

## TP n° 2 : Manipulation de listes

**1 Parcours de listes**

- conc3(L1,L2,L3,L)** L est la concaténation des listes L1, L2, L3.
- membre(E,L)** E est élément de la liste L.
- compte(E,L,N)** N est le nombre d'occurrences de l'élément E dans la liste L.
- renverser(L,R)** R est la liste L à l'envers.
- palind(+L)** L est une liste "palindrome".
- nieme(+N,+L,-E)** E est l'élément de rang N de la liste L.
- rang(-N,+L,+E)** E est l'élément de rang N de la liste L.
- horsde(E,L)** E n'est pas élément de la liste L.
- tousdiff(L)** les éléments de L sont tous différents.
- debpar(L,D)** la liste L commence par la liste D.
- sousliste(L,S)** la liste S est sous-liste de la liste L.
- elim(+L1,-L2)** la liste L2 contient une seule occurrence de chaque élément de la liste L1.

**2 Parcours d'arbre**

*Dans cette question, les termes manipulés sont des arbres. Les arbres seront représentés par des termes récursifs, à l'aide du constructeur binaire **a**, de la façon suivante : **a(Noeud, Listedesousarbres)**. Les feuilles de l'arbre ont le même format. Elles sont donc de la forme **a(Noeud, [])**.*

*Exemple d'arbre : **a(donald, [a(riri, []), a(fifi, []), a(loulou, [])])***

- aplatir(+A,-L)** L est la liste des noeuds de l'arbre A, en suivant un parcours gauche-droite, en profondeur d'abord.

### 3 Parcours d'ensembles

*Dans les 4 questions suivantes, les listes Prolog représentent des ensembles. Nous admettons donc qu'il n'y a jamais deux occurrences d'un même élément dans ces listes.*

**sousensemble(S,E)** tous les éléments de la liste S sont présents dans la liste E.

**union(E1,E2,U)** U est l'union ensembliste des listes E1 et E2.

**inter(E1,E2,I)** I est l'intersection ensembliste des listes E1 et E2.

**diffe(E1,E2,D)** D est la différence ensembliste des listes E1 et E2.

### 4 Questions complémentaires sur les listes

**tri(L,T)** la liste T est le résultat du tri par ordre croissant de la liste d'entiers L.

**permutation(L,P)** P est une liste obtenue par permutation quelconque des éléments de L.

**lire(-L)** L est la liste des caractères entrés au clavier, amputée du marqueur de fin ".". Les caractères d'espacement seront ignorés (utiliser le prédicat **get(C)**, qui lit un caractère au clavier sans tenir compte des blancs ni des retour à la ligne. **C** vaudra le code Ascii du caractère. Ainsi, si on tape un "A" (resp. un "."), **C** vaudra **65** (resp. **46**), qu'on peut également noter **0'A** (resp. **0'**).