

The background of the slide features a digital theme. On the left, there's a world map composed of blue dots. Overlaid on this and the server racks are various data visualizations, including line graphs with red and blue lines, and bar charts. The server racks on the right are illuminated with blue light, creating a bokeh effect in the foreground.

RAPPORT PROJET : WEATHER REPORT

Simon Chanthrabouth-liebbe et Mohamed Amine Khaldi

Professeur : Mr GRIGNON

Romuald



SOMMAIRE

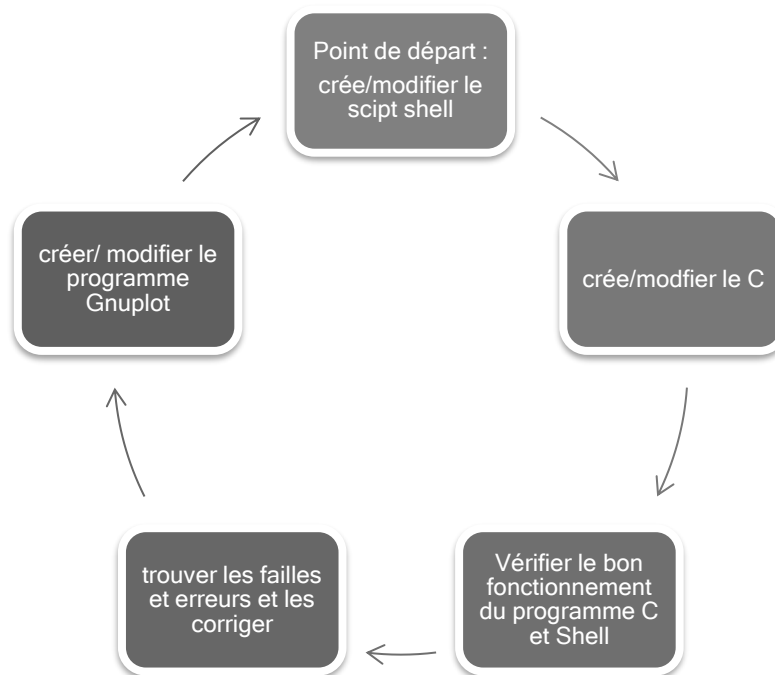
Explication du projet	3
Notre planning	4
Problèmes et solutions	5
Exemple d'exécution de l'application	6
Les options manquantes	10

EXPLICATION DU PROJET

Nous avons nommé notre projet « Weather Report » en référence à un personnage d'un manga japonais nommé Jojo's Bizarre Adventure. Notre application donc, permet de filtrer, trier et construire des graphiques à partir d'un fichier de données météorologiques. Ces données sont donc sous la forme d'un fichier CSV, que vous avez en possession. Vous n'aurez qu'à télécharger notre application depuis le lien GitHub que nous vous avons fourni, puis d'y joindre le fichier de données météorologiques.

Pour ce projet, le travail d'équipe était primordial afin d'arriver à un résultat satisfaisant. En effet, ce projet combine différents langages informatiques. Sans le script Shell pour filtrer les données, la partie programme C est compliquée à effectuer bien que possible. Pour travailler donc efficacement, Simon Chanthrabouth-liebbe a commencé par écrire un début du script Shell. Ce dernier a créé un programme de telle sorte que l'on puisse filtrer les données. Nous avons commencé par le plus simple : Height. En effet, de toutes les options à créer, l'altitude était la plus facile. Simon a donc commencé par filtrer en Shell les différentes altitudes des stations. Une station garde toujours la même altitude, alors Simon a comparé les altitudes entre elle, si c'était le cas, il comparait alors les ID de stations. Si les ID étaient identiques, alors on n'affichait pas la station. C'est ainsi que Simon créa un fichier Height.csv. Ensuite, Mohamed a commencé par créer le programme C pour trier ce fichier avec des ABR, AVL et le système de Chainon (on ajoute les valeurs dans l'ordre que l'on souhaite). Il a aussi créé la structure Mto, qui contient tous les types de données qui existent (Ex : moisture, height, temperature...), mais aussi value_sorted qui nous sert à savoir avec quelle valeur de la structure Mto on doit trier.

