



## TP1 PROLOG (PROgrammation LOGique)

---

1. Le site de SWI-Prolog 5.4 : <http://www.swi-prolog.org/>
2. Mes premiers programmes : Faites fonctionner les programmes ci-dessous pour vous familiariser avec l'environnement de Prolog

```
A) pere(charlie,david) .
   pere(henri,charlie) .
   grandpere(X,Y) :- pere(X,Z) , pere(Z,Y) .
```

Vous tapez ce programme dans un éditeur de texte simple et sauvegardez dans un fichier « prprog.pl » (par exemple, ou tout autre nom avec le suffixe .pl).

```
?- consult('c:/prprog.pl').
   ///** attention à respecter le « chemin » d'accès à votre source Prolog.***/
% c:/prprog.pl compiled 0.00 sec, 0 bytes
Yes
?- pere(X,Y).
X = charlie
Y = david ;
X = henri
Y = charlie ;
No
?- grandpere(X,Y).
X = henri
Y = david ;
No
?- grandpere(henri,X).
X = david ;
No
?-halt.
```

```
B) habite(jean,paris) .
   habite(hans,munich) .
   habite(juan,madrid) .
   capitale(paris) .
   capitale(madrid) .
   habite-capitale(Qui) :-
   habite(Qui,Qqpart) , capitale(Qqpart) .
```

```
Welcome to SWI-Prolog (Version 5.0.9)
Copyright (c) 1990-2002 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.
```

For help, use ?- help(Topic). or ?- apropos(Word).

```

?- consult('c:/dxprog.pl').
% c:/dxprog.pl compiled 0.00 sec, 0 bytes
Yes
?- habite(jean,paris).
Yes
?- habite(jean,Qqpart).
Qqpart = paris ;
No
?- habite-capitale(Qui).
Qui = jean ;
Qui = juan ;
No
?- habite(Qui,paris).
Qui = jean ;
No
?- habite(_,Qqpart).
Qqpart = paris ;
Qqpart = munich ;
Qqpart = madrid ;
No
?- habite(Qui,Qqpart),capitale(Qqpart).
Qui = jean
Qqpart = paris ;
Qui = juan
Qqpart = madrid ;
No
?-halt.

```

### C) Les Mutants

```

non_vider([_|_]).
mutant(S) :- animal(D), animal(F),
             append(Debut,Milieu,D),
             non_vider(Debut), non_vider(Milieu),
             append(Milieu,_,F),
             append(Debut,F,M),
             name(S,M).

animal("alligator").
animal("lapin").
animal("tortue").
animal("pintade").
animal("cheval").

?- consult('c:/mutants.pl').
% c:/mutants.pl compiled 0.00 sec, 1,568 bytes
Yes
?- mutant(X).
X = alligatortue ;
X = lapintade ;
X = chevalligator ;
X = chevalapin ;

No
?-

```