

Correction sujet 2 - Centres Etrangers - 2023

Exercise 1 (4 points)

Partie A

1. a. 192.168.5.1
- b. 254 machines. (256 - 1 adresse réseau - 1 adresse routeur = 254.)
2. a. 255.255.240.0
- b. Le masque réseau est en binaire égal à 11111111.11111111.11110000.00000000.
En prenant l'adresse IP d'une machine comme 192.168.2.3, on effectue un ET binaire entre l'adresse et le masque :
- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
- On obtient donc l'adresse réseau **192.168.0.0**.
- c. Si l'un des routeurs vient à tomber en panne, la communication entre les autres réseaux n'est pas interrompue.

Partie B

1. a. Entre A et E : A - B - E (le plus court)
Entre F et B : F - D - A - B
 F - D - G - B
 F - H - G - B
 F - H - E - B

b.

Table de routage du routeur E		
Destination	Routeur suivant	Distance
A	B	2
B	B	1
C	H	2
D	G	2
F	H	2
G	G	1
H	H	1

Table de routage du routeur G		
Destination	Routeur suivant	Distance
A	B / D	2
B	B	1
C	D / H	2
D	D	1
E	E	1
F	D / H	2
H	H	1

2. a.

Table de routage du routeur F		
Destination	Routeur suivant	Coût total
A	D	1,1
B	H	10,11
C	D	1,1
D	D	0,1
E	H	10,1
G	D	1,1
H	H	0,1

b. E - H - F - D. (Coût : 10,2.)

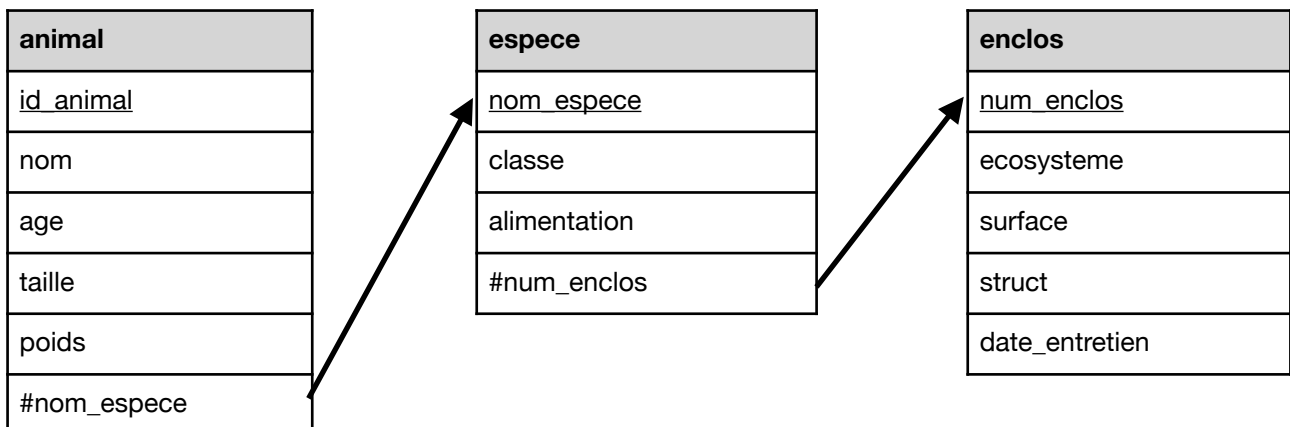
Exercice 2 (4 points)

1. a. 6, 1.70, 100 (En supposant qu'aucun autre animal ne s'appelle Moris...)

b. SELECT nom, age FROM animal
WHERE nom_espece = 'bonobo'
ORDER BY age ;

2. a. La clé primaire de la relation espece est nom_espece, car c'est le seul descripteur de la relation espece qui soit unique.
La clé étrangère de la relation espece est num_enclos, car ce descripteur fait référence à la clé primaire de la relation enclos.

b.



3. a. UPDATE espece
SET classe = 'mammifères'
WHERE nom_espece = 'ornithorynque' ;

b. INSERT INTO animal VALUES (179, 'Serge', 0, 0.8, 30, 'lama') ;
4. a. SELECT nom, nom_espece
FROM animal
JOIN espece ON animal.nom_espece = espece.nom_espece
JOIN enclos ON espece.num_enclos = enclos.num_enclos
WHERE enclos.struc = 'vivarium' AND espece.alimentation = 'carnivore' ;
(Pour la dernière ligne, les caractères *espece*. ne sont pas obligatoires.)

