

Objectif : Allocation dynamique de la mémoire par manipulation de SD complexes (listes, tableaux de listes)

1. Gestion des adhérents d'une association

Une **association** scientifique désire automatiser la gestion de son fichier des "adhérents". Chaque nouvel adhérent remplit une fiche de renseignements ayant la structure suivante :

```
code :      4 caractères (unique et généré automatiquement à la création)
nom  :      20 caractères (nom et prénom)
ville :      15 caractères
domaine d'intérêt : 2 caractères (codé comme suit
                    01: Informatique 02: Automatique 03: Physique ...etc.)
```

Nous allons faire cela en deux étapes : d'abord nous ne considérons qu'une seule ville, ensuite plusieurs villes. Nous devons passer d'une étape à une autre sans modifications majeures (bonnes maintenance). Au niveau développement nous devons respecter quelques règles de bonnes pratiques.

2. Adhérents d'une seule ville (Rabat)

Les adhérents sont supposés tous de Rabat et gérés en mémoire dans une **liste chaînée**. Ils sont classés par ordre alphabétique des **noms**.

Écrire un programme de test en C **affichant un menu()** et permettant au choix :

- d'**insérer** un adhérent dans la structure (liste),
- d'**afficher()** les adhérents de la ville
- de **supprimer()** un adhérent de la structure
- de **quitter()** le programme par **ESC**.

3. Adhérents de plusieurs villes (Maroc)

Nous considérons maintenant les adhérents de **plusieurs villes** du Maroc

Compléter le code de l'exercice 1 pour qu'il soit possible

- d'**afficher()** les adhérents d'une ville donnée et leur nombre
- de **rechercher()** un adhérent dans la structure par son *code* ou son *nom*

Afin de pouvoir éditer, d'une manière performante, la liste de tous les adhérents d'une ville, les données de l'association sont structurées sous forme d'un **tableau de listes** :

