Université Grenoble Alpes
U.F.R. Sciences de l'Homme et de la Société
Licence MIASHS 2ème année

TD Programmation en Logique n°3

Exercice 1

On considère le programme Prolog ci-dessous.
a) Quelles solutions pour X et Y retourne Prolog à la requête r(X,Y) ?

```
r(0,0).
r(X,Y):- p(X), q(Y).
r(Y,X):- p(Y), q(Y), X=3, Y<X.
r(5,5).
p(1).
p(2).
q(2).
q(3).
```

b) Même question avec l'ajout d'un cut en deuxième ligne : `r(X,Y):-p(X), q(Y), !.`

Exercice 2

Soit le programme suivant.
Quelles sont toutes les solutions des requêtes suivantes ?

a) a(0,X).
b) a(1,X).
c) a(2,X).
d) a(3,X).

```
a(0,Y):-!,c(Y).
a(X,Y):-b(X,Y),!,c(X).
a(_,_).
b(1,2).
b(1,3).
b(2,5).
c(1).
c(2).
c(3).
```

Exercice 3

Ecrire le prédicat multListe/3 qui, étant donné une liste d'entiers et un coefficient, détermine une nouvelle liste dans laquelle tous les nombres de la première liste ont été multipliés par le coefficient :

?- multListe([5,3,7,1,0],2,L).
L=[10,6,14,2,0]

Exercice 4

Soit deux listes composées chacune des chiffres de 0 à 9. Ecrire diff/3 qui construit la liste des différences entre les éléments de la première liste et ceux de la seconde liste.

```
?- diff([4,6,1,2,9,0,3,7,5,8],[3,4,9,0,1,5,2,7,8,6],R).
R=[1,2,-8,2,8,-5,1,0,-3,2]
```

Exercice 5

Ecrire le prédicat lettres/2 qui détermine la liste des lettres des mots d'une liste donnée. On utilisera le prédicat atom_chars/2 qui permet de transformer un terme en une liste de lettres.

```
?- atom_chars(bonjour,L).
L = [b, o, n, j, o, u, r].

?- lettres([baba,ane,bien],R).
R=[b,a,e,i,n]
```

Exercice 6

On dispose de faits représentant le prix au kilo de fruits.

```
prix(pomme,7).
prix(melon,13).
prix(banane,8).
...
```

Ecrire le prédicat Prolog *total* qui détermine le montant total d'une liste de fruits avec leur poids. Chaque couple fruit/poids est représenté par une liste à deux éléments. Le premier paramètre est donc une liste de listes.

```
?- total([[banane,1.5],[pomme,0.5]],T).
T=15.5
```

Exercice 7

Ecrire les prédicats intersection/3 et union/3 qui déterminent respectivement l'intersection et l'union de deux ensembles donnés représentés dans des listes. Il n'y a pas d'éléments en double dans ces listes.

Exemples :

```
?- union([6,2,3,9,4],[7,4,6],R).
R=[6,2,3,9,4,7]
```

```
?- intersection([6,2,3,9,4],[7,4,6],R).
R=[6,4]
```

Exercice 8

Ecrire le prédicat tousDifferents/1 qui est vrai si tous les éléments de la liste passée en arguments sont différents.

```
?- tousDifferents([3,6,5,3]).
false
```

Exercice 9

Ecrire le prédicat motPrec/3 qui, étant donné un élément et une liste, détermine la liste de tous les éléments qui précèdent celui-ci.

```
?-motPrec(petit,[le,petit,chat,dort,sur,ce,petit,lit],L).
L=[le,ce]
```

Exercice 10

Ecrire le prédicat permut/2 qui prend deux listes et qui retourne vrai si l'une est une permutation de l'autre.

```
?- permut([a,b,c,d,e],[d,a,e,c,b]).
true
?- permut([a,b,c,d,e],[d,a,c,c,e,e,b]).
False
```