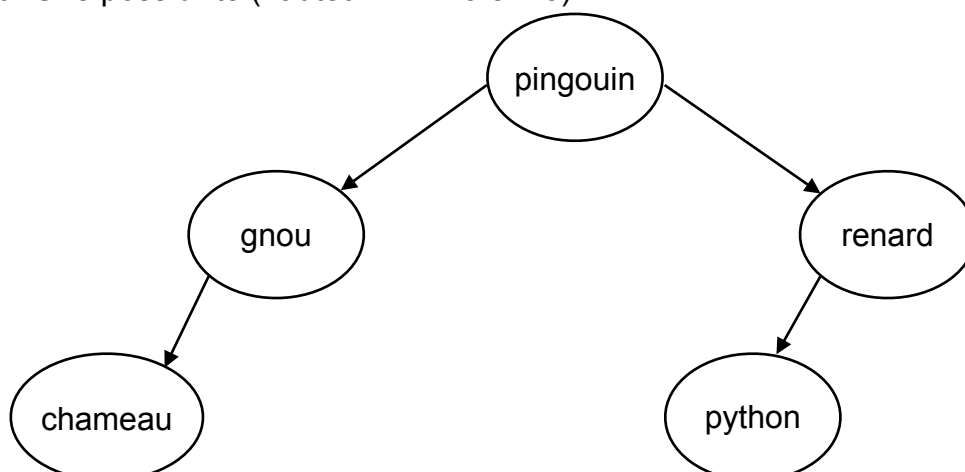
b. SELECT COUNT(*)
FROM animal
JOIN espece ON animal.nom_espece = espece.nom_espece
WHERE espece.classe = 'oiseaux' ;
(Pour la dernière ligne, les caractères *espece*. ne sont pas obligatoires.)

Exercice 3 (4 points)

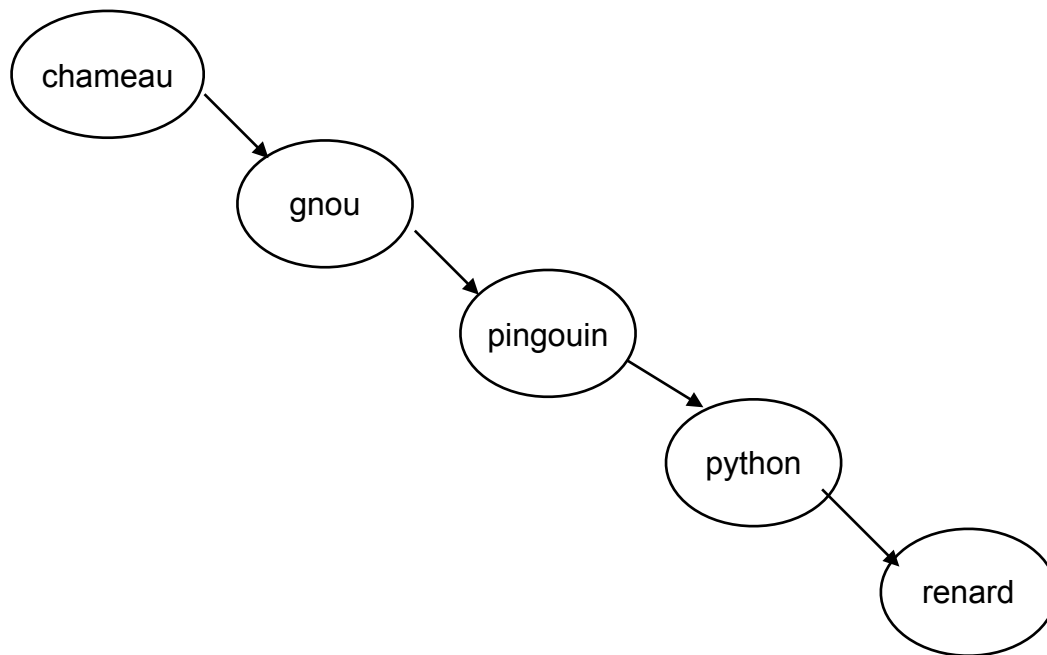
1. a. La méthode retourne la chaîne de caractères 'Bonjour Alan !'.

b. x et y sont des booléens. x vaut *False* et y vaut *True*.

c. def occurrences_lettres (une_chaine, une_lettre) :
 compteur = 0
 for c in une_chaine :
 if c == une_lettre :
 compteur = compteur + 1 #Autre : compteur += 1
 return compteur
2. a. Une possibilité (hauteur minimale = 3) :



b. Une possibilité (hauteur maximale = 5) :



3. a. La méthode retourne 0 si l'Abr ne contient pas de mot, et il ajoute 1 aux appels des sous-arbres gauche et droit s'il en contient un. On peut donc affirmer que la fonction mystere compte le nombre de mots dans l'Abr.

Dans notre cas, l'appel `mystere(abr_mots_français)` renvoie **336 531**.

```
b. def hauteur(un_arbre) :  
    #Condition d'arrêt  
    if un_arbre.est_vide() :  
        return 0  
    #Appel récursif  
    else :  
        a = hauteur(un_arbre.sous_arbre_gauche)  
        b = hauteur(un_arbre.sous_arbre_droit)  
        if a < b :  
            return b + 1  
        else :  
            return a + 1
```

4. a. 4 if `len(liste_mots[i]) == longueur \`
 and `liste_mots[i][position] == lettre :`

b. L'appel retourne ['fax', 'max'].

Il fait un premier appel pour récupérer la liste des mots de longueur 3 avec un 'x' en position 2, puis de cette liste il fait un nouvel appel pour affiner avec les mots de longueur 3 avec un 'a' en position 1.

c.

```
chercher_mots(chercher_mots(chercher_mots(liste_mots_francais,5  
, 't', 3), 5, 'e', 4), 5, 'r', 5)
```