

# Exercices

## Table des matières

ABR (Niveau A).....	2
Strings (Niveau C).....	3
Généalogie (Niveau C).....	4
Il fait beau et chaud (Niveau B).....	5
Logigramme 1 (Niveau A).....	6
Logigramme 2 (Niveau A).....	7
Logigramme 3 (Niveau B).....	8
Voyageur (Niveau A).....	9
Jeux de la vie (Niveau A).....	10
Évaluation de l'examen de BD (Niveau B).....	11
Classification des espèces vivantes (Niveau C).....	12

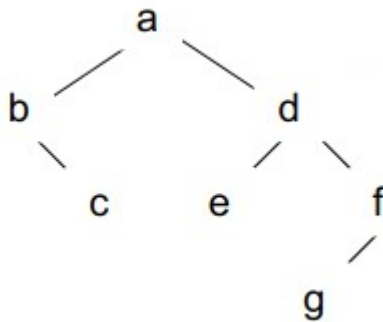
## ABR (Niveau A)

On représente un arbre binaire par une liste PROLOG de trois éléments qui sont :

- la représentation du sous-arbre gauche,
- l'étiquette,
- la représentation du sous-arbre droit.

Un arbre vide est représenté par []

Par exemple



est représenté par la liste `[ [], b , [ [],c,[] ] ,a, [ [ [],e,[] ] , d , [ [ [],g,[] ] , f ,[] ] ] ]`

Rappel : l'étiquette d'un nœud est supérieure à toutes les étiquettes de son sous-arbre gauche et inférieure ou égale à tous les étiquettes du sous-arbre droit. Les sous-arbres sont également des arbres binaires de recherche.

Écrire le prédicat « `ab_valide(L)` » qui est vrai si `L` représente un arbre binaire selon la structure donnée ci-dessus.

Écrire le prédicat « `abr_valide(L)` » qui est vrai si `L` représente un arbre binaire selon la structure donnée ci-dessus.

Écrire les prédicats « `abr_display_prefixe(L)` », « `abr_display_infixe(L)` » et « `abr_display_postfixe(L)` » qui affichent respectivement l'ABR « `L` » de manière préfixe, infixe et postfixe.

Écrire le prédicat « `appartient(X, L)` » qui est vrai si `X` appartient à l'ABR `L`.

## Strings (Niveau C)

Nous considérons une chaîne de caractères comme une liste de caractères en minuscule. Exemple : [t, o, t, o]

Écrire le prédicat « premier\_car(C, M) » qui est vrai si « C » est le premier caractère de « M ».

Exemple :

- premier\_car(t, [t,o,t,o]) → vrai
- premier\_car(o, [t,o,t,o]) → faux

Écrire le prédicat « dernier\_car(C, M) » qui est vrai si « C » est le dernier caractère de « M ».

Exemple :

- dernier\_car(t, [t,o,t,o]) → faux
- dernier\_car(o, [t,o,t,o]) → vrai

Écrire le prédicat « prefixe(P, M) » qui est vrai si « P » est le préfixe de « M ».

Exemple :

- prefixe([t,o], [t,o,t,o]) → vrai
- prefixe([t,o,t], [t,o,t,o]) → vrai
- prefixe([o,t], [t,o,t,o]) → faux
- 

Écrire le prédicat « suffixe(P, M) » qui est vrai si « P » est le suffixe de « M ».

Exemple :

- suffixe([t,o], [t,o,t,o]) → vrai
- suffixe([t,o,t], [t,o,t,o]) → faux
- suffixe([o,t], [t,o,t,o]) → faux

Écrire le prédicat « palindrome(M) » qui est vrai si « M » est un palindrome (un mot qui se lit dans les 2 sens).