En se basant sur le modèle de Height, nous avons créé les différentes fonctions C et script Shell. Pour résumer nous nous sommes basés sur le cycle ci-dessous :



Notre Planning

C'est ainsi que nous nous sommes réparti le travail. Simon s'occupa du script Shell et la moitié du programme C, tant dis que (Mohamed) je m'occupai de du Gnuplot et l'autre moitié du programme C. Nous avons commencé à travailler sur le projet le 09/12/22, et nous avons terminé le projet le 03/02/23, 23h59.

Le planning était simple ; nous travaillons en appel discord, environ tous les jours de la semaine, après les cours. Certaines semaines néanmoins, nous avons manqué de régularité et nous n'avons pas pu travailler sur le projet.

Voici un exemple de notre planning au début :

Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
Autres	Projets	Projets	Projets	Projets	Autres	Autres
Autres	Autres	Autres	Autres	Autres	Autres	Autres

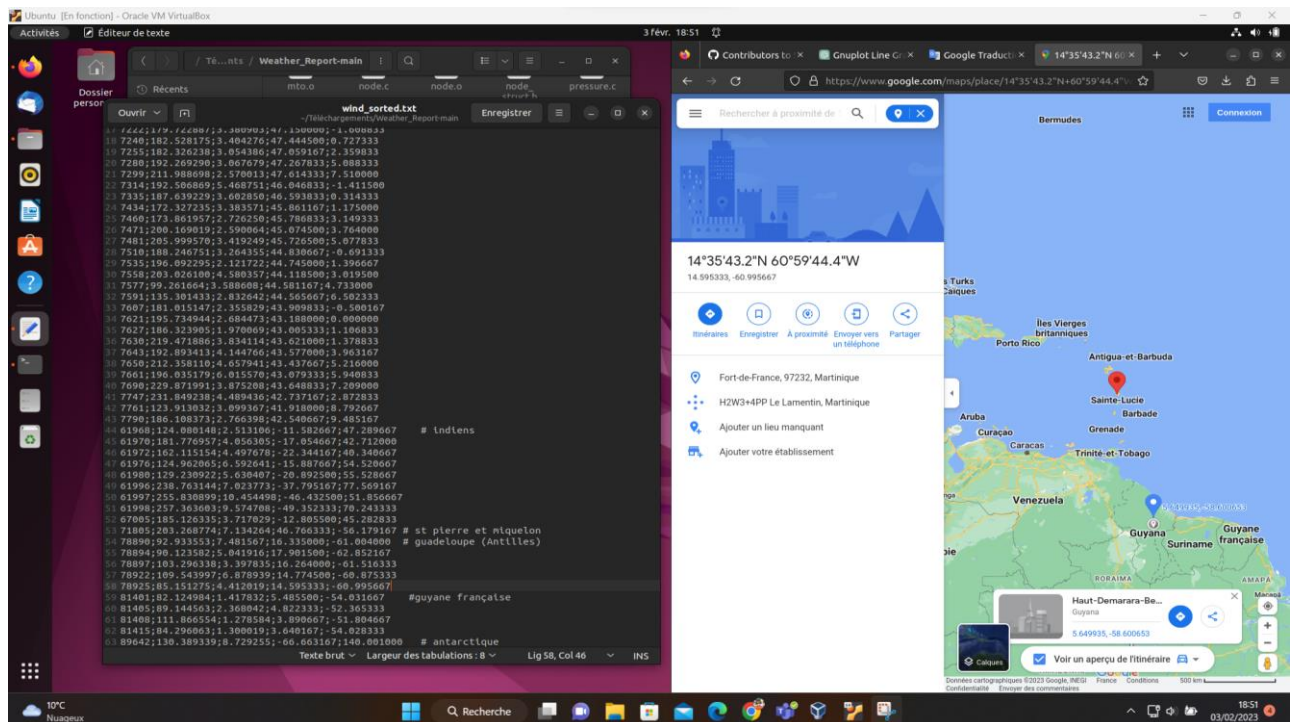
La dernière semaine avant le rendu finale, nous avons travailler toute la journée, chaque jour de la semaine. Nous avons passé des nuits blanches à travailler et chercher des solutions aux problèmes que nous avons rencontré.

