Série de TP N°3

1. La récursivité

- Prolog ne propose pas de structures de contrôle telles que les boucles for, while, etc. Comme dans la plupart des langages fonctionnels et logiques, la manière "officielle" de gérer le contrôle est la *récursivité*.
- Récursivité : un programme est récursif lorsqu'il s'appelle lui même
- En Prolog, les prédicats peuvent "s'appeler" eux-mêmes, c'est- à- dire que le prédicat de tête d'une règle se retrouve dans la queue de la règle
- Il faut au moins deux clauses :
  - Une clause de base qui peut valider la récursivité (terminer la boucle).
  - Et une clause qui contient la récursivité‣
- La condition d'arrêt de la récursivité doit être en première position

Exemple 1:

enfant(omar, ali). %omar est l'enfant de ali.

enfant(mohamed,omar).

enfant(adam,mohamed).

On va définir le prédicat « descendant » : enfants, petits-enfants, arrières petits-enfants, arrières arrières petits-enfants, etc.

```
descendant(X,Y) :- enfant(X,Y).
descendant(X,Y) :- enfant(X,Z), descendant(Z,Y).
```

Exemple 2: La factorielle

factorielle(0) = 1 (Cas d'arrêt)

factorielle(n) = n * factorielle(n-1) si n>0 (Appel récursif)

la factorielle en prolog s'écrit:

```
factorielle(0,1).
factorielle(N,F) :- N>0, X is N-1, factorielle(X,Y), F is Y*N.
```

Exercices :

1. Écrire un prédicat qui permet d'écrire les nombres de 0 à N.
2. Écrire un prédicat qui fait la somme des **n** premiers nombres entiers.
3. Écrire un prédicat qui calcule la puissance N ième d'un nombre
4. Écrire un prédicat qui détermine si un nombre est pair
5. écrire un prédicat calculant le $n^{ième}$ terme de la suite de Fibonacci.

   $U_0 = U_1 = 1$

   $U_n = U_{n-1} + U_{n-2}$ pour n>1
6. Ecrire le prédicat **pgcd**/3 qui détermine le PGCD en tenant compte du contexte suivant :

   Propriétés du PGCD **D** de **X** et **Y**

   si **X** et **Y** sont égaux, **D** vaut **X**

   si **X<Y** alors **D** est le PGCD de **X** et de **Y-X**

   si **Y<X** alors échanger le rôle de **X** et **Y**

Corrigé :

1. Écrire les nombres de 0 à N :

   print_numbers(0) :– write (0).

   print_numbers (N) : – N1 is N–1, print_numbers(N1), nl, write (N) .
2. La somme des **n** premiers nombres entiers

   som(0,0).

   som(N,X) :- N>0, N1 is N-1, som(N1,X1), X is N+X1.
3. La puissance N ième d'un nombre

   puissance(N,1,N).

   puissance(Nb,P,Res):- P>1,P1=P-1,puissance(NB,P1,Res1),Res=Res1*Nb.

   Ou

   puissance(N,1,N):-!.

   puissance(Nb,P,Res):- P1=P-1,puissance(NB,P1,Res1),Res=Res1*Nb
4. Nombre paire

   pair(0).

   pair(X) :- X>0, X2 is X-2, pair(X2).
5. Le $n^{ième}$ terme de la suite de Fibonacci.

   fibo(1,1).

   fibo(2,1).

   fibo(N,X) :- N>2, U is N-1, V is N-2, fibo(U,U1), fibo(V,V1), X is U1+V1.

6. Le PGCD

```
pgcd(X,X,X).
pgcd(X,Y,D):-X<Y,X1 is Y-X,pgcd(X,X1,D).
pgcd(X,Y,D):-X>Y,pgcd(Y,X,D).
```