
Travaux Pratiques
STPI 2A – Algorithmique et Programmation 3
Sujet n°2

Dans ce sujet vous allez simuler le fonctionnement d'un module de gestion des meilleurs scores obtenus à un jeu. Les différents scores obtenus par les joueurs sont classés dans une liste chaînée du meilleur au plus faible.

Le squelette du programme principal vous est donné dans le projet *sujet2.cbp*. Afin de pouvoir implémenter les listes chaînées, vous ajouterez au projet les fichiers *listes.h* et *listes.c* que vous avez brillamment débuggés dans le sujet précédent !

Partie 1 : gestion des scores seuls

Dans cette première partie, seuls les scores seront utilisés.
Vous ne vous servirez pas de la variable contenant les noms des joueurs.

1) Affichage des scores

Créer une fonction *afficheNvaleurs(...)* permettant d'afficher les *n* premiers éléments d'une liste passée en paramètre.

Cette fonction fera partie de *main.c* et ne devra en aucun cas lire directement la liste chaînée. Elle ne pourra qu'utiliser les fonctions fournies avec la structure de données.

Une fois cette fonction terminée, vous pourrez implémenter les cas 1, 2 et 3 du *main*.

Attention : la fonction ne doit pas planter si on demande le top 10 alors qu'il n'y a pas autant de scores rentrés !

2) insertion des nouveaux scores.

Vous allez créer une fonction *insertScore(...)*.

Celle-ci vous permettra d'insérer un score passé en paramètre au bon endroit dans la liste chaînée (passée également en paramètre).

Vous n'utiliserez que les fonctions fournies avec la structure de données.

Une fois cette fonction réalisée, son appel sera réalisé dans le cas 0, juste après la simulation du jeu. La variable *score* vous donnera le score à insérer.

Vous ne tiendrez pas compte de la variable *nom*.

3) Suppression de tous les scores.

Gérez le cas 4 où l'utilisateur souhaite effacer les scores.

Partie 2 : Gestion des scores et des noms associés

Maintenant vous prendrez en compte les noms des joueurs.
La structure de cellule va donc être modifiée de la sorte :

```
struct cellule
{
    Element valeur; //score du joueur
    char nom[255]; //nom du joueur
    struct cellule* suivant; //adresse de la cellule suivante
};
```

1) Mise à jour de la structure de données

Modifiez les fonctions de la structure de données de manière à ce qu'elle tiennent compte de l'ajout du nom à la cellule.
Attention, faites vos modifications de manière à ce que votre programme principal soit le moins impacté possible !

2) accesseur pour le nom d'un joueur

Ajoutez une fonction *getNom(...)* à la structure de données. Celle-ci renverra le nom contenu dans une cellule dont la position sera passée en paramètre.

3) Affichage du meilleur score d'un joueur

Créer une fonction *bestScoreJoueur(...)* affichant le meilleur score obtenu par un joueur dont le nom est passé en paramètre.
Vous pourrez alors gérer le cas 5 en demandant au préalable quel joueur est recherché.

4) Affichage de tous les scores d'un joueur

Créer une fonction *allScoreJoueur(...)* affichant tous les scores obtenus par un joueur dont le nom est passé en paramètre.
Vous pourrez alors gérer le cas 6 en demandant au préalable quel joueur est recherché.

5) Affichage des scores cumulés

Vous allez gérer le cas 7 où l'on souhaite afficher un nouveau classement composé des scores **cumulés** de chaque joueur.
Pour cela vous créerez une fonction *createListeCumul(...)* qui devra créer une nouvelle liste chaînée. Chaque cellule de cette dernière comprendra le nom d'un joueur ainsi que son score cumulé (= la somme de tous ses scores dans la liste de départ). Les joueurs seront bien évidemment classés par ordre décroissant de score. Une fois la fonctions réalisée, vous pourrez afficher la liste ainsi créée.