Baccalauréat Général

Session 2023

Épreuve : **Numérique et sciences de l'ingénieur**

Durée de l'épreuve : 3h30

Coefficient : 16

PROPOSITION DE CORRIGÉ

EXERCICE 1 :

Question 1 :

Noeud B

A	1
C	3
D	2
E	2
F	2
G	1
H	2

Noeud F

A	1
B	2
C	3
D	2
E	1
G	2
H	3

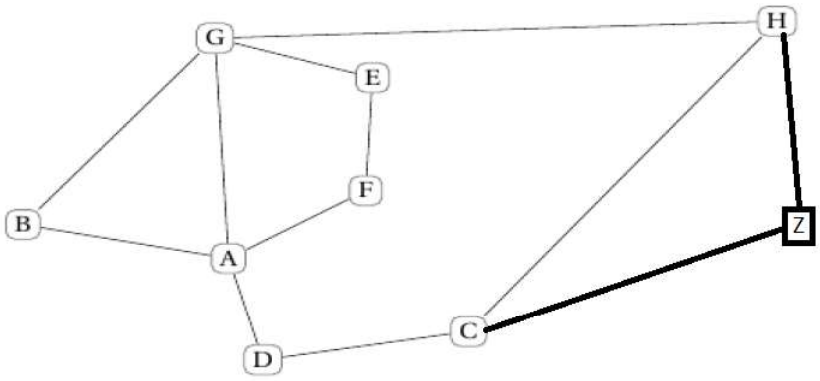
Question 2 :

Il y a deux chemins possibles en appliquant le protocole RIP : avec 3 sauts

F → E → G → H

F → A → G → H

Question 3 :



Question 4 :

Le chemin pris par un paquet d'origine B à destination de H est :

B → G → E → F → A → D → C → H de coût 34, on le trouve en appliquant

l'algorithme de Dijkstra.

EXERCICE 2 :

Question 1 :

a) La clé primaire est la donnée permet d'identifier de façon unique chaque enregistrement d'une table.

b) la requête renvoie une erreur car la valeur 3 est déjà utilisée pour identifier un enregistrement, il ne peut pas y avoir de doublon.

c) Fusee (id fusee : INT, modele : VARCHAR, constructeur : VARCHAR, nb_places : INT)

Question 2 :

a) La requête renvoie le nombre de fusées construites par SpaceX ici 2.

b)

SELECT modele, constructeur

FROM Fusee

WHERE nb_places ≥ 4

c)

SELECT nom, prenom

FROM Astronautes

ORDER BY nom

Question 3 :

a)

INSERT INTO Vol VALUES(5, '12/04/2023');

INSERT INTO Equipe VALUES(5,1) ;

INSERT INTO Equipe VALUES(5,4);

b)

```
SELECT
AST.nom
AST.prenom
VOL.date
FROM
        Astronaute AS AST
        INNER JOIN Equipe AS EQU
                ON EQU.id_Astronaute = AST.id_Astronaute
        INNER JOIN Vol AS VOL
                ON VOL.id_vol=EQU.id_vol
WHERE
        VOL.date=’25/10/2022’
```

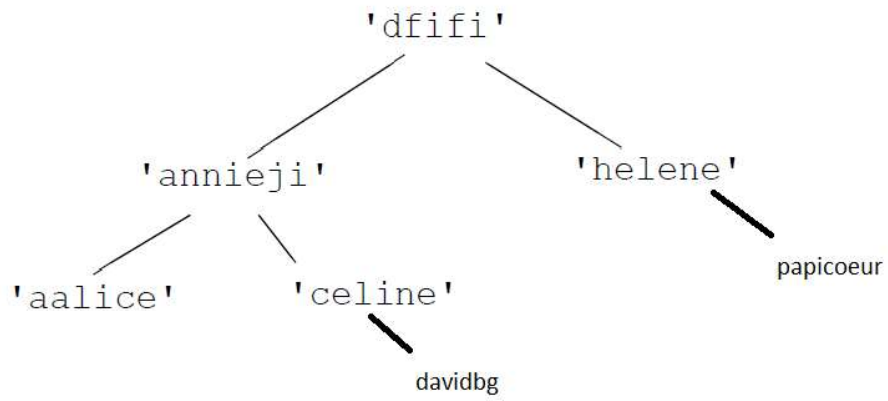
EXERCICE 3 :

Question 1 :

taille : 5

Hauteur : 3

Question 2 :



Question 3 :

Le parcours infixe permet de retrouver l'ordre lexicographique. Réponse C.

Question 4 :

```
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)
```

Question 5 :

a)

est_vide(f1) renvoie ‘False’

b)

‘bac’ ‘nsi’ ‘2023’

c)

‘poule’ ‘python’ ‘castor’

Question 6 :

```
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
```

Question 7 :

Le mot de passe validé est : **C** - `'2!@59fgds'`

Question 8:

```
def ajouter_mot(f, mdp) :
        if longueur(f) == 3 :
            defile(f)
        else :
            enfile(f, mdp)
```

Question 9:

```
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
```

Question 10 :

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