

Proposition de correction

Exercice 1

Q1

Arbre 1

Q2.a

Sous arbre gauche

Q2.b

```
RechercheValeur : clé, ABR → Booleen
début
  si est_vide(ABR) alors
    renvoyer Faux
  si racine(ABR) = clé alors
    renvoyer Vrai
  si racine(ABR) < clé alors
    renvoyer RechercheValeur(sous_arbre_droit(ABR))
  sinon
    renvoyer RechercheValeur(sous_arbre_gauche(ABR))
fin
```

Q3.a

parcours infixé

Q3.b

7-2-1-5-3-6-10-8-9

Q3.c

1-3-6-5-2-9-8-10-7

Q3.d

7-2-10-1-5-8-3-6-9

Exercice 2

Q1.a

5

Q1.b

Les goélands

Q1.c

```
def surface(self) :  
    return sum(self.b.sup(), self.c.sup(), self.d.sup())
```

Q2

```
for villa in v :  
    if villa.equip() == "eq" :  
        print(villa.nom())
```

Q3

appel d'une fonction par elle même

Q4

```
def max_surface(v) :  
    if len(v) == 1 :  
        return v[0]  
    if v[0].surf() <= v[1].surf() :  
        v.pop(0)  
    else :  
        v.pop(1)  
    return max_surface(v)
```

Exercice 3

Q1

identifie de façon unique un enregistrement

Q2

Type(Type_objet, Libelle_objet)

Q3

b

Q4

la clef primaire 'BD' existe déjà dans la table

Q5

2 768,067
11 92,753

Q6

```
SELECT Gaia.Nom_Systeme, Gaia.Nom_objet, Type.Libelle_Objet
FROM Gaia, Type
WHERE Gaia.Parallaxe > 400 AND Gaia.Type_Objet = '*'
AND Gaia.Type_Objet = Type.Type_Objet
```

Q7.a

INSERT INTO Type VALUES('ST', 'Etoile')

Q7.b

1. insérer la nouvelle clef dans la table Type : cf Q7.a
2. mettre à jour la valeur de la clef étrangère de la table Gaia : UPDATE Gaia SET Type_Objet = 'ST' WHERE Type_Objet = '*'
3. supprimer la clef primaire de la table Type : DELETE FROM Type WHERE Type_Objet = '*'

Exercice 4

Q1

a

Q2

192.168.10.2

Q3

$2^{24} - 2 = 16.777.214$

Q4

relier des machines d'un même réseau

Q5

relier des machines de réseaux différents

Q6

Table routage R1		
Destination réseau	Passerelle	Métrique
192.168.10.0/24	0.0.0.0	0
2.100.40.0/24	2.100.40.1	1
3.100.30.0/24	3.100.30.2	1
4.10.10.0/24	4.10.10.2	1
4.20.10.0/24	3.100.30.2	1
7.30.40.0/24	3.100.30.2	3
6.10.30.0/24	4.10.10.2	2
90.10.20.0/24	2.100.40.1	2

Q7

Table routage R1		
Destination réseau	Passerelle	Métrique
90.10.20.0/24	4.10.10.2	4

Exercice 5**Q1**

2 : FIFO

Q2.a

	V	F	val
1 :	0	Client4,Prioritaire,Client3,Client2,Client1	?
2 :	0	Client4,Prioritaire,Client3,Client2	Client1
4 :	Client1	Client4,Prioritaire,Client3,Client2	Client1
5 :	Client1	Client4,Prioritaire,Client3	Client2
4 :	Client1,Client2	Client4,Prioritaire,Client3	Client2
5 :	Client1,Client2	Client4,Prioritaire	Client3
4 :	Client1,Client2,Client3	Client4,Prioritaire	Client3
5 :	Client1,Client2,Client3	Client4	Prioritaire

Q2.b

```

def longueur_File(F) :
    V = creer_file_vide()
    n = 0
    while not est_vide(F) :
        n = n + 1
        val = defiler(F)
        enfiler(V, val)
    while not est_vide(V) :
        val = defiler(V)
        enfiler(F, val)
    return n

```

Q2.c

```

def compter_prio(F : object) -> int :
    """ compte le nombre de personnes prioritaires dans une liste
    @param      F -- liste de personnes
    @return     nombre de personnes prioritaires
    """
    V = creer_file_vide()
    n = 0
    while not est_vide(F) :
        val = defiler(F)
        enfiler(V, val)
        if val == "prioritaire" :
            n = n + 1

```

```
while not est_vide(V) :  
    enfiler(F, defiler(V))  
return n
```