Problèmes et solutions

Nous avons en effet rencontré des difficultés, comme par exemple les dates qui étaient sous la forme YYYY-MM-DDT00:00:00+00:00 et qu'il fallait donc transformer en un nombre. Pour cela nous avons décomposé ces valeurs en une structure Time.

Je vous laisse aller voir le programme pour comprendre comment nous avons réussi à faire cela.

Pour les stations, on sait d'après les consignes qu'il y a 6 régions. On a donc trié les stations par leurs ID, nous avons remarqué que les ID étaient similaires suivant les régions aux quels ils appartiennent. C'est ainsi que par exemple, pour la France, les ID se situent entre 7005 et 79790. De plus, les coordonnées des stations étaient proches à quelques dizaines près.



Pour les valeurs qui sont inexistantes dans le fichier csv, comme pour la température ou la pression par exemple, on remplace la valeur de la température par -9999, une valeur impossible.

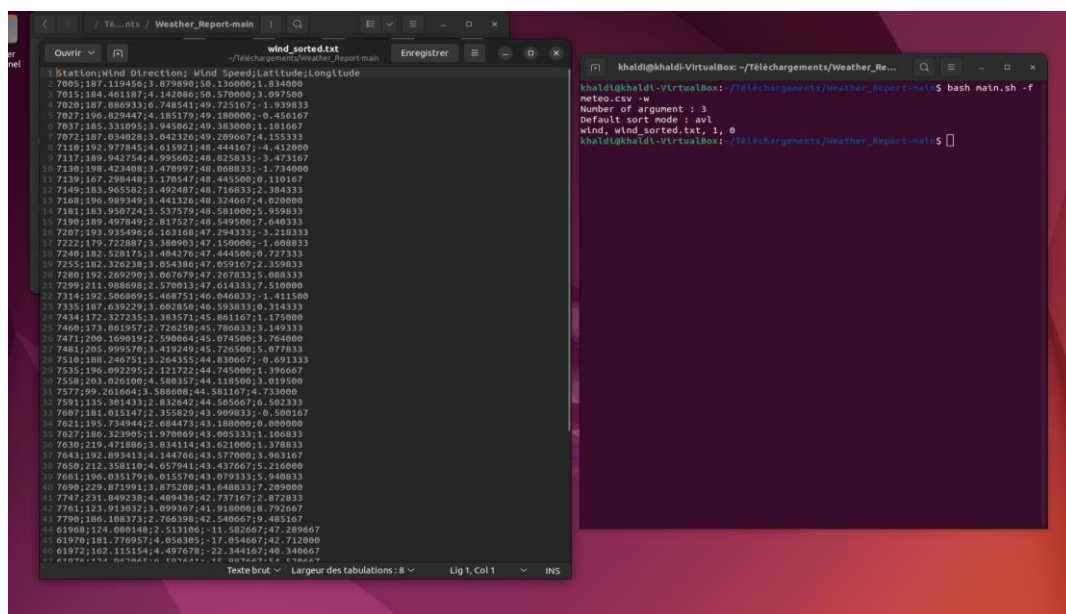
Si on ne le fait pas, la température sera considéré comme 0, ce qui peut fausser les résultats de la moyenne des température

Lorsque le code rencontre une valeur qui est à -9999, on l'ignore, ou on la remplace par une autre valeur qui existe

Concernant la vitesse d'exécution avec les options, la vitesse est relativement rapide, sauf pour l'option Tab combiné avec l'option t2 ou p2, sans d'autres options de stations. Le langage Gnuplot était un peu compliqué au départ à comprendre, le timing était serré et il fallait donc apprendre vite. Bien entendu, des recherches sur internet en étaient effectuées.

