

TD Programmation en Logique n°4

Exercice 1 : Intégramme célèbre

Un des intégrammes les plus célèbres est celui des cinq maisons, parfois appelé *énigme d'Einstein* en référence au fait qu'il aurait été inventé par le physicien Albert Einstein. Certains ajoutent qu'Einstein aurait dit que seulement 2 % de la population est capable de le résoudre. Il est aussi parfois attribué à Lewis Carroll. Cependant, il n'existe aucune preuve de la paternité de l'un ou de l'autre sur cette énigme.

Voici cette énigme :

Cinq hommes de nationalité et de professions différentes habitent avec leur animal préféré cinq maisons de couleurs différentes où ils boivent leur boisson préférée. Nous disposons de 14 indices :

1. *L'anglais habite dans la maison rouge.*
2. *Le chien appartient à l'Espagnol.*
3. *On boit du café dans la maison verte.*
4. *L'ukrainien boit du thé.*
5. *La maison verte est située à droite de la maison blanche.*
6. *Le sculpteur élève des escargots.*
7. *Le diplomate habite dans la maison jaune.*
8. *On boit du lait dans la maison du milieu.*
9. *Le norvégien habite la première maison à gauche.*
10. *Le médecin habite une maison voisine de celle où demeure le propriétaire du renard.*
11. *La maison du diplomate est à côté de celle où il y a un cheval.*
12. *Le violoniste boit du jus d'orange.*
13. *Le japonais est acrobate.*
14. *Le norvégien habite à côté de la maison bleue*

Questions : Qui boit de l'eau ? Qui possède le zèbre ?

Ecrire un programme Prolog qui trouve la solution de cette énigme, en appliquant les principes de modélisation vus en cours. On représentera la solution par une liste de 5 termes, chaque terme aura la forme suivante : maison(couleur,nationalité,animal,boisson,métier)
 On appellera **einstein/0** le prédicat qui traduit chaque indice de l'énigme en Prolog et affiche la liste solution si toutes les conditions sont remplies.

Exercice 2

Soit une matrice représentée par une liste de liste. Ainsi, la matrice ci-contre est représentée par la liste $[[1,0,4,2],[3,2,1,1],[0,1,0,6],[2,7,4,3]]$.

1	0	4	2
3	2	1	1
0	1	0	6
2	7	4	3

Ecrire le prédicat somme/2 qui fait la somme des éléments d'une matrice.

?- somme([[1,0,4,2],[3,2,1,1],[0,1,0,6],[2,7,4,3]],R).

R=37

Exercice 3

Ecrire une version du prédicat `palindrome/1`, qui reconnaisse les phrases palindromes :

?- `palindrome ('engage le jeu que je le gagne')`.

`true`

Exercice 4

Ecrire le prédicat `Prolog nb/3` qui détermine le nombre d'éléments d'une liste qui appartiennent à une seconde liste.

?- `nb([tulipe,rose,bleuet],[rose,oeillet,muguet,pivoine,tulipe,jonquille],R)`.

`R=2`

Exercice 4 : Arbres binaires

a) Ecrire un prédicat `maxi/2` qui recherche la plus grande valeur d'un arbre binaire d'entiers.

`maxi(A,M)` unifie `M` avec la valeur maximale de l'arbre `A`.

b) Ecrire le prédicat `profond_arbre/2` qui détermine la profondeur d'un arbre binaire donné.

c) Ecrire le prédicat `maxi2/2`, qui recherche la plus grande valeur dans un arbre binaire de recherche, c'est-à-dire un arbre dont les nœuds vérifient les propriétés suivantes :

- Tout nœud situé dans un sous-arbre gauche est plus petit ou égal à la racine
- Tout nœud situé dans un sous-arbre droit est plus grand que la racine

d) Ecrire le prédicat `creer_ord/2` qui crée un arbre binaire de recherche à partir d'une liste d'entiers.

Pour réaliser ce prédicat on utilisera un prédicat intermédiaire avec un accumulateur : l'arbre en cours de construction qui, à l'appel initial, sera appelé avec un arbre vide.