Exemple :

- palindrome([t,o,o,t]) → vrai
- palindrome([t,o,t,o]) → faux
- palindrome([i,c,i]) → vrai

# Généalogie (Niveau C)

Considérons la base de faits :

% bio(nom, genre, naissance, mort, nom\_mere, nom\_pere)

bio(louis13, h, 1601, 1643, marie\_medicis, henri4).  
bio(elisabeth\_france, f, 1603, 1644, marie\_medicis, henri4).  
bio(marie\_therese\_autriche, f, 1638, 1683, elisabeth\_france, philippe4).  
bio(louis14, h, 1638, 1715, anne\_autriche, louis13).  
bio(grand\_dauphin, h, 1661, 1711, marie\_therese\_autriche, louis14).  
bio(louis\_bourbon, h, 1682, 1712, marie\_anne\_baviere, grand\_dauphin).  
bio(philippe5, h, 1683, 1746, marie\_anne\_baviere, grand\_dauphin).  
bio(louis15, h, 1710, 1774, marie\_adelaide\_savoie, louis\_bourbon).  
bio(louis\_dauphin, h, 1729, 1765, marie\_leczcynska, louis15).  
bio(louis16, h, 1754, 1793, marie\_josephe\_saxe, louis\_dauphin).  
bio(louis18, h, 1755, 1824, marie\_josephe\_saxe, louis\_dauphin).  
bio(charles10, h, 1757, 1836, marie\_josephe\_saxe, louis\_dauphin).  
bio(clotilde, f, 1759, 1802, marie\_josephe\_saxe, louis\_dauphin).  
bio(louis17, h, 1785, 1795, marie\_antoINETTE, louis16).  
bio(philippe1, h, 1640, 1701, anne\_autriche, louis13).  
bio(philippe2, h, 1674, 1723, charlotte\_baviere, philippe1).  
bio(louis\_orleans, h, 1703, 1752, francoise\_marie\_bourbon, philippe).  
bio(louis\_philippe, h, 1725, 1785, augusta\_marie\_bade, louis\_orleans).  
bio(philippe\_egalite, h, 1747, 1793, louise\_henriette\_bourbon\_conti, louis\_philippe).  
bio(louis\_philippe1, h, 1773, 1850, louise\_marie\_adelaide\_bourbon, philippe\_egalite).

1. Demander à prolog quelles sont les noms des femmes mentionnées dans la base de faits.
2. Écrire le prédicat « mere(X) » qui est vrai si X est une mère dans la base de faits.
3. Écrire le prédicat « enfant(X,Y) » qui est vrai si X est l'enfant de Y.
4. Écrire le prédicat « contemporain(X,Y) » qui est vrai si X est née entre la naissance et la mort de Y, ou Y est née entre la naissance et la mort de X.
5. Écrire le prédicat « descendant(X,Y) » qui est vrai si X est un descendant de Y.

## Il fait beau et chaud (Niveau B)

Si la pression atmosphérique est supérieure à 1015 hPa nous sommes dans un anticyclone, si elle est inférieure à 1015 hPa nous sommes dans une dépression.

Si la température est supérieure à 23° il fait chaud, sinon il fait froid.

Si l'hygrométrie est supérieure à 80 %, le temps est instable, sinon il est stable.

Modéliser ceci en Prolog

Selon les régions du monde ces valeurs ne sont pas aussi strictes, reprendre la modélisation en considérant que nous sommes en dépression en dessous de 1019 hPa et en anticyclone au dessus de 1011 hPa, qu'il fait froid en dessous de 24° et qu'il fait chaud au dessus de 21°, et pour finir que le temps est stable en dessous de 79 % d'humidité et instable au dessus de 71 % d'humidité.

Si nous sommes dans un anticyclone et qu'il fait chaud, je suis heureux. Si nous sommes en dépression et qu'il fait froid, je suis morose. Sinon, je suis neutre.

Modéliser ceci en Prolog

Si je suis heureux et que le temps est instable, je dense ; Si je suis heureux et que le temps est stable, je me baigne ; si je suis morose et que le temps est instable je cocoone, si je suis morose et que le temps est stable je jardine ; si je suis neutre et que le temps est stable je me promène, si je suis neutre et que le temps est instable je m'ennuie.

Demander à Prolog ce que je fait, la pression est à 1015 hPa, il fait 21° et l'humidité est à 72 %.

Proposer la transformation de mon activité en une valeur numérique représentant mon taux d'activité, sachant que je suis entre 0 et 5 % quand je cocoone, entre 5 et 10 % quand je m'ennuie, entre 10 et 35 % quand je me promène, entre 35 et 60 % quand je jardine, entre 60 et 80 % quand je me baigne et entre 80 et 100 % quand je dense.