Exemples d'exécution de l'application

Nous allons voir différents exemples d'exécution de weather report :



```

1 Station;Wind Direction; Wind Speed;Latitude;Longitude
2 7005;187.119456;3.879890;50.136000;1.834000
3 7015;184.461187;4.142086;50.570000;3.097500
4 7020;187.080933;6.740541;49.725167;-1.939833
5 7027;196.829447;4.185179;49.180000;-0.456167
6 7037;185.331095;3.945862;49.383000;1.181667
7 7072;187.034028;3.042326;49.209607;4.155333
8 7108;192.977845;4.615921;48.444167;-4.412000
9 7117;189.942754;4.995602;48.825833;-3.473167
10 7130;198.423400;3.470997;48.068833;-1.734000
11 7139;167.298448;3.170547;48.445500;0.110167
12 7149;183.965582;3.492487;48.716833;2.384333
13 7168;196.989349;3.441326;48.324607;4.620000
14 7181;183.950724;3.537579;48.581000;0.959833
15 7190;189.497849;2.817527;48.549500;7.640333
16 7207;193.935496;6.163108;47.294333;-3.218333
17 7222;179.722887;3.380903;47.150000;-1.608833
18 7240;182.528175;3.404276;47.444500;0.727333
19 7255;182.326238;3.054386;47.059167;2.359833
20 7280;192.269290;3.067679;47.267833;5.088333
21 7299;211.988698;2.570013;47.614333;7.510000
22 7314;192.506869;5.468751;46.046833;-1.411500
23 7335;187.639229;3.602850;46.593833;0.314333
24 7434;172.327235;3.383571;45.861167;1.175000
25 7460;173.861957;2.726250;45.786833;3.149333
26 7471;200.169919;2.590004;45.074500;3.764000
27 7481;205.999570;3.419249;45.726500;5.077833
28 7510;188.246751;3.264355;44.830607;-0.691333
29 7535;196.092295;2.121722;44.745000;1.396607
30 7558;203.026100;4.580357;44.118500;3.019500
31 7577;199.261604;3.580808;44.581167;4.733000
32 7591;135.301433;2.832642;44.565607;0.502333
33 7607;181.015147;2.355829;43.909833;-0.500167
34 7621;195.734944;2.684473;43.180000;0.000000
35 7627;186.323905;1.970009;43.005333;1.180333
36 7630;219.471886;3.834114;43.621000;1.378833
37 7643;192.893413;4.144766;43.577000;3.963167
38 7650;212.358110;4.657941;43.437607;5.216000
39 7661;196.035179;6.015570;43.079333;5.940833
40 7690;229.871991;3.875208;43.648833;7.209000
41 7747;231.849238;4.409436;42.737167;2.872833
42 7761;123.913832;3.099367;41.918000;8.792667
43 7790;186.108373;2.766398;42.540607;9.485167
44 81968;124.080148;2.511186;11.582667;47.289667
45 61970;181.776957;4.056305;17.054667;42.712000
46 61972;162.115154;4.497678;22.344167;48.348667
47 61973;162.115154;4.497678;22.344167;48.348667

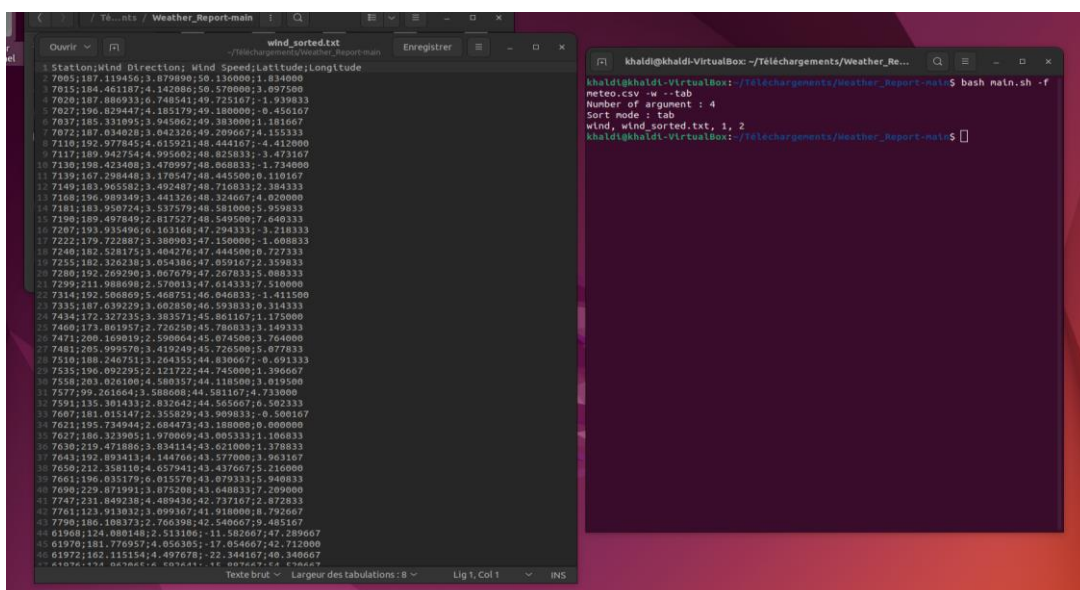
```

```

khaldi@khaldi-VirtualBox: ~/Téléchargements/Weather_Re...
khaldi@khaldi-VirtualBox: ~/Téléchargements/Weather_Report-main$ bash main.sh -f
meteo.csv -w
Number of argument : 3
Default sort mode : avl
wind, wind_sorted.txt, 1, 0
khaldi@khaldi-VirtualBox: ~/Téléchargements/Weather_Report-main$

