

## Applications

### A Un Petit Annuaire

Pour écrire un programme de gestion d'annuaire, on utilise la structure suivante :

```
1 typedef struct Fiche * Fiche;
2 struct Fiche {
3     char * nom;
4     char * prenom;
5     char tel[10];
6 };
```

A.1 Ecrire les fonctions `void affiche(Fiche)` et `Fiche saisieFiche(void)` d'affichage et de saisie. (*Indication* Pensez à utiliser votre `libchaine`.)

A.a **En Tableaux** Pour commencer, le type `Annuaire` est un tableau de `Fiche` de taille `MAXFICHES`.

```
1 #define MAXFICHES 1000
2 typedef Fiche AnnuaireT[MAXFICHES];
```

A.2 Ecrire la fonction `void afficheAnnuaireT(AnnuaireT)` qui affiche l'ensemble des fiches. Comment gérer efficacement les annuaires de taille inférieure à `MAXFICHES` ?

A.3 Ecrire la fonction `Fiche rechParNomT(AnnuaireT annuaire, char * nom)` qui renvoie la première fiche de l'annuaire répondant au critère de `nom`.

(Facultatif) Ecrire la fonction `Fiche rechParTel(Annuaire, char[10])` qui effectue la même recherche mais cette fois par numéro de téléphone. TP7

#### A.b En Listes Chaînées

Pour pouvoir manipuler un nombre a priori non borné de fiches, on définit la structure suivante :

```
1 typedef struct FicheChaine * FicheChaine;
2 typedef FicheChaine AnnuaireL ;
3
4 struct FicheChaine {
5     Fiche fiche;
6     FicheChaine suivante;
7 };
```

A.4 Ecrire les fonctions `void afficheChaine(FicheChaine)` et `FicheChaine saisieFicheChaine(void)` d'affichage et de saisie.

A.5 Ecrire une fonction `AnnuaireL convertit(AnnuaireT)` qui convertit un annuaire implémenté avec un tableau en un annuaire implémenté par liste chaînée.

A.6 Ecrire les fonctions de recherche `AnnuaireL rechParNom(AnnuaireL, char *)` et facultativement `AnnuaireL rechParTel(AnnuaireL, char[10])` qui renvoient l'ensemble des fiches correspondant à la recherche.

On cherche désormais à imposer un “ordre” sur la liste chaînée.

A.7 Ecrire une fonction `int compare(FicheChaine, FicheChaine)` qui compare deux fiches par ordres alphabétiques des noms, des prénoms, puis du numéro de téléphone. Cette fonction renvoie un entier strictement négatif, nul, ou strictement positif selon le résultat de la comparaison.

A.8 En utilisant la fonction précédente écrire une fonction de tri en place de la structure de données `void trier(AnnuaireL)`. On utilisera un classique tri par comparaison.