

Projet IAP

# Table des Matières :

- Présentation du Projet.....	3
- Organisations des tests et bilan de validation.....	5
- Bilan De Projet.....	6
-Annexe.....	6

## Présentation du Projet :

Le projet consiste à afficher le résultat des joueuses d'un tournoi pour 10 tournois différents. Pour cela, on a inclus différentes fonctions dans un programme principal afin de donner des résultats.

***definir\_nombre\_tournois()*** est une fonction permettant d'inscrire le nombre de tournois maximum qu'on veut enregistrer. Il prend le tableau ***listeTournois*** comme [in-out].

***creationJoueuse()***, ***rechercheIndexJoueuse()*** et ***assignationPointsJoueuses()*** sont des fonctions secondaires (c'est-à-dire des fonctions supplémentaires à celles demandés) de la fonction ***enregistrement\_tournoi()***.

Elles ont respectivement comme but : la création des joueuses dans ***listeTournoi***, la recherche des index des joueuses déjà créées, ainsi que l'attribution des points en fonction de leur classement.

Elles prennent respectivement : ***listeTournois*** en [in-out], ***listeTournois*** en [in] et l'index de la joueuse recherché en [out], ***listeTournois*** en [in-out].

NB : La fonction ***creationJoueuse()*** « crée » de nouvelles joueuses même si elles sont déjà enregistrées dans un autre tournoi.

***affichageTypeMatch()*** est une fonction secondaire de la fonction ***affichage\_matches\_tournoi()***, qui affiche les matchs demandés par ***affichageTypeMatch()***, mais aussi les informations du type « 64<sup>e</sup> de finale », etc.

Elle prend comme [in] l'index du tournoi demandé ainsi que ***listeTournois***.

***affichageMatchJoueuse()*** affiche l'ensemble des matchs auquel a participé une joueuse au tournoi demandé par ***afficher\_matches\_joueuse()***.

Cette fonction prend comme [in] le nom de la joueuse demandée, l'index du tournoi demandé, ainsi que ***listeTournois***.

***triParSelectionOrdreLexicographique()*** trie le tableau de joueuse fourni par ***affichage\_joueuses\_tournoi()*** par ordre lexicographique.

Elle prend en [in-out] ledit tableau de joueuse, préalablement copié de ***dataJoueuse*** par ***affichage\_joueuses\_tournoi()***.

***copieTableauJoueuse()*** copie un nombre déterminé de joueuses (demandé par ***afficher\_classement()***) de ***dataJoueuse***, en prenant soin d'éliminer les éventuels doublons.

Elle prend en [in] le nombre de joueuses, l'index à partir duquel on commence à prendre les joueuses de ***dataJoueuse***, et ***listeTournois***, et en [in-out] le tableau de joueuse copié.

***triParSelectionClassement()*** à trier par nombre de points, puis par ordre lexicographique le tableau de joueuses fourni par ***afficher\_classement()***.

Cette fonction prend en [in] l'index de la joueuse à trier et la longueur du tableau occupé par les joueuses.

Elle prend en [in-out] le tableau de joueuse à trier.

### **Organisation des tests et bilan de validité :**

Nous nous sommes limités aux fichiers textes fournis par les professeurs. Nous nous sommes assurés d'avoir mélangé les instructions de ces fichiers fournis par divers « copiés-collés » (fichiers *in* et *out* de chaque sprint). Cela inclut les lignes de commandes et les noms de Joueuses. Nous avons fait 3 tests consécutifs de ces fichiers « aléatoires » en plus des tests donnés par défaut.

## **Bilan :**

Nous avons eu d'énormes difficultés dans la fonction ***enregistrement\_tournois()***. Nous avons eu des problèmes au niveau des validations des sprints à partir du sprint 3. En effet, à cause des différents copiés-collés, il était difficile de prévoir les fichiers out.txt correspondants, notamment à cause des points de joueuses associées à l'ordre des matchs.

Nous avons aussi eu des problèmes de logiciel pour le projet. À la base, nous utilisons **Visual Studio Code**, mais lors du jour de la validité, on nous a imposé de créer le .exe sur **Visual Studio 2019**. À cause de ce changement, nous n'avions pu prévoir les subtilités supplémentaires de ce logiciel (devoir désactiver un warning, réorganiser les fonctions dans un certain ordre...), et le programme a finalement crash pendant la recette.

Nous sommes néanmoins satisfaits de notre travail, car au moment où nous écrivons ces lignes, nous pensons être prêt à valider le Sprint 5.

## **Annexe :**

Les fichiers annexes sont composés des fichiers Sprint*n*.c, correspondant au code du Sprint n et des précédents.

Le fichier run.txt correspond à la trace du code du Sprint5.exe passé l'aide du inSp5.txt.