```

Exemple du tri par avl du vent



```

1 Station;Wind Direction; Wind Speed;Latitude;Longitude
2 7005;187.119456;3.879890;50.136000;1.834000
3 7015;184.461187;4.142086;50.570000;3.097500
4 7020;187.080933;6.740541;49.725167;-1.939833
5 7027;196.829447;4.185179;49.180000;-0.456167
6 7037;185.331095;3.945862;49.383000;1.181667
7 7072;187.034028;3.042326;49.209607;4.155333
8 7108;192.977845;4.615921;48.444167;-4.412000
9 7117;189.942754;4.995602;48.825833;-3.473167
10 7130;198.423400;3.470997;48.068833;-1.734000
11 7139;167.298448;3.170547;48.445500;0.110167
12 7149;183.965582;3.492487;48.716833;2.384333
13 7168;196.989349;3.441326;48.324607;4.620000
14 7181;183.950724;3.537579;48.581000;0.959833
15 7190;189.497849;2.817527;48.549500;7.640333
16 7207;193.935496;6.163108;47.294333;-3.218333
17 7222;179.722887;3.380903;47.150000;-1.608833
18 7240;182.528175;3.404276;47.444500;0.727333
19 7255;182.326238;3.054386;47.059167;2.359833
20 7280;192.269290;3.067679;47.267833;5.088333
21 7299;211.988698;2.570013;47.614333;7.510000
22 7314;192.506869;5.468751;46.046833;-1.411500
23 7335;187.639229;3.602850;46.593833;0.314333
24 7434;172.327235;3.383571;45.861167;1.175000
25 7460;173.861957;2.726250;45.786833;3.149333
26 7471;200.169919;2.590004;45.074500;3.764000
27 7481;205.999570;3.419249;45.726500;5.077833
28 7510;188.246751;3.264355;44.830607;-0.691333
29 7535;196.092295;2.121722;44.745000;1.396607
30 7558;203.026100;4.580357;44.118500;3.019500
31 7577;199.261604;3.580808;44.581167;4.733000
32 7591;135.301433;2.832642;44.565607;0.502333
33 7607;181.015147;2.355829;43.909833;-0.500167
34 7621;195.734944;2.684473;43.180000;0.000000
35 7627;186.323905;1.970009;43.005333;1.180333
36 7630;219.471886;3.834114;43.621000;1.378833
37 7643;192.893413;4.144766;43.577000;3.963167
38 7650;212.358110;4.657941;43.437607;5.216000
39 7661;196.035179;6.015570;43.079333;5.940833
40 7690;229.871991;3.875208;43.648833;7.209000
41 7747;231.849238;4.409436;42.737167;2.872833
42 7761;123.913832;3.099367;41.918000;8.792667
43 7790;186.108373;2.766398;42.540607;9.485167
44 81968;124.080148;2.511186;11.582667;47.289667
45 61970;181.776957;4.056305;17.054667;42.712000
46 61972;162.115154;4.497678;22.344167;48.348667
47 61973;162.115154;4.497678;22.344167;48.348667

