type: none"> ✓ Discover and understand the instructions ✓ Create a GitHub link ✓ Invite Camille and Alexandre to participate to the GitHub link ✓ Make the GitHub link public ○ C code : "the ABR sorting" part [/01/23] 	<ul style="list-style-type: none"> ✓ Discover and understand the instructions ✓ Start the PDF file ✓ Assign tasks to the team ○ C code : "the AVL sorting" part [18/01/23] 	<ul style="list-style-type: none"> ✓ Discover and understand the instructions ✓ Shell code : "the checking of settings" part ✓ Analyse the <i>meteo_filtered_data_v1</i> file ✓ C code : the common functions ✓ C code : "the list sorting" part
From 15/01/23 to 22/01/23	<ul style="list-style-type: none"> ○ C code : "the main program" part [31/01/23] 	<ul style="list-style-type: none"> ○ Shell code : "the drawing graphs" part (using Gnuplot) [24/01/23] 	<ul style="list-style-type: none"> ✓ Shell code : "the creating temporary file" part ✓ Code the Makefile
From 22/01/23 to 29/01/23		<ul style="list-style-type: none"> ✓ Change the C codes to adapt to Gnuplot needs ✓ Shell code : "the calling Gnuplot program" part 	<ul style="list-style-type: none"> ✓ Write the <i>README</i> file ✓ Write the <i>-help</i> command ○ Shell code : "the moving temporary file" part [30/01/23]
From 29/01/23 to 03/02/23	<ul style="list-style-type: none"> □ Merge the GitHub branch □ Test the whole program 	<ul style="list-style-type: none"> ✓ Merge the GitHub branch □ Test the whole program □ Complete the PDF file 	<ul style="list-style-type: none"> ✓ Shell code : "the calling C programs" part ✓ Merge the GitHub branch □ Test the whole program

Remaining tasks :

- Give the GitHub link

Legend :

- To do
- ✓ Made in time
- Made late [final date of making]



Functional limitations

--DESCRIPTION--

CY-METEO is a program designed to produce graphics from meteorological measurements, such as temperature, pressure, wind, height, moisture, in French regions of the world.

It was made in C and script shell. What we used to make graphs is GNUPLOT.

The program awaits a command from the user to start filtering data from a meteorological file. Using C functions, it then sorts the data in specific ways in order to allow gnuplot to produce the graphs.

The data can be filtered depending on time and regions, and the order of the sorts reversed. Three sorting modes are also available.

--HOW TO USE--

1. Download our files from this repository on your computer and put them in a directory of your choosing.
2. Inside your terminal, enter that directory.
3. Use the command "bash script.sh" with the options of your choice to start creating graphics ! Warning : some options must be used. If you have any question about how to use CY-METEO, use the following command : "bash script.sh --help"
4. If you have chosen multiple options, you will be able to see the graphs one at a time. To go on to the next one, simply close the current gnuplot window.
5. Type "make clean" in your terminal to delete all our executables / object files if you don't want them.
6. Have fun !



--LIST OF OPTIONS--

-f : serves to specify the path to the meteorological data file. Not using it will cause an error.

--avl : data is sorted by using AVL trees. If no sorting mode is specified, this is the default mode.

--abr : data is sorted by using ABR/BST trees.

--tab : data is sorted by using lists.

-r : the sorts will be reversed (ascending becomes descending and descending becomes ascending).

-t : the program will produce graphics on temperature in French regions.

You need to specify a mode as an argument.

Use -t 1 to get average, minimum and maximum temperature for each station.

Use -t 2 to get worldly average temperatures for each date and time.

Use -t 3 to get temperature measurements for each station, date and time.

-p : the program will produce graphics on pressure in French regions.

You need to specify a mode as an argument.

Use -p 1 to get average, minimum and maximum pressure for each station.

Use -p 2 to get worldly average pressure measurements for each date and time.

Use -p 3 to get pressure measurements for each station, date and time.

