Éléments de correction sujet 04 (2023)

Exercice 1

Partie A

1.
   a. adresse IP routeur F : 192.168.5.254
   b. Il sera possible de connecter 256 - 2 = 254 machines au maximum (en comptant le routeur)
2.
   a. masque de sous-réseau : 255.255.240.0 soit en binaire 11111111.11111111.11110000.00000000.
   b. On effectue un ET logique entre le masque de sous réseau et l'adresse IP 192.168.2.2 (en binaire 11000000.10101000.00000010.00000010), ce qui donne une adresse réseau 11000000.10101000.00000000.00000000 soit en décimal **192.168.0.0**
   c. Cette interconnexion permet au réseau de mieux résister en cas de panne d'un routeur en proposant plusieurs chemins possibles. Par exemple, pour aller du routeur A au routeur E, il existe 2 chemins possibles : A-F-E ou A-B-E. Si le routeur F tombe en panne, il y a toujours la possibilité de passer par A-B-E

Partie B

1.
   a. entre A et E : A - B - E
   entre F et B : F - H - G - B ou F - D - A - B ou F- H - E - B ou F- D - G - 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	2
B	B	1
C	D	2
D	D	1
E	E	1
F	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.

le chemin serait E - H - F - D pour un coût de 10 + 0,1 + 0,1 = 10,2

<u>Exercice 2</u>

1.

a.

On obtient les données suivantes :

6	1.70	100

b.

```
SELECT nom, age
FROM animal
WHERE nom_espece = “bonobo”
ORDER BY age
```

2.

a.

clé primaire d'espece : "nom_espece" (chaque entrée est unique et peut donc jouer le rôle de clé primaire)
clé étrangère d'espece : "num_enclos" ("num_enclos" correspond à "num_enclos" de la relation enclos)

b.

```
animal (id_animal  : INT, nom : TEXT, age : INT, taille :
FLOAT, poids : INT, #nom_espece : TEXT)
enclos (num_enclos : INT, ecosysteme : TEXT, surface : INT,
struct : TEXT, date_entretien : DATE)
espece (nom_espece : TEXT,  classe : TEXT, alimentation :
TEXT, #num_enclos : INT)
```

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, animal.nom_espece
FROM animal
JOIN espece ON animal.nom_espece = espece.nom_espece
JOIN enclos ON espece.num_enclos = enclos.num_enclos
WHERE enclos.struct = 'vivarium' and alimentation =
'carnivore'
```

b.

```
SELECT COUNT(*)
FROM animal
JOIN espece ON animal.nom_espece = espece.nom_espece
WHERE classe = 'oiseaux'
```

Exercice 3

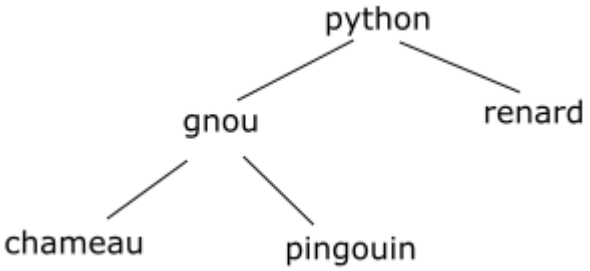
1.

résultat de l'exécution : Bonjour Alan
x et y sont de type booléen. x est False et y est True

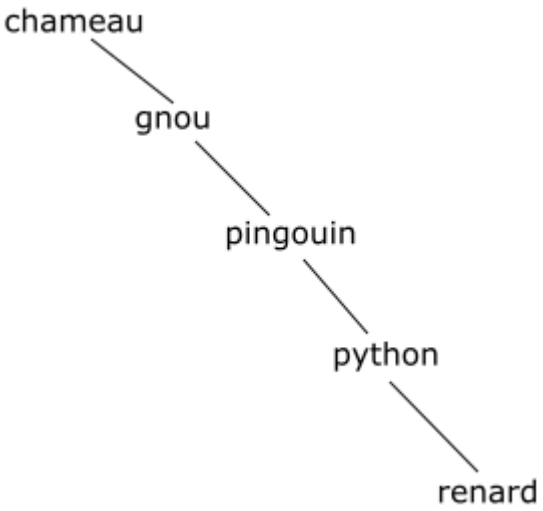
```
def occurrences_lettre(une_chaine, une_lettre):
    compteur = 0
    for l in une_chaine:
        if l == une_lettre:
            compteur = compteur + 1
    return compteur
```

2.

a.



b.



3.

a. La fonction mystere est une fonction récursive qui renvoie la taille de l'arbre. Dans l'exemple proposé, la fonction mystere renvoie donc 336531

b.

```
def hauteur(un_abr):
    if un_abr.est_vide():
        return 0
    else :
        return 1 + max(hauteur(un_abr.sous_arbre_gauche) \
            ,hauteur(un_abr.sous_arbre_droit))
```

4.

a.

```
def chercher_mots(liste_mots, longueur,lettre,position):
    res = []
    for i in range (len(liste_mots)):
        if len(liste_mots[i]) == longueur \
         and liste_mots[i][position] == lettre :
            res.append(liste_mots[i])
    return res
```

b.

Permet de trouver les mots de 3 lettres qui se terminent par ax (par exemple fax)

c.

```
chercher_mots(chercher_mots(chercher_mots(liste_mots_français,
5, 'r', 4), 5, 'e', 3), 5, 't', 2)
```