

L3 informatique – partiel de Prolog**Durée : 80 minutes – sans document ni moyen électronique***Répondre uniquement dans les cadres prévus à cet effet*

Nom :	N° dossier :
Prénom(s) :	Signature :
Date de naissance :	Section :

Exercice 1 (4 ●) Résolvez les systèmes d'équations suivants dans l'algèbre des termes finis :

a) $\{p(Y, X, h(Y)) = p(h(g(a), Z), f(V), V)\}$

b) $\{p(a, X) = p(X, Y, Z)\}$

c) $\{p([X|Y], [Y|Z], V) = p([a, b, c], W, [X, W])\}$

d) $\{p(X, Z, X) = p(g(Z, Z), a, g(b, Y))\}$

Exercice 2 (6 ●) Considérez le programme suivant :

```
p(X,Y) :- q(X), r(X,Y).    % c1
q(X) :- t(X).             % c2
q(X) :- s(X).             % c3
s(a).                      % c4
s(b).                      % c5
r(a,b).                    % c6
r(b,c).                    % c7
r(c,b).                    % c8
```

et dessinez *soigneusement* l'arbre de dérivations de la requête :- p(A,B).



Nom :	N° dossier :
Prénom(s) :	Signature :
Date de naissance :	Section :

Exercice 3 : (10 ●) Un *arbre binaire* est soit l'arbre vide, soit un arbre ayant exactement deux sous-arbres qui sont eux aussi des arbres binaires. La valeur attachée à un nœud de l'arbre s'appelle *étiquette* du nœud. On représente les arbres binaires par :

- la constante `vide`;
- un terme de la forme `noeud(Etiquette,Gauche,Droit)` où `Etiquette` est l'étiquette de la racine, `Gauche` est le sous-arbre binaire gauche et `Droit` le sous-arbre binaire droit.

Donnez le code Prolog (sans coupure ni négation) des relations suivantes :

3.1 (1 ●) `racine(A,E)` ssi `E` est l'étiquette de la racine de l'arbre `A`.

```
?- racine(noeud(a,vide,vide),R).
```

```
R = a.
```

```
?-
```

3.2 (1 ●) `vide(A)` ssi `A` est l'arbre vide.

```
?- vide(noeud(a,vide,vide)).
```

```
false.
```

```
?-
```

3.3 (1 ●) `feuille(A)` ssi `A` est un arbre binaire réduit à une feuille.

```
?- feuille(noeud(a,vide,vide)).
```

```
true.
```

```
?-
```

3.4 (1 ●) `dans(E,A)` ssi `E` est une étiquette de l'arbre binaire `A`.

```
?- dans(E,noeud(a,vide,vide)).
```

```
E = a ;
```

```
false.
```

```
?-
```

3.5 (1 ●) `taille(A,N)` ssi `N` est le nombre d'étiquettes de l'arbre binaire `A`.

?- `taille(noeud(a,vide,noeud(a,vide,vide)),N)`.

`N = 2.`

?-

3.6 (1 ●) `max(A,B,Max)` ssi `Max` est le maximum des nombres `A` et `B`.

?- `max(2,4,M)`.

`M = 4.`

?-

3.7 (1 ●) `hauteur(A,H)` ssi `H` est la hauteur de l'arbre binaire `A`.

La *hauteur* d'un arbre correspond au nombre de nœuds de la plus longue branche entre la racine et une feuille. L'arbre vide aura la hauteur 0 et une feuille la hauteur 1.

?- `hauteur(noeud(a,vide,noeud(b,noeud(c,vide,vide),vide)),N)`.

`N = 3 ;`

`false.`

?-

3.8 (3 ●) `symetrique(A)` ssi la silhouette de l'arbre binaire `A` est symétrique.

La silhouette d'un arbre binaire est *symétrique* si, en traçant une ligne verticale à partir de la racine de la représentation graphique usuelle de `A`, le sous-arbre gauche est le miroir du sous-arbre droit.

?- `symetrique(noeud(a,noeud(b,vide,noeud(c,vide,vide)),noeud(d,noeud(e,vide,vide),vide)))`.

`true.`

?-