

# NF16 - TP 3 – Liste chaînée

---

## *Gestion d'une ludothèque*

### Introduction

Nous proposons d'écrire un programme permettant la gestion d'une ludothèque (collection de jeux de société).

Pour cela, nous considérons qu'un jeu de société peut être décrit par une fiche contenant les informations suivantes :

- Nom du jeu
- Genre de jeu (un et un seul parmi jeu de PLATEAU, RPG, COOPERATIF, AMBIANCE, HASARD) ;
- Un entier indiquant le nombre de joueurs minimum ;
- Un entier indiquant le nombre de joueurs maximum ;
- Un entier indiquant la durée moyenne d'une partie en minutes ;

### A. Structure de données

Afin d'utiliser une liste chaînée pour gérer une ludothèque, nous proposons la structure **jeu** avec les champs suivants :

- **nom** de type chaîne de caractères ;
  - **genre** une énumération de type **genre\_jeu** ;
  - **nbJoueurMin** de type **int** ;
  - **nbJoueurMax** de type **int** ;
  - **duree** de type **int** ;
  - **suivant** de type **struct \* jeu**
1. Définir le type d'énumération **genre\_jeu**.
  2. Définir la structure **jeu**.
  3. Définir le type **t\_jeu** à partir de la structure **jeu**.
  4. Définir la structure **ludotheque** comme une structure contenant 2 champs :
    - **nb\_jeu** : un entier indiquant le nombre de jeux dans une ludothèque
    - **debut** : un pointeur sur un **t\_jeu** indiquant l'adresse du premier jeu de la collection (NULL si la collection est vide)
  5. Définir le type **t\_ludotheque** à partir de la structure **ludotheque**

### B. Fonction à écrire

1. Une fonction qui crée une ludothèque vide (i.e. **nb\_jeu** = 0 et **debut** = NULL) et qui renvoie l'adresse de cette ludothèque vide (un pointeur sur le type **t\_ludotheque**) si la création est réussie et qui renvoie NULL sinon :

**t\_ludotheque \* creer\_ludotheque()**

2. Une fonction pour créer un jeu en initialisant ses champs :

```
t_jeu * creer_jeu(char * nom, enum genre_jeu genre, int  
nbJoueurMin, int nbJoueurMax, int duree)
```

3. Une fonction pour ajouter un jeu dans une ludothèque en respectant l'ordre alphabétique des noms. Cette fonction renvoie 1 si le jeu a bien été ajouté, 0 sinon :

```
int ajouter_jeu(t_ludotheque* ludo, t_jeu* j)
```

4. Une fonction affichant le contenu de la ludothèque comme suit :

Nom	Type	Nombre de joueurs	Durée moyenne
Time's up	Ambiance	4-8	35'
Endor	Rôle	3-8	90'
Total	2		

```
void affiche_ludotheque(t_ludotheque * ludo) ;
```

5. Une fonction pour retirer un jeu de la base à partir de son nom. La fonction renvoie 1 si le retrait a réussi, 0 sinon :

```
int retirer_jeu(t_ludotheque* ludo, char* nom) ;
```

Vous veillerez à prendre en compte le cas où la ludothèque est vide ou si elle ne contient qu'un seul élément.

6. Une fonction pour détruire une ludothèque et supprimer tous les jeux :

```
void supprimer_ludotheque(t_ludotheque * ludo)
```

7. Une fonction permettant d'interroger la ludothèque pour suggérer des jeux répondant à certains critères :

- Genre du jeu (enum genre\_jeu)
- Nombre de joueur (int) : un jeu répondant à ce critère est jouable avec le nombre de joueur demandé ;
- Durée idéale en minute (int) : un jeu répondant à ce critère a une durée moyenne de jeu égale à la durée demandée  $\pm 10\%$

Un critère peut être ignoré si celui-ci est fixé à -1.

La fonction renvoie une nouvelle **t\_ludotheque** contenant les résultats ;

```
t_ludotheque * requete_jeu(t_ludotheque * ludo, enum genre_jeu  
genre, int nbJoueurs, int duree) ;
```

8. Créer une fonction permettant la fusion de deux ludothèques :

```
t_ludotheque * fusion(t_ludotheque * ludo1, t_ludotheque * ludo2) ;
```

- Le résultat de la fusion est contenu dans une nouvelle liste ;
- La nouvelle ludothèque doit être encore ordonnée alphabétiquement ;
- La nouvelle ludothèque ne doit pas contenir de doublons.

-

### **C. Programme Principal :**

Programmer un menu pour gérer une ludothèque. Celui-ci doit incorporer les fonctionnalités suivantes :

1. Créer une ludothèque
2. Afficher une ludothèque
3. Ajouter un jeu dans la ludothèque en saisissant ses caractéristiques ;
4. Effectuer une recherche de jeu à partir de critères saisis par l'utilisateur ;
5. Créer 2 ludothèques, les afficher, les fusionner puis afficher la nouvelle ludothèque ;
6. Quitter.

### **Consignes générales :**

A la fin du programme, la mémoire allouée dynamiquement doit être libérée.

L'organisation minimale du projet est la suivante :

- Fichier d'entête tp3.h contenant les déclarations des structures et fonctions de bases ;
- Fichier source tp3.c contenant la définition de chaque fonction ;
- Fichier source main.c contenant le programme principal.