

TP2 prolog L3 - INFO 2009-2010

En PROLOG

P Exemple : soit $L = [a,b,c]$.

P ?- $L = [Head|Tail]$ reussit avec $Head = a$ et $Tail = [b,c]$ et $L = .(a, Tail)$

P - $[a,b,c] = [a|[b,c]] = [a,b|[c]] = [a,b,c|[]]$

1. Ecrire un prédicat prolog qui est vrai si x est un élément de la liste L
2. premier(E,L) est vrai si E est le premier élément de L
3. Ecrire un prédicat Prolog der qui trouve le dernier élément d'une liste L
4. Ecrire un prédicat Prolog avDer qui trouve le l'avant-dernier élément d'une liste L
5. Ecrire un prédicat prolog qui supprime le K^{ième} élément d'une liste L1 et écrit le résultat dans une liste L2

Example:

?- sup_k(X,[a,b,c,d],2,R).

X = b

R = [a,c,d])

6. Définir le prédicat longueur(L,N), qui étant donnée la liste L calcule sa longueur N.
7. Définir le prédicat pair(L) qui est vrai si L a un nombre pairs d'éléments
8. Définir le prédicat concat(L1,L2,L3) où L3 est le résultat de la concaténation de L1 et L2 (sans utiliser append).
9. Définir le prédicat palindrome(L) vrai si la liste L est sa propre image renversée. (exemple: [x,a,m,a,x].)

Correction

10. `element(E,L)` est vrai si E appartient a L (non deterministe , si E et L instancie et si L contient plusieurs fois E)

```
element(E,[E|_]).
```

```
element(E,[_|T]) :- element(E,T).
```

11. `premier(E,L)` est vrai si E est le premier element de L

```
premier(X,[X|_]).
```

12. `dernier(E,L)` est vrai si E est le dernier element de L

```
dernier(X,[X]).
```

```
dernier(X,[_|L]) :- dernier(X,L).
```

13. `avdernier(X,L)`

```
avdernier(X,[X,_]).
```

```
avdernier(X,[_|Y|Ys]) :- avdernier(X,[Y|Ys]).
```

14. `sup_k(X,L,K,R)` :- X is the Kieme element de L; R est la liste L sans X.

```
sup_k(X,[X|Xs],1,Xs).
```

```
sup_k(X,[Y|Xs],K,[Y|Ys]) :- K > 1, K1 is K - 1, sup_k(X,Xs,K1,Ys).
```

15. `Longeur(L,N)`

```
longeur([], 0).
```

```
longeur( [_|T], N) :- longeur (T, N1), N is N1 + 1.
```

16. `pair(L)` est vrai si L a un nombre pairs d'elements

```
pair([]).
```

```
pair( [_|_|S]) :- pair(S).
```

17. `concat(L1,L2,L3)` est vrai si `L3` est la concatenation de `L1` et `L2`

```
concat([],L,L).
```

```
concat([H|T],L,[H|Tr]) :- concat(T,L,Tr).
```

18. `Palindrome(L)`

```
palindrome(L) :- miroir(L, L).
```

```
miroir(X, Y) :- transfere(X, [], Y).
```

```
transfere([], L, L).
```

```
transfere([H|T], L1, L2) :- transfere(T, [H|L1], L2).
```

```
/* palindrome(Xs) is true if Xs is a palindrome. */
```

```
/* e.g. palindrome([m,a,d,a,m, i,m, a,d,a,m]). */
```

```
palindrome([]).
```

```
palindrome([_]).
```

```
palindrome([X|Xs]) :- append(Xs1,[X],Xs), palindrome(Xs1).
```