# Logigramme 1 (Niveau A)

Question de pot

Un magasin de jardinage vend des pots de fleurs. Retrouver les caractéristiques de chacun.

Indices

1. Le pot en terre n'est pas hexagonal et son prix est supérieur à celui du pot en plastique.

2. Le pot de 40cm n'est pas en plastique et coûte 10€ de moins que le pot rectangulaire.

3. Le pot en résine ne coûte pas 25€.

4. Le pot carré ne coûte pas 10€. Il a 10cm de plus que celui en terre mais 10 de moins que celui qui coûte 25€.

		Forme				Taille				Prix			
		Carré	Hexagone	Rectangle	Cercle	20 cm	30 cm	40 cm	50 cm	10 €	15 €	20 €	25 €
Matière	Plastique												
	Résine												
	Terre												
	Zinc												
Prix	10 €												
	15 €												
	20 €												
	25 €												
Taille	20 cm												
	30 cm												
	40 cm												
	50 cm												

Matière	Forme	Taille	Prix
Plastique			
Résine			
Terre			
Zinc			

## Logigramme 2 (Niveau A)

### L'été indien

Après un été mi-figue mi-raisin, l'été indien s'installe sur le pays. Découvrez les températures estivales qui s'affichent sur le thermomètre ainsi que le début et la durée de la période de chaleur dans différentes villes.

1. La température à Clermont-Ferrand est supérieure de 2°C à celle de Mulhouse et inférieure de 2°C à celle qui dure 5 jours.
2. À Brest, la vague de chaleur débute le 21 septembre. Il y fait 1°C de plus que dans la ville qui a chaud 8 jours et qui dépasse les 27°C.
3. La période de chaleur de Lille débute 2 jours avant celle de la ville où il fait 25°C et 2 jours après celle qui dure 4 jours.
4. La chaleur à Mulhouse dure un jour de moins qu'à Lille et un de plus que dans la ville où le phénomène débute le 21 septembre.
5. La chaleur qui débute le 25 septembre dure plus de 6 jours.

vos notes

		Début					Record					Durée				
		21 sept.	23 sept.	25 sept.	27 sept.	29 sept.	25°C	26°C	27°C	28°C	29°C	4 jours	5 jours	6 jours	7 jours	8 jours
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Ville	Brest	1														
	Clermont	2														
	Lille	3														
	Mulhouse	4														
	Reims	5														
Durée	4 jours	6														
	5 jours	7														
	6 jours	8														
	7 jours	9														
	8 jours	10														
Record	25°C	11														
	26°C	12														
	27°C	13														
	28°C	14														
	29°C	15														

Ville	Début	Record	Durée
Brest			
Clermont			
Lille			
Mulhouse			
Reims			

# Logigramme 3 (Niveau B)

**Le pitch**  
Retrouvez qui fait quoi et avec quel outil !

**Les indices**

- 1 - Princesse aime Manger
- 2 - On ne joue pas avec le coussin
- 3 - L'outil d'Aramis n'est ni la salade ni le coussin
- 4 - On utilise pas la salade pour dormir
- 5 - L'outil de Gribouille est le coussin

		Qui			Quoi		
		Aramis	Princesse	Gribouille	Jouer	Dormir	Manger
Outil	Salade						
	Coussin						
	Balle						
Quoi	Jouer						
	Dormir						
	Manger						



## Voyageur (Niveau A)

Étant donnés deux ensembles A et B représentés par des listes, construire les prédicats prolog `union(A,B,R1)` et `inter(A,B,R2)` qui réussit si  $R1 = A \cup B$  et  $R2 = A \cap B$  respectivement.

On possède les données suivantes :

```
#equipe(PERSONNE, EQUIPEMENTS)
equipe_avec(jean,[bottes,bonnet,pull]).
equipe_avec(paul,[parapluie, sandales, kway, short, gants]).
equipe_avec(jacques,[bob, moufles]).
```

```
#temps(REGION, TEMPS)
temps(nord,[froid,pluie]).
temps(sud,[soleil]).
temps(est,[neige,froid]).
temps(ouest,[pluie, chaud]).
```

```
#requis(TEMP, EQUIPEMENTS)
equipe_requis(froid, [pull,bonnet,gants,echarpe]).
equipe_requis(pluie, [parapluie,kway]).
equipe_requis(soleil, [sandales, bob, short, lunettes]).
equipe_requis(neige, [pull,moufles,gants]).
equipe_requis(chaud, [sandales, short]).
```

Écrire un prédicat `equipe_pour(Region,Personne)` qui réussit si la personne « Personne » est équipée pour partir dans la région « Region », c'est-à-dire, a un équipement complet pour la région.

