

TP N°3 : Prolog (Expressions arithmétiques)

Exercice 1 :

- 1) Exécuter les commandes suivantes :
M = 1 + 1.
N is 1 + 1.
N is 1+1, M is N+3, P = N + M.
- 2) Que remarquez-vous ?

Exercice 2 :

- 1) Exécuter les commandes suivantes :
write('bonjour'), write('à toi').
write('bonjour'), nl, write('à toi').
X is 2, write(X).
a_pour_successeur(X,Y) :- Y is X+1.
- 2) Comment exprimer qu'on veut calculer la somme de deux nombres « somme(X,Y,S) » ?
- 3) Comment exprimer qu'on veut connaître le plus grand parmi 2 nombres ? parmi 3 ?

Exercice 3 :

Afficher N fois 'bonjour' (Où : **ecrit(N)** est vrai si le message 'bonjour' est écrit N fois).

Exercice 4 :

- 1) Ecrire le programme Prolog suivant :
val-aff(0).
val_aff(N) :- N>0, write(N), nl, N1 is N-1, val_aff(N1).
- 2) Que fait ce programme ?
- 3) Ecrire un programme Prolog qui permet d'afficher les nombres de 1 à N.

Exercice 5 :

- 1) Ecrire le programme Prolog suivant :
nbre_p(0).
nbre_p(X) :- X>0, X2 is X-2, nbre_p(X2).
- 2) Que fait ce programme ?
- 3) Ecrire un programme Prolog qui trouve la somme des N premiers entiers (où: som(N,X) est vrai si X est la somme des entiers de 1 à N).
- 4) Ecrire un programme Prolog qui trouve la factorielle d'un nombre (où: fact(N,X) est vrai si X vaut N !).

Solution TP N°3 : Prolog (Expressions arithmétiques)

Exercice 1 : Exécuter les commandes suivantes :

```
M = 1 + 1.  
M = 1+1.  
N is 1 + 1.  
N = 2.  
N is 1+1, M is N+3, P = N + M.  
N = 2,  
M = 5,  
P = 2+5.
```

1) Que remarquez-vous ?

Le symbole « = » en prolog signifie l'unification et pas l'affectation. Pour affecter une valeur numérique à une variable, en évaluant mathématiquement le résultat, il faut utiliser « is ».

Exercice 2 : Il existe quelques actions prédéfinies comme write(), qui est applicable à une constante, une variable ou une chaîne de caractères entre guillemets simples, nl (passage à la ligne). En prolog, on ne définit pas de fonction ou de procédures mais des prédicats.

Donc, pour faire l'équivalent d'une procédure, on dit qu'un certain prédicat est vrai à condition qu'un certain nombre d'actions soient faites, par exemple : pour définir une fonction $f(X_1, \dots, X_n) = Y$, on définit un prédicat $P(X_1, X_2, \dots, X_n, Y)$ qui est vrai si $f(X_1, \dots, X_n) = Y$, par exemple: affiche :- write('bonjour').

1) Exécuter les commandes suivantes :

```
?- write('bonjour'), write('à toi').  
bonjourà toi  
true.  
?- write('bonjour'), nl, write('à toi').  
bonjour  
à toi  
true.  
?- X is 2, write(X).  
2  
X = 2  
a_pour_successeur(X,Y) :- Y is X+1. (c'est une règle)  
?- a_pour_successeur(2,Y)  
Y = 3.
```

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Relizane

Département Informatique
Intelligence Artificielle : L3 - SI

Année académique 2021/2022
Responsable du module : **S. Bella**

- 2) Comment exprimer qu'on veut calculer la somme de deux nombres « somme(X,Y,S) » ?
`somme(X,Y,S) :- S is X+Y.`
- 3) Comment exprimer qu'on veut connaître le plus grand parmi 2 nombres ? parmi 3 ?
`max2(X,Y,X) :- X>=Y.`
`max2(X,Y,Y) :- X<Y.`
`max3(X,Y,Z,M) :- max2(X,Y,MM), max2(MM,Z,M).`

Exercice 3 : En prolog, il n'y a pas d'autres moyens de faire des boucles que d'utiliser la récursivité. Afficher N fois 'bonjour' (Où : `ecrit(N)` est vrai si le message 'bonjour' est écrit N fois).

`ecrit(0).`
`ecrit(N) :- N>0, write('bonjour'), nl, N1 is N-1, escrit(N1).`

Exercice 4 :

- 1) Ecrire le programme Prolog suivant :
`val-aff(0).`
`val_aff(N) :- N>0, write(N), nl, N1 is N-1, val_aff(N1).`
- 2) Que fait ce programme ? **Il permet d'afficher des nombres d'une manière décroissante.**
- 3) Ecrire un programme Prolog qui permet d'afficher les nombres de 1 à N.
`croissant(0).`
`croissant(N) :- N>0, N1 is N-1, croissant(N1), write(N), nl.`
Ou
`croissant(0).`
`croissant(N) :- N>0, N1 is N-1, croissant(N1), write(N), tab(3).`

Exercice 5 :

- 1) Ecrire le programme Prolog suivant :
`nbre_p(0).`
`nbre_p(X) :- X>0, X2 is X-2, nbre_p(X2).`
- 2) Que fait ce programme ? **Dire si un nombre est pair.**
- 3) Ecrire un programme Prolog qui trouve la somme des N premiers entiers (où: `som(N,X)` est vrai si X est la somme des entiers de 1 à N).
`som(0,0).`
`som(N,X) :- N>0, N1 is N-1, som(N1,X1), X is N+X1.`
- 4) Ecrire un programme Prolog qui trouve la factorielle d'un nombre (où: `fact(N,X)` est vrai si X vaut N !)
`fact(0,1).`
`fact(N,X) :- N>0, N1 is N-1, fact(N1,X1), X is N*X1.`