



# Projet C-Wire



---

## Organisation

---

### I. Outils et mode de fonctionnement

Pour mieux organiser notre travail, un groupe WhatsApp a été créé afin de faciliter la communication et le partage d'informations. Pour la partie programmation, nous utilisons le site Replit, qui nous permet de coder dans n'importe quel langage souhaité. Bien que le script Shell soit testé sur le terminal d'un ordinateur sous Linux, Replit nous offre un espace de travail commun pour consulter et travailler sur le code de chaque membre du groupe.

Concernant le Shell, nous avons codé sur une fenêtre unique nommée c-wire.sh. Pour la partie en C, Tran a décidé de créer plusieurs fenêtres pour faciliter le travail, avec des noms spécifiques pour chaque fonction : CreateStation, ResearchStation, rotateLeft, rotateRight, doubleRotationLeft, doubleRotationRight, equiBalance, insertStation, delMinStation, deleteStation, analyzesLine, buildTreeFromData, calculateTotalConsumption, collectResults, freeTree. En cas de difficulté, nous communiquons nos doutes ou incompréhensions afin de nous entraider. Si nécessaire, nous utilisons l'intelligence artificielle pour obtenir des éclaircissements ou des indications.

### II. Répartition du travail

Pour réaliser ce projet, nous avons réparti les tâches de la manière suivante : Elsa et Aida se sont chargées de la partie Shell, tandis que Tran a écrit le programme en C. Pour la partie en C, Tran a utilisé l'intelligence artificielle pour obtenir une liste des fonctions à implémenter. Cela lui a permis de mieux organiser son travail.

Pour la partie Shell, il était essentiel de relire le cours et de bien se remémorer les commandes afin d'avoir une base solide. Elsa et Aida ont commencé par les consignes les plus simples, ce qui leur a permis de gagner en agilité. Grâce à cette approche, l'écriture du code a pu démarrer efficacement.

En ce qui concerne la répartition du travail entre Elsa et Aida, elles ne l'ont pas faite de manière formelle, mais se sont plutôt assistées mutuellement dans l'écriture de chaque ligne de code. À la fin, nous nous sommes réunis pour rédiger le fichier main. Cependant, nous n'avons pas eu le temps de réaliser les bonus.

---

## Écriture du programme

---

Voici la correction et la révision de votre texte, avec des améliorations au niveau de la grammaire, de la syntaxe et de la clarté :

### III. Problèmes rencontrés et solutions trouvées

Durant la réalisation du projet, plusieurs problèmes sont apparus, notamment en ce qui concerne l'utilisation de GitHub. La manipulation du site s'est avérée assez compliquée, en particulier pour l'enregistrement progressif du programme. Aucune d'entre nous ne savait réellement comment procéder. Pour résoudre ce problème, nous avons regardé des tutoriels afin de mieux comprendre le fonctionnement de GitHub et son utilisation dans le cadre du projet.

En outre, lors de la programmation, certaines difficultés se sont présentées. Parmi elles, il y a eu des problèmes de compréhension des consignes, qui ont pris du temps à être claires pour nous. Pour la partie Shell, cela a retardé le début de l'écriture du code, car nous voulions nous assurer de bien comprendre l'objectif du projet. Le principe était assez complexe, et certaines étapes, comme la création de l'exécutable à l'aide du Makefile, nous ont posé problème. Pour mieux comprendre son fonctionnement, nous avons utilisé ChatGPT pour nous aider à trouver la bonne syntaxe, car au début, Elsa et Aida pensaient qu'il fallait utiliser la commande `gcc -o`, ce qui n'était finalement pas le cas.

La consigne de filtrage s'est aussi avérée complexe à coder, car il y avait une grande quantité de données à traiter, ce qui demandait une attention particulière pour ne rien oublier. De plus, il était nécessaire de prévoir toutes

les erreurs possibles et de coder les conditions adéquates pour les gérer correctement.

Pour la partie en C, Tran a rencontré des difficultés pour lire les données CSV, car c'était une tâche nouvelle pour elle. Afin de résoudre ce problème, elle a sollicité des explications plus détaillées de la part de la professeure d'informatique, puis a cherché des solutions sur ChatGPT. Elle a également eu du mal avec la structuration des entrées et la gestion de la compilation du code en même temps que la gestion des dossiers.

La mise en commun du code C et du script Shell a également été un défi. Lors de la compilation, des problèmes se sont souvent posés, soit dans la partie C, soit dans la partie Shell. Il était donc difficile et long de cerner les erreurs et de les résoudre de manière efficace.

L'erreur la plus longue que nous avons rencontrée concernait un problème de mémoire dans le programme en C. Les fonctions utilisées pour coder les arbres AVL étaient issues du cours, et nous pensons que ce sont ces fonctions qui ont provoqué le problème de mémoire. Cela a rendu le déroulement du code Shell particulièrement difficile.

#### IV. Infos supplémentaire à noter

Dans les infos supp, il est dit « Le programme C et tous les fichiers qui s'y rapportent (makefile, exécutable,...) devront être situés dans un dossier 'CodeC'. Nous n'avons pas réussi à mettre le Makefile ni le main (même si cela n'est pas préciser). Sinon le makefile ne compile pas, il ne retrouve pas les fichiers.