# IA02 – TD2 : Listes

**Ecrire les prédicats prolog suivants :**

1- `tete(T,L)` qui unifie la variable T avec la tête de la liste L.

2- `queue(Q,L)` qui unifie la variable Q avec la queue de la liste L.

3- `vide(L)` qui s'efface si la liste L est vide.

4- `imprime(L)` qui affiche le contenu de la liste L.

5- `element(X,L)` qui s'efface si X est présent dans la liste L.

6- `dernier(X,L)` qui unifie X avec le dernier élément de la liste L.

7- `longueur(Long,L)` qui unifie Long avec la longueur de la liste L.

8- `nombre(X,L,NB)` qui compte le nombre de fois où X apparaît dans L et renvoie le résultat dans `NB`.

9- `concat(L1,L2,L3)` qui effectue la concaténation de la liste L1 avec la liste L2 et la range dans L3.

10- `inverse(L,R)` qui inverse la liste L dans R.

11- On veut implémenter un tri sous prolog. On suppose que L est une liste d'entiers à trier. Soit `X` un élément de `L`. On appelle `L1={ Y∈L\X` tel que `Y<=X }` et `L2={ Y∈L\X` tel que `Y>X }`. Alors la liste `L` triée est égale à : [ liste `L1` triée, `X` , liste `L2` triée].
   a) Implémenter `partition(X,L,L1,L2)` qui place dans `L1` les éléments de `L` qui sont inférieurs ou égaux à `X`, et dans `L2` les éléments de `L` qui sont strictement supérieurs à `X`.
   b) Implémenter `tri(L,R)` qui tri la liste `L` et place le résultat dans `R`.

12- `sous_liste(L1,L2)` qui vérifie que `L1` est une sous liste de `L2`.

13- `retire_element(X,L,R)` qui retire la première occurrence de l'élément `X` dans `L` et place le résultat dans `R`.

14- `retire_elements(X,L,R)` qui retire toutes les occurrences de `X` dans `L` et place le résultat dans `R`.

15- `retire_doublons(L,E)` qui transforme la liste `L` en un ensemble `E` (sans redondance).

16- `union(E1,E2,E)` qui effectue l'union de l'ensemble `E1` avec l'ensemble `E2` et place le résultat dans `E`.

17- `intersection(E1,E2,E)` qui effectue l'intersection de l'ensemble `E1` avec l'ensemble `E2` et place le résultat dans `E`.