

Prolog & IA *Série 3*

Exercice 1

Définir les procédures suivantes :

<code>nieme(?N, ?X, +L)</code>	Vraie si X est le N-ieme élément de la liste L.
<code>nb_occur(?X, ?N, +L)</code>	N est le nombre d'occurrence de X dans L.
<code>creerListe(?X, +N, -L)</code>	L est la liste créée à partir de N occurrences du terme X.
<code>supprimer(?X, +L1, ?L2)</code>	Supprime toutes les occurrences de X de L1 pour obtenir L2.
<code>inverser(+L1, ?L2)</code>	L2 est la liste inverse de L1.
<code>substituer(?X, ?Y, +L1, ?L2)</code>	Substituer toutes les occurrences de X par Y dans L1 pour obtenir L2.
<code>decaler(+L, ?Ld)</code>	Ld est la liste L où les éléments ont été décalés circulairement vers la droite.
<code>permuter(+L, ?P)</code>	Qui donne une permutation P de la liste L.
<code>aplatir(+LL, ?L)</code>	LL est une liste de listes. L est la liste LL qui ne contient plus qu'un seul niveau de crochets.

Tester ces procédures avec les buts suivants :

<pre>?- nieme(N, X, [a,b,c]). N = 1, X = a ; N = 2, X = b ; N = 3, X = c ; false. ?- nb_occur(X, N, [a,b,a,c,b,a]). X = a, N = 3 ; X = b, N = 2 ; X = c, N = 1 ; N = 0 ; false. ?- supprimer(a, [a,b,a,c], L). L = [b, c] . ?- suppr(a, [a,b,a,c], L). L = [b, a, c]. ?- inverser([a,b,c], L). L = [c, b, a].</pre>	<pre>?- substituer(X,e,[a,b,a,c],L). X = a, L = [e, b, e, c] ; X = b, L = [a, e, a, c] ; X = c, L = [a, b, a, e] ; L = [a, b, a, c] ; false. ?- decaler([a,b,c],L). L = [c, a, b] ; false. ?- permuter([a,b,c],P). P = [a, b, c] ; P = [b, a, c] ; P = [b, c, a] ; P = [a, c, b] ; P = [c, a, b] ; P = [c, b, a] ; false. ?- aplatir([a,[b,[[]]],[c]],L). L = [a, b, c] ; false.</pre>
---	---

Exercice 2

Dans la suite, on notera par p un prédicat qui permet de définir un ordre, on prendra par exemple :

$p(X,Y) :- X @=< Y.$

1. Définir le prédicat :

`est_trie(+L).`

qui est vrai si la liste L est triée selon l'ordre donné par le prédicat p .

2. Ecrire la procédure :

`min_liste(-Min, +L).`

où Min est le plus petit des éléments de la liste L , selon l'ordre défini par p .

3. Ecrire les procédures correspondantes aux différentes méthodes de tri : tri par sélection, tri à bulles, tri insertion, tri fusion et tri rapide, qui permettent de trier une liste selon l'ordre défini par p .

Exercice 3

On représente les ensembles par des listes sans doublons. Ecrire les procédures suivantes :

<code>ensemble(+L)</code>	Vrai si L représente un ensemble
<code>creer_ensemble(+L, -E)</code>	E est l'ensemble créé à partir de la liste L .
<code>sous_ensemble(?SE, +E)</code>	SE est un sous-ensemble de E .
<code>intersection(+E1, +E2, -E)</code>	E est l'intersection de $E1$ et $E2$.
<code>reunion(+E1, +E2, -E)</code>	E est la réunion de $E1$ et $E2$.
<code>difference(+E1, +E2, -E)</code>	$E = E1 - E2$.

Exercice 4

On représente un vecteur par une liste et une matrice par la liste de ses vecteurs-lignes. Ecrire les procédures suivantes :

<code>est_matrice(+M, ?NL, ?NC)</code>	Vraie si M est une matrice $NL \times NC$.
<code>transposee(?M, ?TM)</code>	TM est la matrice transposée de la matrice M .
<code>pscalaire(+U, +V, -P)</code>	P est le produit scalaire de U et V .
<code>produit_mv(+M, +U, -V)</code>	V est le vecteur produit de la matrice M par le vecteur U .
<code>produit_m(M1, M2, M)</code>	M est le produit des deux matrices $M1$ et $M2$.

Exercice 5

Trois personnes, de nationalités différentes (marocaine, algérienne et tunisienne) et pratiquant des sports différents (football, boxe et tennis), habitent dans la même rue, dans trois maisons de couleurs distinctes (blanc, bleu, vert). Une des maisons est située au début de la rue, une autre au milieu, et la troisième au bout de la rue.

On dispose des 5 indices :

- Dans la maison verte on pratique la boxe.
- La maison verte est située avant la maison de l'algérien.
- Le marocain habite la maison blanche.
- La maison blanche est située avant la maison où on pratique le football.
- Le tennisman habite au début de la rue.

Ecrire en prologue une procédure qui permet de trouver les caractéristiques de chacune de ces 3 maisons.