-w : the program will produce graphics on wind in French regions. The output will be a vector map representing average wind measurements in each region of the world.

-h : the program will produce graphics on the height of meteorological stations in French regions. The output will be a map with colors representing heights in each region.

-m : the program will produce graphics on moisture levels in French regions. The output will be a map with colors representing moisture levels in each region.

Fonctional limitations

-d : allows to specify a time interval from which the data used will come. The format of the time interval is "YYYY-MM-DDYYYY-MM-DD".

-F : the data used will only be from the stations located in metropolitan France and Corsica.

-G : the data used will only be from stations located in French Guiana.

-A : the data used will only be from stations located in the West Indies.

-S : the data used will only be from the meteorological station of St-Pierre et Miquelon.

-Q : the data used will only be from stations located in Antarctica.

-O : the data used will only be from stations located in the Indian Ocean.

--help : puts a help message on the terminal.

Please note that :

Using -f is mandatory to allow the program to get the path to the data file.

Using at least one option between -t -p -w -h -m is mandatory.

A mode always needs to be specified with -t and -p.

Using two options or more for the place is not possible (-F -G -A -S -Q -O).



Fonctional limitations

--INPUT FILE--

The input file is a CSV meteorological data file, with separator ";"

Every line corresponds to a measurement made by a station at a given time.

15 fields are used to provide informations and describe various aspects of the weather :

Field 1 : Station ID // 5 figure number

Field 2 : Date // datetime format : 2010-01-05T10:00:00+01:00 (str)

Field 3 : Pressure at sea level // Pa (int)

Field 4 : Average wind orientation over 10 min // degrees (int)

Field 5 : Average wind speed over 10 min // m/s (float)

Field 6 : Moisture // % (int)

Field 7 : Pressure // Pa (float)

Field 8 : Pressure variation over 24 hours // Pa (float)

Field 9 : Precipitations in the last 24 hours // mm (float)

Field 10 : Coordinates // (float),(float)

Field 11 : Temperature // °C (float)

Field 12 : Minimal temperature over 24 hours // °C (float)

Field 13 : Maximal temperature over 24 hours // °C (float)

Field 14 : Height // m (int)

Field 15 : City's postcode when there is one // (str)

Note : In some cases, data is missing on a line and nothing is written in certain fields. In this situation, if this data is needed for one option, the program will simply ignore this line.



--LIST OF FILES--

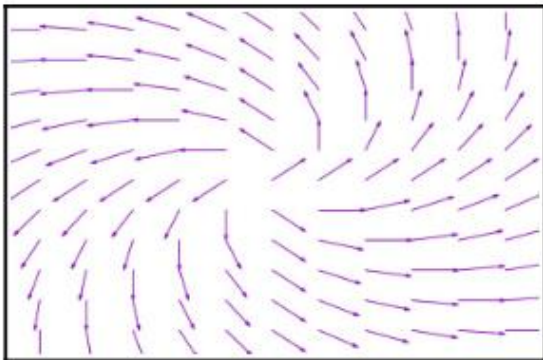
- **makefile** : Our makefile. It serves to compile all the C files and clean temporary files afterwards.
- **script.sh** : The main shell file that executes our project entirely.
- **C directory** :
 - **main.c** : Our main C program. It calls the different sorting modes depending on activated options.
 - **shared.h** : C header for the shared.c file.
 - **shared.c** : C file containing functions that are used in every sorting mode.
 - **tab.h** : C header for the tab.c file.
 - **tab.c** : C file containing all the functions that are necessary for sorting files using lists.
 - **avl.h** : C header for the avl.c file.
 - **avl.c** : C file containing all the functions that are necessary for sorting files using AVL trees.
 - **abr.h** : C header for the abr.c file.
 - **abr.c** : C file containing all the functions that are necessary for sorting files using ABR/BST trees.
 - **C/data directory** :
 - [Temporary files]
 - [Meteorological data file]

