

## R3.02 Développement efficace

### TP2 : les listes

#### Exercice 1 : listes chaînées

On souhaite manipuler une liste de personnes (nom et âge) et pour cela écrire avec tous les commentaires nécessaires :

1. les définitions des types `tpersonne`, élément `telement` et liste `tliste` ;
2. la procédure d'initialisation d'une `tliste l` ;
3. la procédure d'insertion d'une nouvelle personne (nom et âge) en tête d'une `tliste l` ;
4. la procédure d'affichage à l'écran des éléments d'une `tliste l` (pensez à définir une procédure pour afficher une `tpersonne`) ;

Testez les trois procédures précédentes en demandant des informations de personnes (nom et âge) au clavier, personnes que l'on insérera successivement (on arrête la saisie lorsque le nom de la personne est la chaîne "-1").

Testez dans un main les fonctions suivantes :

5. La fonction de recherche d'une personne à partir de son nom dans une `tliste l`, retournant l'adresse de l'élément où elle se trouve dans la liste si la personne est présente, `NULL` sinon .
6. La fonction supprimant le `telement` en tête d'une `tliste l`.
7. Écrire une fonction permettant de compter le nombre de personnes d'un même nom dans une `tliste l`. Il sera judicieux d'utiliser la fonction de recherche de la 5e question.
8. Écrire une fonction permettant de rechercher le `ième` `telement` dans une `tliste l` et de retourner son adresse, ou `NULL` si la liste contient moins de `i` `telement`.
9. Une fonction pour insérer une personne en fin de liste.
10. Proposez une procédure pour insérer une personne après le `Iième` élément de la liste (on ne fait rien si la liste ne comporte pas de `Iième` élément).

#### Exercice 2 : listes doublement chaînées

Soit le type `tlisteDC` permettant de définir une liste d'entiers doublement chaînée :

```
typedef struct elem {
    struct elem* prec;
    int val;
    struct elem* svt;
} telement;
typedef telement* tlisteDC;
```

Un `telement` d'une `tlisteDC` contient un pointeur `prec` vers la structure précédente, une valeur `val`, et un pointeur `svt` vers la structure suivante. Le premier `telement` a une valeur de `prec` égale à `NULL`, de même que le dernier `telement` a une valeur de `svt` égale à `NULL`.

1. Faites un schéma d'une telle liste avec quelques éléments
2. Proposer une procédure pour initialiser une `tlisteDC lDC`.
3. Proposer une procédure pour insérer un entier en tête d'une `tlisteDC lDC`.
4. Proposez une procédure pour afficher une telle liste.
5. Proposez une procédure pour insérer un entier après le `I`ième élément de la liste (on ne fait rien si la liste ne comporte pas de `I`ième élément).

-----