Proposition de correction

Exercice 1

Partie A

Q1.a

les nom, prénom et numéro de toutes les joueuses de l'équipe de France

Q1.b

```
SELECT nom, prenom
FROM JOUEUSE
WHERE age >= 30
ORDER BY nom, prenom
```

Q2.a

```
UPDATE JOUEUSE
SET age = 33
WHERE idjoueuse = 101
```

Q2.b

```
INSERT INTO JOUEUSE
VALUES(105, "Angleterre", "Warm", "Suzanna", 29, 6)
```

Q3.a

```
SELECT nom
FROM JOUEUSE
JOIN SELECTION ON SELECTION.idjoueuse = JOUEUSE.idjoueuse
WHERE SELECTION.points >= 10
```

Q3.b

```
SELECT COUNT(SELECTION.idjoueuse)
FROM SELECTION, JOUEUSE
WHERE JOUEUSE.idjoueuse = 305
AND SELECTION.idjoueuse = JOUEUSE.idjoueuse
```

Q3.c

```
SELECT MATCH.stade
FROM MATCH, SELECTION, JOUEUSE
WHERE MATCH.idmatch = SELECTION.idmatch
AND SELECTION.idjoueuse = JOUEUSE.idjoueuse
```

AND JOUEUSE.idjoueuse = 305

ORDER BY MATCH.stade

Partie B

Q4

G → E → C → B

Q5

Table du routeur F de l'Amérique du Sud		
Destination	Routeur suivant	distance
A	A	1
B	A	3
C	A	2
D	D	1
E	D	2
G	D	2

Q6

panne routeur D (Europe)

passage obligatoire par routeur C (Asie)

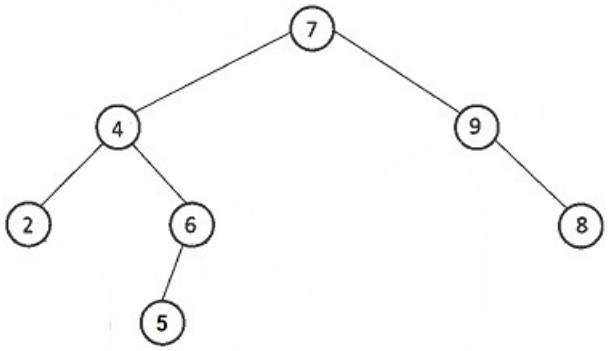
Exercice 2

Partie A

Q1

- Hauteur = 4
- Taille = 12

Q2.a



Q2.b

```
cons(12,
    cons(10
```

```
            cons(8, arbre_vide(), arbre_vide()),
            cons(11, arbre_vide(), arbre_vide())
            ),
    con(14
            arbre_vide(),
            cons(16, arbre_vide(), arbre_vide())
            )
    )
```

Q3.a

10, 15, 7, 17, 8, 11, 4, 20, 6, 12, 14, 9

Q3.b

proposition 2

Partie B

Q6

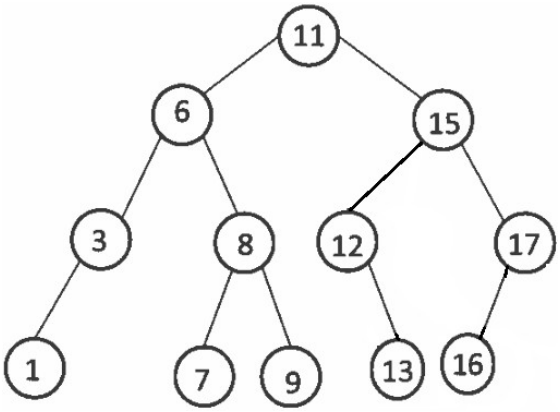
Arbre 1 : 9 < 11

Arbre 2 : 4 < 5

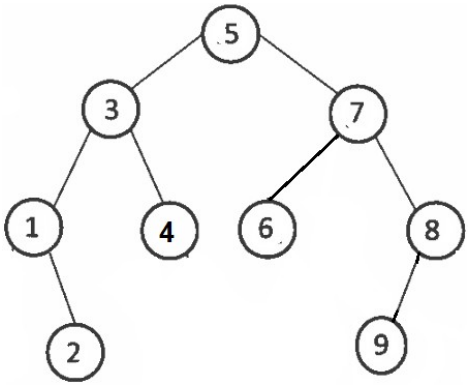
Q5

parcours infixe : sous arbre gauche ≤ racine < sous arbre droit

Q6.a



Q6.b



Q7

```
def tri(T : list) -> list :
    """ Tri le tableau de nombres T. """
    Abr = arbre_vide()
    for item in T :
            inserer_dans_ABR(Abr, item)
    return parcours(Abr)
```

Exercice 3

Q1.a

ls

Q1.b

Propositions : 1, 4

Q1.c

cp fich2.txt /home/camille/devoirs

Q2.a

camille n'a pas les droits d'écriture (w)

Q2.b

chmod u+x go-r algo.py

Q3.a

```
def classement_type(L) :
    photos = []
    sons = []
    videos = []
    for fichier in L :
            if fichier["type"] == "photo" :
                    photos.append(fichier)
            elif fichier["type"] == "son" :
                    sons.append(fichier)
            else :
                    videos.append(fichier)
    return photos, sons, videos
```

Q3.b

```
def photo_juillet(L : list) -> int :
    nb_photos = 0
    for fichier in L :
            if ( fichier["type"] == "photo" ) and ( fichier["mois"] ==7 ) :
                    nb_photos += 1
    return nb_photos
```

Q3.c

algorithme glouton : on enregistre les plus petits fichiers en premier jusqu'à saturation du disque

précondition : la liste des fichiers doit être trié par ordre de taille croissante