

**Projet IAP : Les fondamentaux de la programmation impérative**

**Objet : Rapport du projet IAP**



**Noms des participants :** Rams Léo, Lenouvel Louis, Esteves Gabriel

**Groupe :** 104



## Table des matières

<-----Présentation du projet----->

<-----Organisation des tests----->

<-----Bilan du projet----->



## Présentation du projet

### **Brève présentation :**

L'objectif de ce projet était d'apprendre à manipuler un fichier texte en plus d'utiliser tous types d'algorithmes qui font utiliser un grand nombre de fonctions en langage C. Nous avons pour chaque Sprint un fichier texte avec dedans des noms de fonctions suivis de ce que l'on doit utiliser dans ladite fonction. Il y avait trois types de fonctions : la première permettait de définir le nombre de tournois traités qui était inscrit dans le fichier texte; la seconde permettait d'enregistrer un tournoi avec ses caractéristiques qui sont son nom, sa date et la liste de ses matchs avec pour chaque match le nom de la gagnante d'abord et le nom de la perdante ensuite. La dernière permettait d'afficher du texte en fonction de ce que l'on voulait comme résultat.

### **Rôles et entrées sorties des Sprints :**

Pour le Sprint 1, Le fichier texte entré correspond à 1 tournoi de 127 Matches et 128 Joueuses, avec sa date et son nom. L'objectif était de retourner toutes les informations enregistrées, soit le nom et la date du tournoi ainsi que la liste de ses matchs dans l'ordre.

Pour le Sprint 2, le fichier texte entré est le même que pour le Sprint 1 mais l'affichage attendu est différent. L'objectif ici est d'afficher tous les matchs d'une joueuse dont le nom est donné à la fin du document.

Pour le Sprint 3, le fichier texte entré est toujours le même mais l'affichage attendu est les noms des joueuses triés par ordre alphabétique suivis à chaque fois de leurs points dont la valeur dépend du nombre de victoires de la joueuse lors du tournoi.

Pour le Sprint 4, Il y a pour la première fois plusieurs tournois à enregistrer, et l'affichage attendu est les noms des joueuses triés en fonction de leurs points cumulés lors de tous les tournois, les noms doivent également être triés par ordre alphabétique lorsque les points de plusieurs joueuses sont égaux.

Pour le Sprint 5, Il y a également plusieurs tournois et l'affichage attendu est similaire à celui du Sprint 4 mais la différence est que la fonction `afficher_classement` apparaît plusieurs fois et doit à chaque appel afficher le classement correspondant aux 4 derniers tournois enregistrés. Il y a également plus de tournois que pour le Sprint 4.



## Organisation des tests

### **Organisations des tests :**

Les tests ont été effectués sur l'invite de commande `cmd` dans le dossier contenant le `Sprint.exe`. Nous exécutons le `Sprint` sur lequel nous travaillons en important le fichier `txt` correspondant et vérifions si le résultat renvoyé était cohérent avec le résultat attendu. S'il nous semblait que le résultat était bel et bien cohérent, nous utilisons un comparateur de texte pour vérifier que le résultat était bien celui attendu. Si c'était le cas, alors nous utilisons plusieurs comparateurs, 2 à 3 pour être sûrs, et si les textes étaient identiques à chaque fois, alors nous passons au `Sprint` suivant.

Lorsqu'il y avait une erreur cependant, nous utilisons bien souvent des `printf` que nous mettons à chaque endroit où il pouvait potentiellement y avoir une erreur pour voir plus précisément ce que faisait le programme. Nous avons surtout souvent mis des `printf("IF")` pour des `if` par exemple ou des `printf("FOR")` pour des `for` pour ainsi voir combien de fois le programme exécutait les boucles ou les conditions. Si l'erreur n'était pas trouvée par ce moyen, nous imaginions ligne par ligne l'exécution du programme pour comprendre à quel moment il ne faisait pas ce qui était attendu.