```

```

khaldi@khaldi-VirtualBox: ~/Téléchargements/Weather_Re...
khaldi@khaldi-VirtualBox: ~/Téléchargements/Weather_Report-main$ bash main.sh -f
meteo.csv -w --tab
Number of argument : 4
Sort mode : tab
wind, wind_sorted.txt, 1, 2
khaldi@khaldi-VirtualBox: ~/Téléchargements/Weather_Report-main$

```

Exemple du tri par Chainon du vent


```

khalid@khalid-VirtualBox: ~/Téléchargements/Weather_Report-main$ bash main.sh -f
meteo.csv -w --tab
Number of argument : 4
Sort mode : tab
wind, wind_sorted.txt, 1, 2
khalid@khalid-VirtualBox: ~/Téléchargements/Weather_Report-main$ bash main.sh -f
meteo.csv -n --abr
Number of argument : 4
Sort mode : abr
moisture, moisture_sorted.txt, 1, 1
khalid@khalid-VirtualBox: ~/Téléchargements/Weather_Report-main$

```

The terminal window shows the execution of a script that sorts data by humidity. The command used is `sort -n -k 1,1 moisture_sorted.txt`. The output is a list of coordinates and humidity values sorted in ascending order.

Exemple du tri par abr de l'humidité

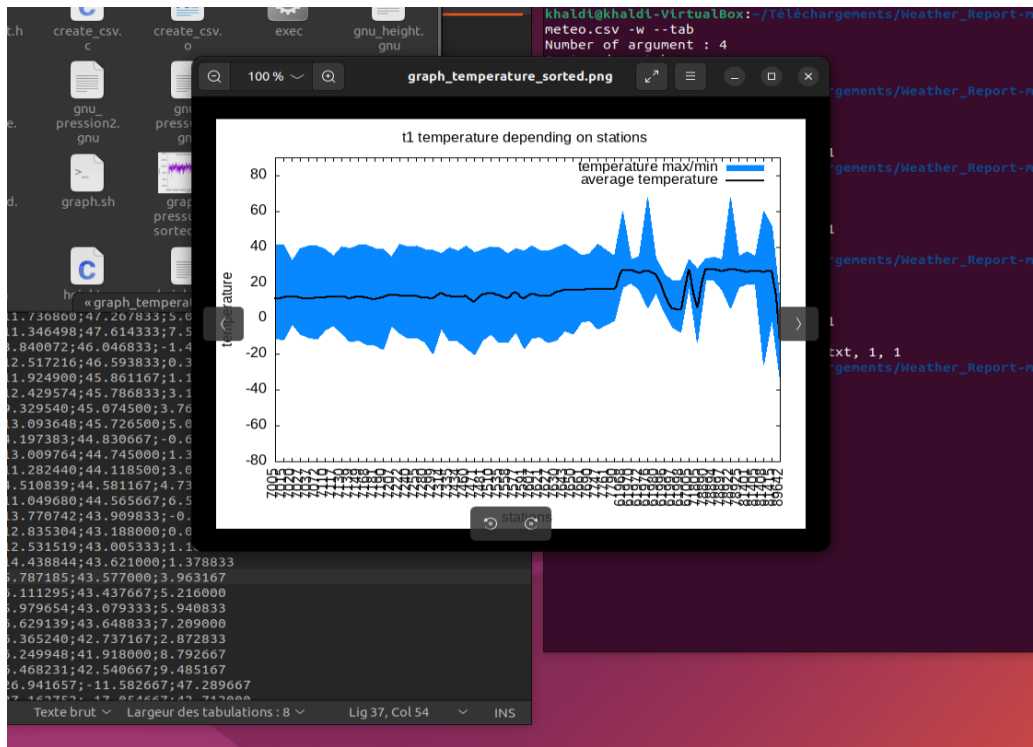
```

khalid@khalid-VirtualBox: ~/Téléchargements/Weather_Report-main$ bash main.sh -f
meteo.csv -w --tab
Number of argument : 4
Sort mode : tab
wind, wind_sorted.txt, 1, 2
khalid@khalid-VirtualBox: ~/Téléchargements/Weather_Report-main$ bash main.sh -f
meteo.csv -n --abr
Number of argument : 4
Sort mode : abr
moisture, moisture_sorted.txt, 1, 1
khalid@khalid-VirtualBox: ~/Téléchargements/Weather_Report-main$ bash main.sh -f