Fonctional limitations

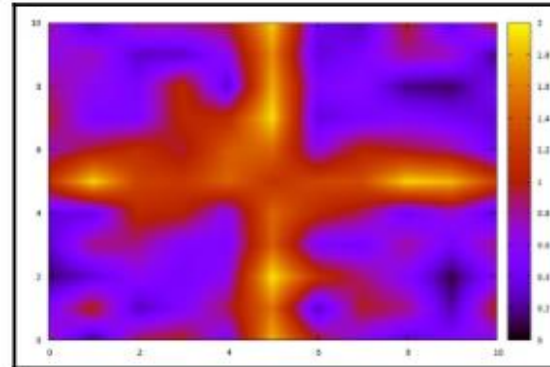
- **Gnuplot directory** :
 - **gnuT1.sh** : shell file used to make the graph of the option "temperature mode 1".
 - **gnuT2.sh** : shell file used to make the graph of the option "temperature mode 2".
 - **gnuT3.sh** : shell file used to make the graph of the option "temperature mode 3".
 - **gnuP1.sh** : shell file used to make the graph of the option "pressure mode 1".
 - **gnuP2.sh** : shell file used to make the graph of the option "pressure mode 2".
 - **gnuP3.sh** : shell file used to make the graph of the option "pressure mode 3".
 - **gnuW.sh** : shell file used to make the graph of the option "wind".
 - **gnuH.sh** : shell file used to make the graph of the option "height".
 - **gnuM.sh** : shell file used to make the graph of the option "moisture".
 - **Gnuplot/data directory** :
 - [Temporary files]
- **Informations directory** :
 - Specifications (PDF)
 - General informations (PDF)

Execution examples

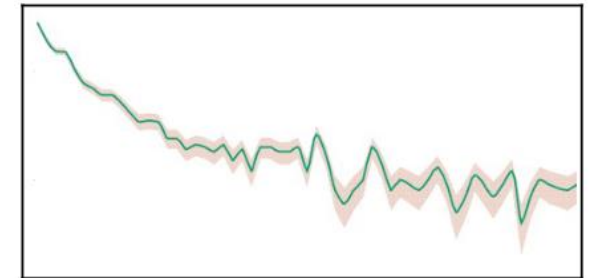
commande



commande



commande



(Exemple du cours à modifier avec les nôtres)

Analyse du fichier données

Lieu	ID
France métro politaine et Corse	00001 - 40349
Saint-Pierre et Miquelon	71805
Antarctique	89642
Guyane française	80001 - 88998
Océan Indien	61000 - 67006
Antilles	78000 - 79000

Le fichier *meteo_filtered_data_v1* est un fichier CSV. Le séparateur est ;

Il y a plusieurs colonnes (champs) qui correspondent chacune à une information précise pour une mesure météorologique d'une station. Chaque ligne est ainsi une mesure prise par une station. Une station peut avoir plusieurs mesures.

Les champs sont les suivants :

- Champ 1 : ID OMM station - nombre à 5 chiffres (str)
- Champ 2 : Date - format de la forme 2010-01-05T10:00:00+01:00 (str)
- Champ 3 : Pression au niveau mer - en Pa (int)
- Champ 4 : Direction du vent moyenne sur 10 mn - en degrés (int)
- Champ 5 : Vitesse du vent moyenne sur 10 mn - en m/s (float)
- Champ 6 : Humidité - en % (int)
- Champ 7 : Pression station - en Pa (float)
- Champ 8 : Variation de pression en 24 heures - en Pa (float)
- Champ 9 : Précipitations dans les 24 dernières heures - en mm (float)
- Champ 10 : Coordonnées - sous la forme de deux valeurs (float) séparés par une ,
- Champ 11 : Température - en °C (float)
- Champ 12 : Température minimale sur 24 heures - en °C (float)
- Champ 13 : Température maximale sur 24 heures - en °C (float)
- Champ 14 : Altitude - en m (int)
- Champ 15 : Code commune quand il y en a un - formats divers (str)