# Jeux de la vie (Niveau A)

## Présentation du Jeu de la Vie

Le jeu de la vie est une création de John Conway. C'est le plus connu des automates cellulaires : une grille de *cellules* évoluant au cours du temps selon un petit nombre de règles. Ici, on se place dans une grille simple, faite de carrés, chaque cellule est donc encadrée par 8 autres cellules. Malgré des règles très simples, le jeu de la vie permet le développement de motifs extrêmement complexes.

## Règles du Jeu de la Vie

Les règles de ce jeu sont les suivantes :

- Chaque cellule ne peut avoir que deux état : la vie (1) et la mort (0).
- Si une cellule en vie n'a qu'une voisine ou aucune, elle meurt de solitude.
- Si une cellule en vie a 4 voisines ou plus, elle meurt de surpopulation.
- Avec 2 ou 3 voisins, une cellule en vie reste en vie.
- Quand exactement 3 cellules vivantes encadrent une quatrième morte, celle-ci renaît à l'étape suivante.

```
plateau([  
[ 0, 0, 0, 0, 0],  
[ 0, 0, 1, 0, 0],  
[ 0, 0, 1, 0, 0],  
[ 0, 0, 1, 0, 0],  
[ 0, 0, 0, 0, 0]])
```

Conseil : écrivez les prédicats `case(X, Y, V)`, qui est vrai si V est la valeur contenue à la case (X,Y), et le prédicat `valeurs_voisines(X, Y, L)` qui est vrai si L est la liste des valeurs des cases voisines de (X,Y).

Note il y a 8 cases voisines : 2 verticales, 2 horizontales et 4 diagonales.

Écrire le prédicat `joue(P, N)`, et tous les prédicats intermédiaire nécessaire, qui affiche N tour de jeux :

exemple :

?- : `plateau(P), joue(P, 1).`

affiche :

```
[[ 0, 0, 0, 0, 0],  
[ 0, 0, 1, 0, 0],  
[ 0, 0, 1, 0, 0],  
[ 0, 0, 1, 0, 0],  
[ 0, 0, 0, 0, 0]]
```

```
[[ 0, 0, 0, 0, 0],  
[ 0, 0, 0, 0, 0],  
[ 0, 1, 1, 1, 0],  
[ 0, 0, 0, 0, 0],  
[ 0, 0, 0, 0, 0]]
```



## Évaluation de l'examen de BD (Niveau B)

L'examen de BD comporte 4 exercices évalués chacun A, B, C, D ou E. Ce qui signifie :

A (20, 16) : Excellent travail comportant quelques erreurs négligeable ou d'étourderie

B (17, 13) : Très bon travail comprenant des erreurs non négligeables

C (14, 10) : Travail contenant des erreurs majeurs, mais ne remettant pas en cause la compréhension