meteo.csv -n --abr -w
Number of argument : 5
Sort mode : abr
moisture, moisture_sorted.txt, 1, 1
wind, wind_sorted.txt, 1, 1
khalid@khalid-VirtualBox: ~/Téléchargements/Weather_Report-main$ bash main.sh -f
meteo.csv -n --abr -w -t1
Number of argument : 6
Sort mode : abr
moisture, moisture_sorted.txt, 1, 1
wind, wind_sorted.txt, 1, 1
temperature1, temperature1_sorted.txt, 1, 1
khalid@khalid-VirtualBox: ~/Téléchargements/Weather_Report-main$

```

The terminal window shows the execution of a script that sorts data by multiple criteria. The command used is `sort -n -k 1,1 -k 2,2 -k 3,3 temperature1_sorted.txt`. The output is a list of coordinates and temperature values sorted in ascending order.

Exemple de plusieurs options à la fois



Exemple de graph de température

```

khalidi@khalidi-VirtualBox:~/Téléchargements/Weather_Report-main$ bash main.sh -f
meteo.csv -h -tab
Number of argument : 4
Default sort mode : avl
height, height_sorted.txt, 1, 0
Unknown option '-tab', please choose between '-t1' '-t2' '-t3', use --help for more
details
khalidi@khalidi-VirtualBox:~/Téléchargements/Weather_Report-main$

```

Exemple d'erreur

```

khalidi@khalidi-VirtualBox:~/Téléchargements/Weather_Report-main$ bash main.sh met
eo.csv -h --tab
Number of argument : 3
Sort mode : tab
There is no meteo data file selected, please use -f [file_name.csv], use --help
for more details
khalidi@khalidi-VirtualBox:~/Téléchargements/Weather_Report-main$

