

TP n° 3 : Représentation des arbres n-aires par liste des fils**A réaliser dans Hop3x dans l'activité TP3-TDA****Objectifs :**

- Entretenir votre agilité dans la manipulation des pointeurs en C.
- Vous préparer pour les éliminatoires de la coupe du monde des parcours d'arbres.
- Vérifier que vous savez conserver des objets en stockage indirect dans des conteneurs (ici des arbres) sans provoquer de catastrophes en mémoire dynamique.

Sujet : (Examen de L2 Décembre 1995, Exercice 2 sur 10 points, 1 h 30)

Un arbre (sous-entendu général) est caractérisé par l'étiquette de sa racine et par la liste de ses sous-arbres.

Remarque : un arbre n'est jamais vide.

On considère les primitives suivantes sur les arbres :

```
/* primitives de construction- destruction */
ARBRE ArbreConstruire(ELEMENT e, LISTEDARBRES L) ;
/* retourne un arbre dont la racine a pour etiquette e et les fils sont dans la liste l */
void ArbreDetruire(ARBRE A) ;
/* libère la mémoire allouée pour l'arbre A */

/* primitives de consultation */
ARBRE PremierFils(ARBRE A) ;
/* retourne le premier fils de A si A n'est pas une feuille */
ARBRE FilsSuivant(ARBRE F, ARBRE A) ;
/* retourne le fils de A suivant de F si F n'est pas le dernier */
ELEMENT Etiquette(ARBRE A) ;
/* retourne l'étiquette de A */

/* primitives de test */
bool EstFeuille(ARBRE A) ;
/* retourne vrai si A est une feuille (liste des fils est vide) */
bool DernierFils (ARBRE F, ARBRE A) ;
/* retourne vrai si F est le dernier fils de A */
```

En utilisant les primitives du TDA ELEMENT

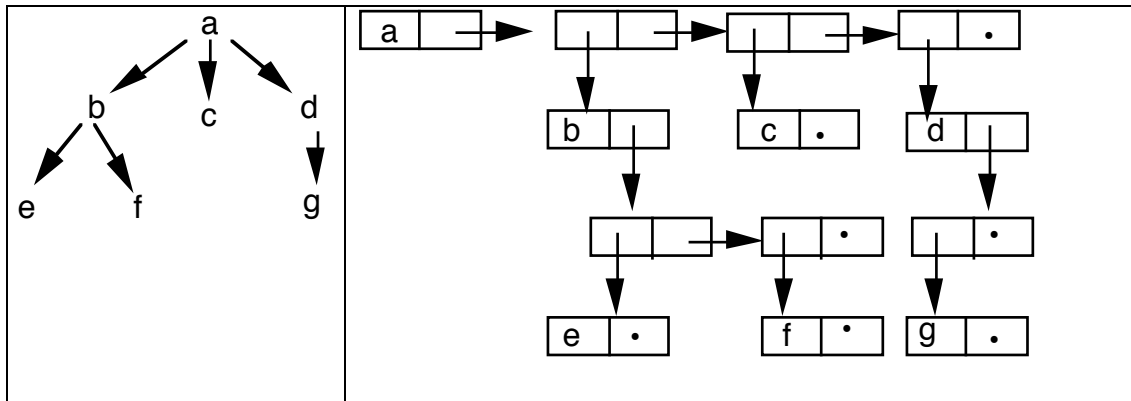
1) Donner le code définissant les primitives précédentes en adoptant les structures de données suivantes :

- un arbre est un pointeur sur une structure formée d'une étiquette et de la liste de ses fils
- une listes d'arbres est un pointeur sur une cellule composée d'un arbre et d'un pointeur sur la cellule suivante.

```
typedef struct cell {
    struct arb *      fils ;
    struct cell *     suivant ;
} * LISTEDARBRES ;

typedef struct arb {
    ELEMENT          etiquette ;
    LISTEDARBRES     listeDesFils ;
} * ARBRE ;
```

Exemple :



2) A l'aide des primitives de la question 1 et de celles du TDA ELEMENT, donner le code définissant les opérations suivantes :

- AffichePref** qui affiche l'arbre par ordre préfixe
- Lire** qui retourne un arbre après avoir demandé les étiquettes de sa racine et de ses fils
- Retourne le père d'un nœud donné dans un arbre donné
- Affiche l'arbre par niveau
- Affiche les feuilles de l'arbre

3) Tester ces fonctions en réalisant le TDA élément avec des chaînes de caractères