

Proposition de correction

Exercice 1

Q1

```
def __init__(self, nature, surface, prix_moy):
    self.nt = nature
    self.sf = surface
    self.pm = prix_moy
```

Q2

140000.0 (70.0 x 2000.0)

Q3

```
def estim_prix(self):
    prix_brut = self.sf * self.pm
    if self.nt == 'maison':
        return prix_brut * 1.1
    elif self.nt == 'bureau':
        return prix_brut * 0.8
    else :
        return prix_brut
```

Q4

```
def nb_maison(lst : list) -> int:
    total = 0
    for l in lst :
        if l.nt == 'maison':
            total = total + 1
    return total
```

Q5.a

parcours infixe : b2, b4, b1, b5, b3, b6

Q5.b

```
def contient(surface, abr):
    if abr.est_vide():
        return False
    elif abr.get_v().sf >= surface:
        return True
    else :
```

```
return contient(surface, abr.get_d())
```

Exercice 2

Q1

R2

Q2.a

```
SELECT Client.nom, Client.avis
FROM Client, Reservation
WHERE Reservation.jour = '2021-06-05' AND Reservation.heure = '19:30:00'
AND Reservation.idClient = Client.idClient
ORDER BY Client.nom
```

Q2.b

```
SELECT DISTINCT Plat.nom
FROM Plat, Reservation, Commande
WHERE (Plat.categorie = 'plat principal' OR Plat.categorie = 'dessert')
AND Reservation.jour = '2021-04-12'
AND Plat.idPlat = Commande.idPlat
AND Reservation.idReservation = Commande.idReservation
ORDER BY Plat.nom
```

Q3

ajoute un plat d'identifiant 58, ayant pour nom "Pêche Melba", pour catégorie "dessert", pour description "Pêches et glace vanille" et pour prix 6,5€.

Q4.a

```
DELETE FROM Commande
WHERE idReservation = 2047
```

Q4.b

```
UPDATE Plat
SET prix = prix * 1.05
WHERE prix < 20.0
```

Exercice 3

Q1.a

- L1 : 192.168.1.0/24
- L2 : 172.16.0.0/16

Q1.b

- L1 : 192.168.1.1 à 192.168.1.254
- L2 : 172.16.0.1 à 172.16.255.254

Q1.c

- L1 : $254 = 2^8 - 2$
- L2 : $65534 = 2^{16} - 2$

Q2.a

en cas de panne ou de surcharge d'un routeur

Q2.b

R1 → R2 → R5 → R6 : 2 sauts

Q2.c

R1 → R2 → R3 → R4 → R5 → R6 : $10 + 1 + 1 + 1 + 10 = 23$

Q3

R5 :

IP réseau de destination	Passerelle suivante	Interface
10.1.3.0/24	10.1.3.2	Interface 1
10.1.4.0/24	10.1.4.2	Interface 2
10.1.6.0/24	10.1.6.2	Interface 3
10.1.7.0/24	10.1.7.1	Interface 4
172.16.0.0/16	10.1.7.2	interface 4
192.168.1.0/24	10.1.3.1	interface 1

R6 :

IP réseau de destination	Passerelle suivante	Interface
172.16.0.0/16	172.16.0.1	Interface 1
10.1.7.1/24	10.1.7.1	interface 2
192.168.1.0/24	10.1.7.1	interface 2

Exercice 4

Partie A

Q1

Programme 1 demande la table traçante et l'obtient.

Programme 2 demande le modem et l'obtient.

Programme 3 demande l'imprimante et l'obtient.

Programme 1 demande le modem et est en attente.

Programme 2 demande l'imprimante et est en attente.

Programme 3 demande la table traçante et est en attente.

→ tous les programmes s'attendent

Q2

demander (table traçante)

demander (imprimante)

exécution

libérer (imprimante)

libérer (table traçante)

Q3

bloqué

Partie B

Q1

ps -ef

Q2

831

Q3

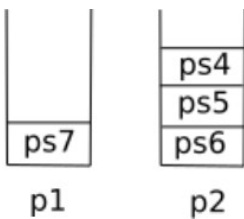
6211

Exercice 5**Q1**

File

Q2

```
def ajouter(lst : list, proc : object):  
    lst.append(proc)
```

Q3**Q4.a**

```
def est_vide(f : tuple) -> bool:  
    p1, p2 = f  
    return pile_vide(p1) and pile_vide(p2)
```

Q4.b

```
def enfiler(f : tuple, elt : object):  
    empiler(f[0], elt)
```

Q4.c

```
def defiler(f : tuple) -> object:  
    p1, p2 = f  
    if pile_vide(p2):  
        while not pile_vide(p1):  
            empiler(p2, depiler(p1))  
    return depiler(p2)
```