

Éléments de correction sujet 13 (2022)

Exercice 1

1. a -> non, car la mère peut accoucher de jumeaux (2 naissances avec le même idMere)
b -> oui, car le couple (date, rang) est unique
c -> non, car 2 bébés peuvent avoir la même taille et le même poids
2. Parce que la clé étrangère idMere de la relation Naissances est "liée" avec l'attribut numpatiente de la relation Patientes. La suppression d'une entrée dans la relation Patientes provoquera un problème dans la relation Naissances.
3.

```
INSERT INTO Patientes
VALUES
(13862, "Belanger", "Ninette", "La Rochelle")
```
4.

```
UPDATE Naissances
SET prenom = "Laurette"
WHERE idMere = 13860 AND prenom = "Lorette"
```
5.

```
SELECT nom, prenom
FROM Patientes
WHERE commune = "Aigrefeuille d'Aunis"
```
6.

```
SELECT AVG(poids)
FROM Naissances
JOIN TypesAccouchement ON acc = idAcc
WHERE libelleAcc = "césarienne"
```
- 7.

Berthelot	Michelle
Samson	Marine
Baugé	Gaëlle
Baugé	Gaëlle

Exercice 2

1. `attente.append((50,4))`

2. a. le tri par sélection
ATTENTION : Malgré ce qui est indiqué dans l'énoncé, cette fonction ne renvoie pas la liste triée (elle renvoie None), elle modifie la liste passée en paramètre à la fonction.

b. quadratique : $O(n^2)$

3. a.

```
def quitte(attente):  
    return [a for a in attente if a[1] != 1]
```

b.

```
def maj(attente):  
    t = []  
    for a in attente:  
        t.append((a[0], a[1]-1))  
    return t
```

4. a.

```
def priorite(attente,p):  
    for a in attente:  
        if a[0] == p:  
            return a[1]
```

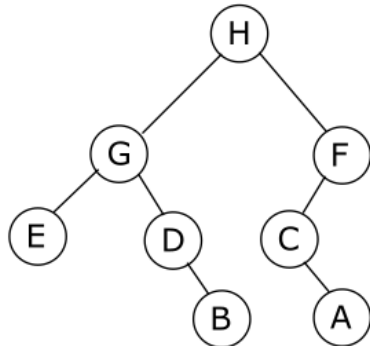
ATTENTION : si p n'est pas présent, la fonction renvoie None, ce qui n'est pas forcément une bonne chose, mais l'énoncé ne prévoit rien dans ce cas.

b.

```
def revise(attente, p):  
    nouvelle = []  
    n = priorite(attente, p)  
    for (patient, prio) in attente:  
        if patient == p:  
            nouvelle.append((patient, 1))  
        elif prio < n:  
            nouvelle.append((patient, prio+1))  
        else :  
            nouvelle.append((patient, prio))  
    return nouvelle
```

Exercice 3

1. taille = 11 ; hauteur = 5
2.
 - a. arbre 2sujet
 - b.



3.
 - a. parcours suffixe :
d
b
g
f
a
 - b.

```
def parcours_maladies(arb):  
    if arb=={}:  
        return None  
    parcours_maladies (arb['sag'])  
    parcours_maladies (arb['sad'])  
    if arb['sag'] == {} and arb['sad'] == {}:  
        print(arb['etiquette'])
```

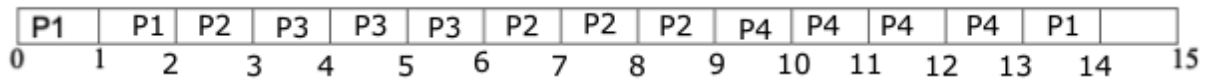
4.

```
def symptomes(arb, mal):  
    if arb['sag'] != {}:  
        symptomes(arb['sag'], mal)  
    if arb['sad'] != {}:  
        symptomes(arb['sad'], mal)  
    if arb['etiquette'] == mal:  
        arb['surChemin'] = True  
        print('symptômes de', arb['etiquette'],':')  
    else :  
        if arb['sad'] != {} and arb['sad']['surChemin']:  
            print(arb['etiquette'])  
            arb['surChemin'] = True  
        if arb['sag'] != {} and arb['sag']['surChemin']:  
            print('pas de ',arb['etiquette'])  
            arb['surChemin'] = True
```

Exercice 4

Partie A

1.



2.

Processus	Temps d'exécution	Instant d'arrivée	Temps de séjour	Temps d'attente
P1	3	0	$14-0=14$	$14-3=11$
P2	4	2	$9-2=7$	$7-4=3$
P3	3	3	$6-3=3$	$3-3=0$
P4	4	5	$13-5=8$	$8-4=4$

3.

S'il commence dès qu'il arrive et s'il n'est pas interrompu par un autre processus (processus prioritaire)

Partie B

1.

L'analyseur d'échantillon attend D4 car D4 est utilisée par le SGBD
Le SGBD attend D5 car D5 est utilisée par le tableur
Le traitement de texte attend D3 car D3 est utilisée par le tableur
Le tableur attend D1 car D1 est utilisée par le tableur

2.

Cette situation s'appelle un interblocage (deadlock en anglais)

3.

Tableur (parce que D1 est libre) - Traitement de texte (parce que le tableur a libéré D3) - SGBD (car le tableur a libéré D5) - Analyseur d'échantillon (car le SGBD a libéré D4)

Exercice 5

Partie A

1. adresse réseau : 192.168.1.0 ; masque : 255.255.255.0 (hors programme)
2. adresses interfaces réseaux de R5 : 192.168.5.254 ; 175.89.50.254 ; 44.197.5.1
3.
 - a. 192.168.1.1 à 192.168.1.254
 - b. 254 machines adressables

Partie B

1. SP -> R5 -> R1 -> R0 -> RL R
2. SP -> R5 -> R4 -> R2 -> R0 -> RL R

Partie C

1.
 $C = 10^9 / 400 \cdot 10^6 = 2,5$ donc $C = 3$
2.
 $BP = 10^9 / 5 = 2 \cdot 10^8$ b/s = 200 Mb/s
3.
SP -> R5 -> R4 -> R2 -> R1 -> R0 -> RL R pour un coût total $C = 1+4+1+2=8$
4.
SP -> R5 -> R4 -> R3 -> R0 -> RL R pour un coût total $C = 1+5+4=10$