

RAPPORT DU PROJET D'INFORMATIQUE



Le projet a débuté au début du mois de novembre, avec un rendu fixé au 22 décembre au plus tard. Le groupe était composé de trois membres : Synthia, Hajar et Samia. Lors de la première semaine, nous avons commencé par examiner le sujet en détail en le lisant plusieurs fois, en prenant des notes et en identifiant les différentes étapes nécessaires pour mener à bien le projet. Nous nous sommes également réunies pour travailler sur le shell et Unix, car nous n'avions pas encore terminé les travaux dirigés correspondants. Cette étape a permis à toutes les trois d'acquérir une bonne base en shell. Samia a ensuite créé un document Google partagé ainsi qu'un espace de code sur GitHub, afin que nous puissions collaborer efficacement. Nous avons démarré par les vérifications en shell, en écrivant les premières lignes de code pour vérifier l'existence du fichier input, la lisibilité des fichiers, etc. Ces étapes étaient cruciales. Par la suite, nous avons abordé la partie "data_exploration". Il fallait créer un fichier de sortie et le nommer dynamiquement en fonction des données fournies par l'utilisateur. Après plusieurs essais de codes différents, nous avons trouvé une version fonctionnelle que nous avons comité. La partie "execute_program" était également difficile, car elle consistait à obtenir un fichier de sortie depuis le shell pour l'utiliser dans notre programme en C, puis le retranscrire dans le shell. Synthia et Samia se sont occupées de cette partie, tandis qu'Hajar développait les fonctions de base en C. Elle a notamment travaillé sur les AVL (arbres à recherche équilibrée) en divisant le code en plusieurs fichiers : un fichier pour les fonctions AVL (AVL_tree), un autre pour la structure d'un AVL, et un Makefile que nous avons finalisé ultérieurement. Après plusieurs tests, nous avons trouvé une version correcte pour la partie "execute_program" et avons complété le Makefile, ce qui a permis de finaliser cette partie. À ce stade, tout fonctionnait correctement, sauf pour le cas où l'utilisateur saisissait "lv all", car la partie min-max n'était pas encore codée. Nous avons alors réfléchi à la manière de trier le fichier "lv all" en fonction de la consommation, puis de la valeur absolue de la différence entre capacité et consommation. En parallèle, Synthia travaillait sur l'ajout d'un en-tête aux fichiers de sortie, comprenant les informations sur le type de station, la capacité et la consommation en première ligne. Enfin, Samia et Hajar ont trouvé une solution permettant de gérer correctement le cas "lv all". Le fichier min-max dans le dossier "tests" respectait désormais les normes définies par le professeur. Nous avons ainsi peaufiné quelques erreurs pour que notre code soit capable de gérer tous types de problèmes. Ainsi, notre code était robuste.