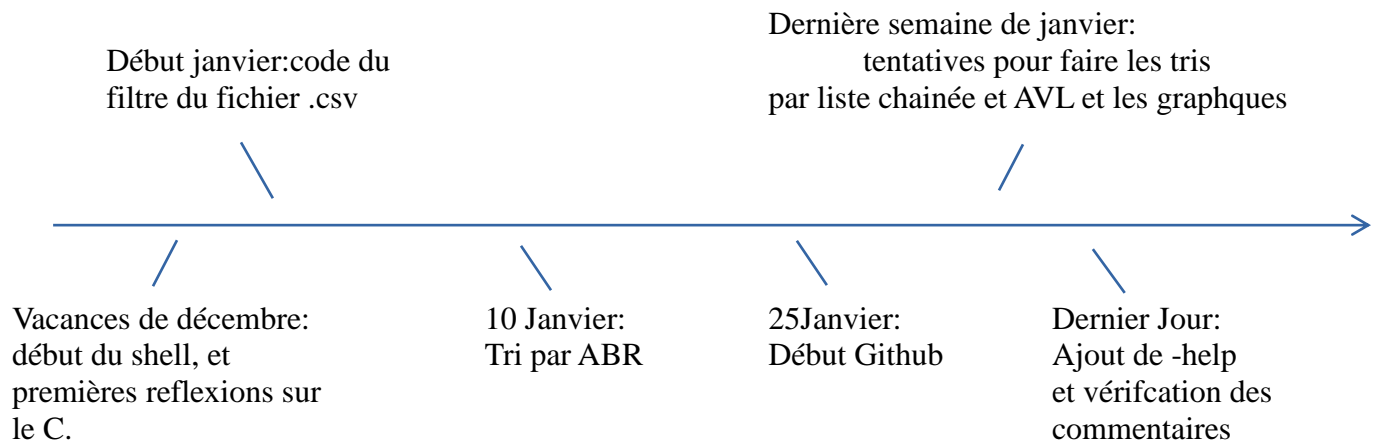


# Compte Rendu



Le projet n'est pas entièrement fini: nous avons bien un filtre qui s'effectue sur notre fichier `meteo_filtered_datav1.csv` qui nous permet d'avoir un nouveau fichier .csv qui ne comporte que les colonnes en lien avec les paramètres rentrés par l'utilisateur. Le tri par ABR fonctionne, ce qui n'est pas le cas du tri par des listes chaînés ou du tri AVL. Les deux sont cependant implémentés. Nous avons 2 graphiques qui marchent, celui de P1 et celui de T1, trié par ABR.

Makefile:

Ce bout de programme est un fichier Makefile qui définit comment construire un projet en utilisant GCC.

Il compile les fichiers source "mainc.c", "abr.c", "avl.c" et "linkedlist.c" en fichiers objets "mainc.o", "abr.o", "avl.o" et "linkedlist.o". Les fichiers objets sont ensuite liés ensemble pour produire l'exécutable "main-project".

Les options de compilation sont définies par FLAGS et incluent l'optimisation du code pour une exécution plus rapide (-O3) et la génération de symboles de débogage (-ggdb).

Le répertoire de sortie pour les fichiers objets est défini par BIN\_DIR.

La cible "clean" permet de supprimer les fichiers objets du répertoire bin/ et l'exécutable "main-project".

abr.c

Ce code implémente un ABR (Arbre binaire de recherche) en C. Il y a plusieurs fonctions qui permettent de créer un ABR, d'insérer des éléments dans l'ABR, de parcourir l'ABR en ordre infixe et de libérer la mémoire allouée pour l'ABR. La fonction creerABR alloue de la mémoire pour un nouvel élément de l'ABR et initialise ses membres (elt, right, left). La fonction insertp1ABR et insertl1ABR insèrent respectivement des éléments dans l'ABR en comparant leur "codes". La fonction infixeABR parcourt l'ABR en ordre infixe et écrit les informations de chaque élément dans un fichier. Enfin, la fonction freeABR libère la mémoire allouée pour l'ABR.

avl.c

Ce code implémente un arbre AVL (un type d'arbre binaire de recherche équilibré) pour stocker des données de stations météorologiques. La structure d'arbre comporte des informations sur le code de la station, la moyenne, le minimum et le maximum des données météorologiques. Les fonctions "creer", "rotright", "rotleft", "doubleg", "doubled" et "equilibrer" sont utilisées pour maintenir l'équilibre de l'arbre lors de l'insertion de nouveaux éléments. La fonction "insertAVL" permet d'insérer de nouvelles données de station dans l'arbre. La fonction "infixeAVL" permet de parcourir l'arbre de manière infixée et d'écrire les données de chaque nœud dans un fichier.

## linkedlist.c

Ce bout de programme est un code C qui implémente liste chaînée. La liste chaînée est utilisée pour stocker des données structurées dans un tableau où chaque élément peut être ajouté ou retiré sans avoir à déplacer les autres éléments.

Les fonctions principales sont :

- ⑩ createLL() : crée une liste chaînée vide.
- ⑩ is\_empty(LinkedList l) : vérifie si la liste est vide.
- ⑩ allocNode(Station st) : alloue un nouveau noeud dans la liste.
- ⑩ getNode(LinkedList l, Station st) : trouve un élément dans la liste et le retourne. Si l'élément n'est pas trouvé, il est ajouté à la liste.
- ⑩ writeCSV(LinkedList l, FILE \*fp) : écrit les données de la liste dans un fichier CSV.
- ⑩ freeLL(LinkedList l) : libère la mémoire allouée pour la liste

## mainc.c

Ce code C lit un fichier CSV qui contient des informations sur les stations météorologiques. Il insère les informations lues dans un arbre AVL ou un arbre binaire de recherche en fonction de la méthode choisie (définie par la constante #define). Les données sont lues ligne par ligne et insérées dans l'arbre en tant que nœuds, chaque nœud contenant les informations d'une station (code, avg, min, max).

Une fois que toutes les données ont été insérées dans l'arbre, il parcourt l'arbre en utilisant la fonction infixe pour écrire les données dans un fichier CSV nommé "step\_2.csv".

## .gp

Un fichier .gp est un fichier de script Gnuplot, utilisé pour générer des graphiques en 2D et 3D à partir de données.

w

Exemples:

Voici le fichier de données d'entrées:(fichier donnée)  
meteo\_filtered\_data\_v1.csv

Voici le fichier de données intermédiaires(filtré):  
step\_2.csv

Ce fichier prend les colonnes qui ont des valeurs en rapport avec les arguments donnés par l'utilisateur.

Voici le fichier de fin trié par ABR:

last.csv

Ce fichier est le fichier trié par ABR qui contient que les bonnes colonnes

Voici le graphique, lorsque que nous executons cette commande:

```
./main.sh -O -f meteodata.csv -abr -o last.csv -p1
```

Voici le graphique lorsque que nous executons cette commande:

```
./main.sh -O -f meteodata.csv -abr -o last.csv -t1
```

Liam MARTIN  
Marius BROSSET