

# Programmation C

## TP 13 - Gestion des fichiers

# I - Objectifs

Nous allons créer **un système de Hight Score de jeu vidéo**. L'objectif est de lister les 5 meilleurs score réalisés sur le jeu ainsi que les noms des personnes qui les ont réalisées. Il faut alors enregistrer cette liste dans un fichier pour la rendre persistante au redémarrage du jeu. le fichier HightScore, sera constitué d'une liste de nom et score rangés dans l'ordre décroissant de score.

## Hight Score

Classement	Nom	Score
1	Toto	65234
2	Lydie	63221
3	John	59321
4	IKO	35648
5	---	0

*Résultat final*

Toto 65234  
Lydie 63221  
John 59321  
IKO 35648

*contenue de HightScore.txt*

## II - Préparatifs

1- Créer un nouveau projet C du nom de **"tp13\_gestion\_des\_fichiers"**.

2- Remplacer le code du fichier main.c par:

```
#define TAILLE_LISTE 5
#define TAILLE_MAX_NOM 10

typedef struct HIGHT_SCORE
{
    char nom[TAILLE_MAX_NOM];
    long score;
}hightScore;

int main()
{
    char* path = "datas/HightScore.txt";
    hightScore scores[TAILLE_LISTE];

    return 0;
}
```

# III - Instructions

- 1- Gérer l'ouverture du fichier 'HightScore.txt'
- 2- Parcourir le fichier 'HightScore.txt' pour remplir le tableau 'scores'
- 3- Afficher le tableau des high score

Hight Score		
Classement	Nom	Score
-----		
1	Toto	65234
2	Lydie	63221
3	John	59321
4	IKO	35648
5	---	0

*Résultat final*

Toto 65234  
Lydie 63221  
John 59321  
IKO 35648

*contenue de HightScore.txt*

## IV - Pour aller plus loin

A - Créer une fonction d'ajout d'un nouveau high score. Attention à conserver un classement des scores

B - Enregistrer les modifications du tableau des high score dans le fichier 'HightScore.txt'

# V - SPOIL! (aide)

1- Ouvrir le fichier en mode “r+” pour prévoir la suite (bonus)

2- **int loadHightScore(FILE\* pt\_fichier, hightScore\* scores, int taille)**: Il faut utiliser la fonction fscanf pour lire à chaque fois le nom et le score. Puis il faut enregistrer ces informations dans le tableau de structure ‘scores’. Attention à gérer le cas où notre fichier à moins de 5 lignes. Dans ce cas remplir la structure avec des valeurs en dure “---” et 0;

3- **void afficherHightScore(hightScore\* scores, int taille)** : Une simple boucle for pour parcourir le tableau ‘scores’ et afficher son contenu avec des printf.

A- **addHightScore(hightScore\* scores, int taille, hightScore nouveau)**: Il faut faire l’ajout dans ‘scores’ tout en conservant un classement des scores. Pensez à découper le problèmes en sous problèmes.

- gérer le cas où le nouveau score est inférieur au moins bon Hight score. Dans ce cas, rien à faire.
- gérer le cas où c’est le meilleur score
- gérer les autres cas

B- **saveHightScore(FILE\* pt\_fichier, hightScore\* scores, int taille)**: Utiliser la fonction fprintf tout en parcourant ‘scores’ pour réécrire l’ensemble du fichier. Utiliser la fonction rewind() pour se positionner en début de fichier avant de faire l’écriture.