

Proposition de correction

Exercice 1

Q1

Nœud B	
Destination	Coût
A	1
C	3
D	2
E	2
F	2
G	1
H	2

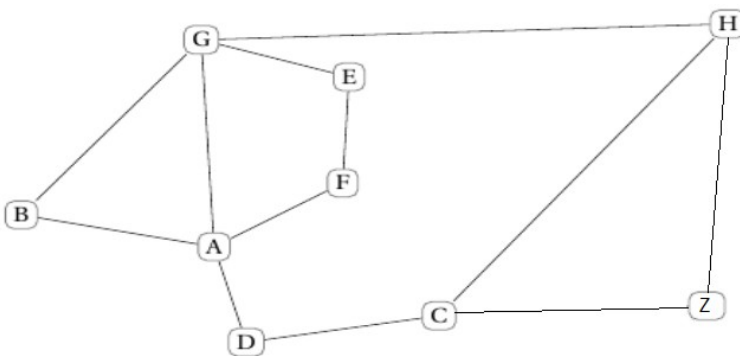
Nœud F	
Destination	Coût
A	1
B	2
C	3
D	2
E	1
G	2
H	3

Q2

F → E → G → H

F → A → G → H

Q3



Q4

B → G → E → F → A → D → C → H

coût minimal = 1 + 1 + 1 + 1 + 10 + 10 + 10 = 34

Exercice 2

Q1.a

Identifie de façon unique un enregistrement

Q1.b

la clef primaire de valeur 3 est déjà utilisée

Q1.c

Fusee (id_fusee : INT, modele : TEXT, constructeur : TEXT, nb_places : INT)

Q2.a

2

Q2.b

```
SELECT modele, constructeur
FROM Fusee
WHERE nb_places > 3
ORDER BY modele, constructeur
```

Q2.c

```
SELECT nom, prenom
FROM Astronaute
ORDER BY nom
```

Q3.a

```
INSERT INTO Vol VALUES(5, 3, '2003-04-12');
INSERT INTO Equipe VALUES(5, 1);
INSERT INTO Equipe VALUES(5, 4);
```

Q3.b

```
SELECT Astronaute.nom, Astronaute.prenom
FROM Astronaute, Vol, Equipe
WHERE Vol.date = '2022-10-25'
AND Equipe.id_vol = Vol.id_vol
AND Astronaute.id_astronaute = Equipe.id_astronaute
ORDER BY Astronaute.nom, Astronaute.prenom
```

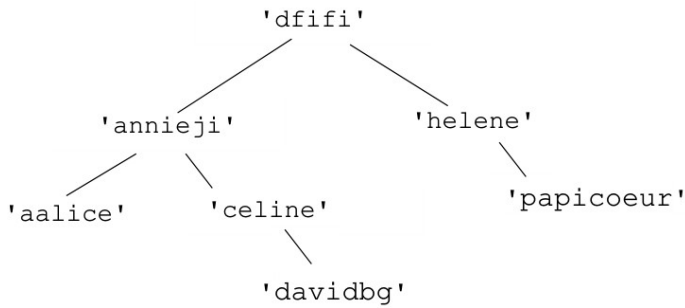
Exercice 3

Partie 1

Q1

Taille = 5, hauteur = 3

Q2



Q3

C

Q4

```

def present(self, identifiant):
    if self.est_vide():
        return False
    elif self.racine() == identifiant:
        return True
    elif self.racine() < identifiant:
        return self.sd().present(identifiant)
    else:
        return self.sg().present(identifiant)
  
```

Partie 2

Q5.a

False

Q5.b

'bac'	'nsi'	'2023'
-------	-------	--------

Q5.c

'poule'	'python'	'castor'
---------	----------	----------

Q6

```
def longueur(f):
    resultat = 0
    g = creer_file()
    while not est_vide(f) :
        elt = defiler(f)
        resultat = resultat + 1
        enfiler(g, elt)
    while not(est_vide(g)):
        enfiler(f, defiler(g))
    return resultat
```

Q7

2!@59fgds

Q8

```
def ajouter_mot(f : file f, mdp : str) :
    """
    @param      f -- file qui a au plus 3 éléments
    @param      mdp -- chaine de caractères valide
    """
    if (longueur(f) < 4) and est_valide(mdp):
        enfiler(f, mdp)
        if longueur(f) > 3 :
            defiler(f)
```

Q9

```
def mot_file(f, mdp):
    g = creer_file()
    present = False
    while not(est_vide(f)):
        elt = defiler(f)
        enfiler(g, elt)
        if elt == mdp:
            present = True
    while not(est_vide(g)):
        enfiler(f, defiler(g))
    return present
```

Q10

```
def modification(f : file, nv_mdp : str) :
    if (not mot_file(nv_mdp)) and est_valide(mdp) :
        ajouter_mot(f, nv_mdp)
        return True
    return False
```