

TP 6 : La Récursivité

Dans ce TP, on reprend le TP 4 mais en implémentant les méthodes d'une manière récursive.

On définit alors la liste chaînée en utilisant la structure suivante :

```
typedef struct noeud_S *pNoeud;  
typedef struct noeud_S {  
    int info;  
    pNoeud lien;  
} liste;
```

1. Ecrire une fonction qui permet d'afficher une liste.

Son prototype est : **void afficherR** (pNoeud l).

2. Ecrire une fonction qui retourne la longueur d'une liste.

Son prototype est : **int longueurR**(pNoeud l).

3. Ecrire une fonction qui permet de calculer le nombre d'occurrences d'une donnée dans une liste chaînée. Son prototype est : **int nbrOccuR**(pNoeud l, **int** donnee)

4. Ecrire une fonction qui permet d'ajouter un élément à la fin d'une liste.

Son prototype est : **pNoeud ajoutfinR**(pNoeud l, **int** donnee).

5. Ecrire une fonction qui permet de supprimer le dernier élément de la liste.

Son prototype est : **pNoeud supprimerfinR**(pNoeud l).

6. Ecrire une fonction qui permet de supprimer l'élément de la position i.

Son prototype est : **pNoeud concatR**(pNoeud l1, pNoeud l2).

7. Ecrire une fonction qui permet d'inverser une liste.

Son prototype est : **pNoeud inverserListeR**(pNoeud l).