```

Exemple d'erreur


```

khalid@khalid-VirtualBox: ~/Téléchargements/Weather_Report-main
khalid@khalid-VirtualBox:~/Téléchargements/Weather_Report-main$ bash main.sh --help
This script filter and sort a meteo data csv file (specially the one we are using in this project) depending on the option we are selecting
Syntax : (bash) main.sh [argument]<option>
Argument :
--help      Show this help text
Exemple : bash main.sh --help
-f [file]   Select the meteo data file (THIS COMMAND IS ESSENTIAL)
Exemple : bash main.sh -f meteo.csv
-t<1|2|3>   Create a file containing the temperature depending of the option : 1) Average temperature, minimum et maximum temperature depending of the station
                                                2) Average temperature depending of the time
                                                3) Average temperature depending of the time and the station
Exemple : bash main.sh -f meteo.csv -t2
-p<1|2|3>   Create a file containing the atmospheric pressure depending of the option : 1) Average atmospheric pressure depending of the station
                                                2) Average atmospheric pressure depending of the time
                                                3) Average atmospheric pressure depending of the time and the station
Exemple : bash main.sh -f meteo.csv -p1
-w         Create a file containing the wind with the average direction and average wind speed for each station
Exemple : bash main.sh -f meteo.csv -w
-h         Create a file containing the height for each station
Exemple : bash main.sh -f meteo.csv -h
-n         Create a file containing the maximum moisture for each station
Exemple : bash main.sh -f meteo.csv -n
-r         Ascending to descending mode or descending to ascending mode
Exemple : bash main.sh -f meteo.csv -n -r
--avl      Choose the avl sort mode, the fastest (default sort mode)
Exemple : bash main.sh -f meteo.csv -n -avl
--abr      Choose the abr sort mode, slower than the avl mode
Exemple : bash main.sh -f meteo.csv -h --abr
--tab      Choose the tab sort mode, the slowest (with the use of nodes)
Exemple : bash main.sh -f meteo.csv -w --tab
khalid@khalid-VirtualBox:~/Téléchargements/Weather_Report-main$

```

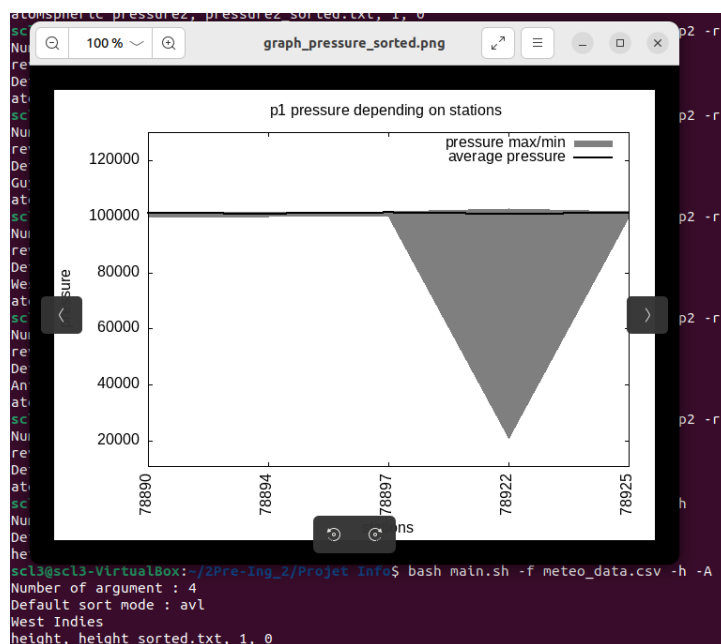
Exemple de la commande -help

```

scl3@scl3-VirtualBox:~/2Pre-Ing_2/Projet Info$ bash main.sh -f meteo_data.csv -p2 -r -G
Number of argument : 5
reversed order mode
Default sort mode : avl
Guyana
atmospheric pressure2, pressure2_sorted.txt, 0, 0
scl3@scl3-VirtualBox:~/2Pre-Ing_2/Projet Info$ bash main.sh -f meteo_data.csv -p2 -r -A
Number of argument : 5
reversed order mode
Default sort mode : avl
West Indies
atmospheric pressure2, pressure2_sorted.txt, 0, 0

```

Exemple de commande de régions



Exemple de pression pour les Antilles

Les options manquantes

Malheureusement, faute de temps, nous n'avons pas pu faire les options -t3 et -p3, ainsi que leurs programmes gnuplot respectifs. Nous n'avons pas aussi fait les options et arguments du script [DATES].

Merci pour votre lecture, nous espérons que vous alliez apprécier notre projet !