### **Bilan de validation des Sprints :**

La plupart du temps, nous trouvions assez rapidement la structure du programme et le moyen de réussir à obtenir ce que l'on souhaitait, mais nous oublions un détail ou commettons une erreur discrète qui nous prenait beaucoup de temps à trouver et à régler. Nous avons passé la majeure partie de notre temps à chercher où nous nous étions trompés en utilisant les méthodes détaillées plus tôt. Un exemple à cela est lors de la réalisation du code du `Sprint 4` où nous avons défini un nouveau `typedef struct` qui correspondait à la liste des joueuses, le problème étant que ce nouveau `typedef` amenait à des problèmes de mémoire et que donc le programme affichait à partir d'un certain moment des caractères non saisis et supprimait des joueuses. Nous avons essayé plusieurs fonctions dont la fonction `malloc` qui permet de déplacer des données vers le tas mais n'avons rien trouvé de concluant. Nous avons finalement changé d'approche en définissant la liste des joueuses et celle des points associés séparément, ce qui a fonctionné.

La plupart du temps, lorsque nous comparons les textes affichés et attendus avec un comparateur de texte, le résultat était le bon, car les erreurs étaient assez visibles lors des essais effectués avant.



## Bilan du projet

### **Difficultés rencontrées :**

Plusieurs difficultés ont été rencontrées, la plus grande étant celle citée plus haut, soit celle à propos de la mémoire et du tas. Il y a également souvent eu des problèmes de base comme des soucis d'indice dans les boucles for ou encore des soucis de tailles de listes.

Le programme étant de plus en plus long, il était de plus en plus difficile de voir où était l'erreur, surtout lorsqu'il fallait modifier des fonctions du début du programme, lorsqu'il a fallu considérer plusieurs tournois par exemple. Le programme nécessitait beaucoup de boucles for, ce qui signifie que les indices n'étaient pas toujours évidents à implémenter.

Enfin, la majeure partie des fonctions utilisées n'ont pas été vues en cours, ce qui signifie que bien souvent nous ne savions pas que le problème pouvait être résolu par l'utilisation d'une fonction et surtout nous ne savions ni laquelle utiliser, ni comment l'utiliser. Nous avons par exemple mis du tout à trouver la fonction strcpy et essayions avant cela de changer le texte de fonction avec des "=" comme pour les entiers.

### **Ce qui est réussi :**

Ce projet est un très bon moyen de progresser en langage C et de réviser indirectement pour les DST, il nous fait utiliser tous types de fonctions et d'algorithmes et nous en fait découvrir des bien utiles. Nous avons gagné en aisance en C grâce à ce projet, en grande partie car il nous a pris beaucoup de temps et que donc nous avons passé beaucoup de temps à coder.

### **Ce qui peut être amélioré :**

Comme dit plus tôt, la majeure partie des fonctions utilisées n'ont pas été vues en cours, nous avons l'impression durant tout le projet d'être en retard sur le programme quand bien même nous n'avions pas de difficultés en cours. Certaines fonctions ont été très brièvement abordées en amphithéâtre, mais ce n'est pas suffisant pour correctement s'en souvenir et savoir quand et comment l'utiliser. Nous trouvons également ce projet trop long pour toute personne n'ayant jamais fait de programmation avant de venir à l'IUT, nous y avons passé plusieurs dizaines d'heures au total.



## Conclusion :

Ce projet était intéressant et utile pour la familiarisation avec le langage C et pour les révisions pour les DST, mais était sûrement trop difficile par rapport à ce qui a été vu en cours, il manquait aussi sûrement quelques précisions dans les documents pdf donnés. Néanmoins le bilan de ce projet est plutôt positif car la charge de travail qu'il apportait était surmontable et il nous a sûrement beaucoup aidé à progresser en programmation.

## Sources :

Le cours d'initiation au développement, openclassroom, développez.net, KooR.fr, docs.microsoft.com, askcodez.com .