D (9, 0) : Travail ne mettant pas en évidence la compréhension de l'exercice

E (0) : Non évaluable

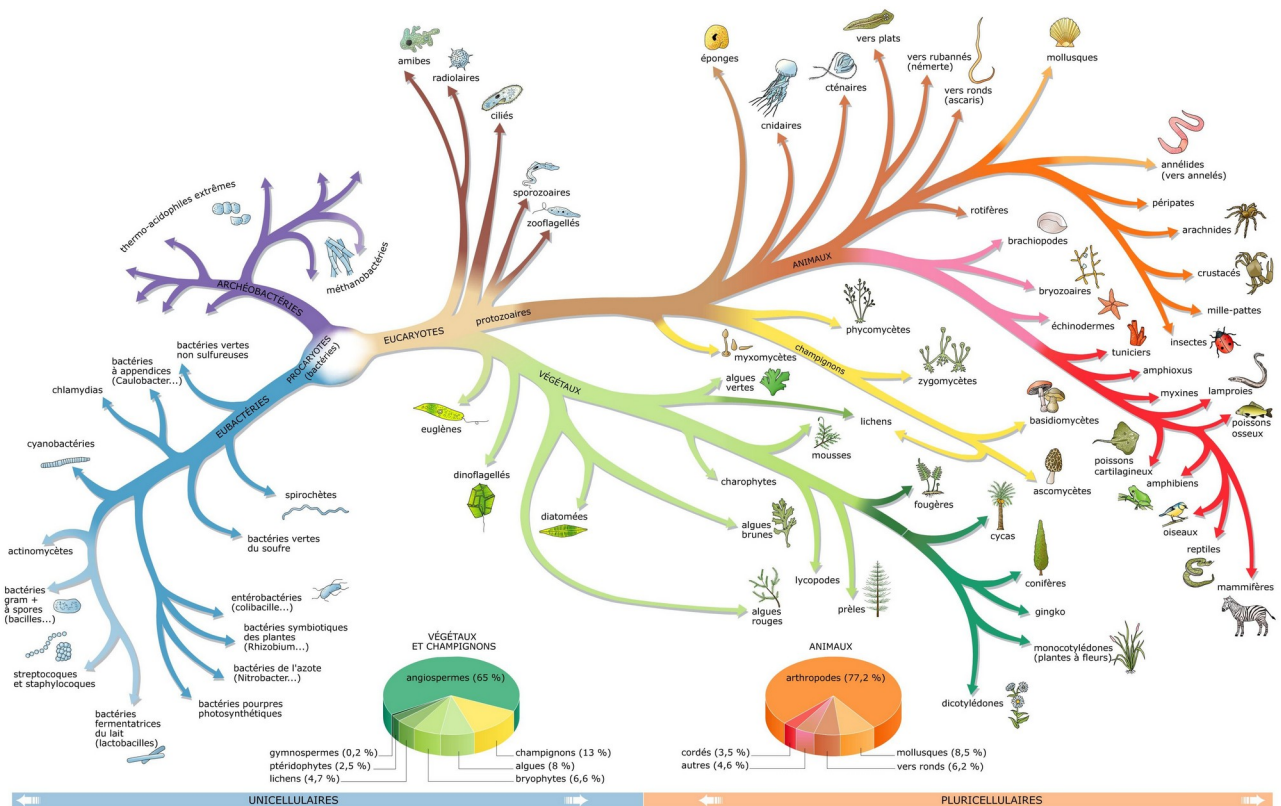
Étant donné les évaluations E1 et E2 de 2 exercices, le tableau ci-après donne les règles d'inférence de la moyenne des 2 exercices.

E1\E2	A	B	C	D	E
A	A	B	C	D	E
B	B	B	C	D	E
C	C	C	C	D	E
D	D	D	D	D	E
E	E	E	E	E	E

Écrire le programme prologue qui infère la note numérique à partir des évaluations E1, E2, E3 et E4 des 4 exercices.

# Classification des espèces vivantes (Niveau C)

Proposer un programme prolog permettant de représenter la base de connaissance de la classification des espèces vivantes.



source : [https://www.larousse.fr/encyclopedie/cartes/Classification\\_des\\_esp%C3%A8ces\\_vivantes/1309193](https://www.larousse.fr/encyclopedie/cartes/Classification_des_esp%C3%A8ces_vivantes/1309193)

Vous ne prendrez naturellement qu'un sous ensemble de la classification dans votre base.

Proposer un prédicat listeEspeces(ETRE\_VIVANT, LISTE\_ESPECES) qui est vrai si l'être vivant est un descendant de la liste d'espèces.

Exemple : listeEspeces(algues\_rouges, [procaryotes, eucaryotes, vegetaux]) est vrai.

Proposer un prédicat etreVivant(ESPECE, LISTE\_ETREs\_VIVANT) qui est vrai si la liste d'êtres vivant contient tous les descendants de l'espèce.

Exemple : etreVivant(protozoaires, [eponges, cnidaires, ctenaires, vers\_plats, ..., lichens, myxomycetes]) est vrai.

