

# TP de programmation fonctionnelle et logique

## TP 2 : arbres n-aires et fonctionnelles

### Arbres n-aires

Un arbre n-aire est soit un nœud interne (composé d'un entier, la clef, et d'une liste de sous-arbres), soit une feuille, c'est-à-dire un nœud sans fils. Nous définissons un arbre n-aire par le type suivant :

```
#type arbre = Noeud of (int * arbre list);;
type arbre = Noeud of (int * arbre list)
```

### Fonctions élémentaires sur les arbres

1. Écrire une fonction `make_arbre` qui construit un arbre à partir d'un entier et d'une liste de sous-arbres.
2. Écrire des fonctions `clé` et `fils` qui extraient respectivement d'un nœud interne sa clé et la liste de ses fils.
3. Définissez une fonction `feuille` qui renvoie le booléen `true` si et seulement si son argument est une feuille.

### Calcul sur les arbres

1. Définissez une fonction qui calcule la hauteur d'un arbre.
2. Construisez les fonctions `nb_feuilles` et `nb_noeuds` qui calculent respectivement le nombre de feuilles et de nœuds (nœuds internes et feuilles) d'un arbre.

### Parcours dans les arbres

1. Écrivez une fonction `parcours_profondeur` qui énumère les clefs d'un arbre en profondeur d'abord.
2. Écrivez une fonction `appartient` qui renvoie `true` si et seulement si l'entier argument est contenu dans l'arbre étudié.
3. Écrivez une fonction `arbres_égaux` qui teste l'égalité de deux arbres.
4. Écrivez une fonction `parcours_largeur` qui énumère les clefs d'un arbre en largeur d'abord.