

[← Index des sujets 2023](#)

23-NSIJ1G11 : Corrigé

Année : **2023**

Centre : **Centres-Etrangers**

Jour : **J1**

Enoncé : [!\[\]\(e3f8612927870f2e0f9f5989e6dd3064_img.jpg\) PDF](#)

1. Exercice 1 (3 points)

SQL

1.

2. a.

theme	annee
Le système d'enseignement supérieur français est-il juste et efficace ?	2022
Trois innovations pour la croissance futurez (1/3) : la révolution blockchain	2021

b.

Requête SQL

```
SELECT theme
FROM podcast
WHERE annee=2019
```

c.

Requête SQL

```
SELECT theme, annee
FROM podcast
ORDER BY annee ASC
```

3. a. La requête affiche les divers thèmes présent dans la table podcast sans doublon.

b.

Requête SQL

```
DELETE
FROM podcast
WHERE id_podcast=40
```

4. a.

 Requête SQL

```
UPDATE emission
SET animateur='Emmanuel L.'
WHERE nom='Le Temps du débat'
```

b.

 Requête SQL

```
INSERT INTO emission (id_emission, nom, radio, animateur) VALUES (12850, 'Hastag', 'France Inter', 'Mathieu V.')
```

5.

 Requête SQL

```
SELECT podcast.theme, emission.nom, description.resume
FROM description
JOIN emission ON emission.id_emission = podcast.id_emission
JOIN podcast ON description.id_emission=podcast.id_emission
WHERE description.duree < 5
```

2. Exercice 2 (3 points)

Réseaux et protocoles de routages

1. a. On a :

- 164 = 10100100
 - 178 = 10110010
 - 2 = 00000010
 - 13 = 00001101
- On obtient donc 10100100.10110010.00000010.00001101

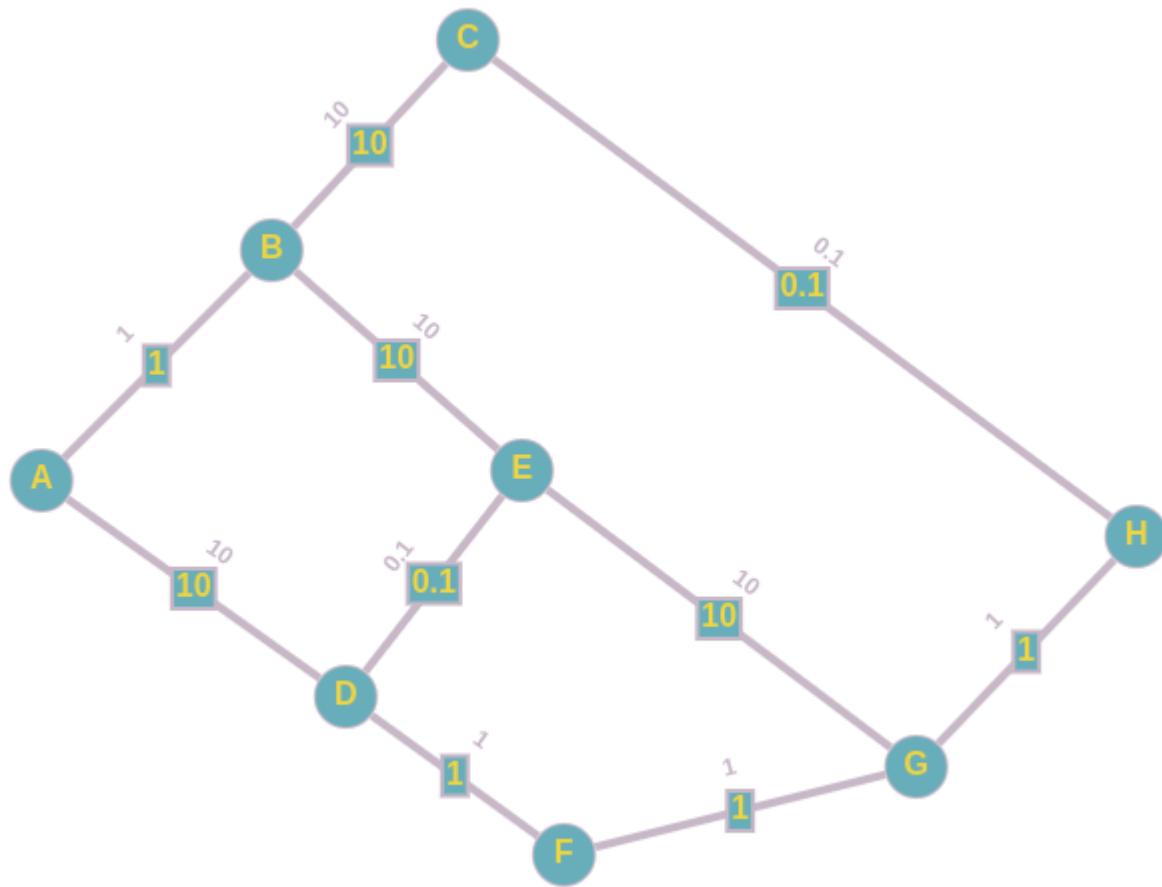
b. L'adresse en notation CIDR 164.172.2.13/24 signifie que les 24 premiers bits sont réservés pour l'adresse réseau, on a alors :

- 164.172.2.0 comme adresse réseau.

2. Protocole RIP :

- A - D - F - G
- A - D - E - G

- A - B - E - G



3. a.

b. Le protocole OSPF optimise le coût, donc ici on obtient :

- A - D - F - G soit un coût de $1 + 10 + 1 = 12$.

c. Avec le routeur F en panne, on obtient :

- A - B - C - H - G pour un coût de $1 + 10 + 0.1 + 1 = 12.1$.

3. Exercice 3 (6 points)

Structures de Files

1. Script Python

```

1 def ajout(f):
2     couleurs = ("bleu", "rouge", "jaune", "vert")
3     indice = randint(0, 3)
4     enfiler(f, couleur[indice])
5     return f
  
```

2. Script Python

```

def vider(f):
    while not est_vide(f):
        ...
  
```

```
defiler(f)
```

3. Script Python

```
1 def affich_seq(sequence):
2     stock = creer_file_vide()
3     ajout(sequence)
4     while not est_vide(sequence):
5         c = defiler(sequence)
6         affichage(c)
7         time.sleep(0.5)
8         enfiler(stock, c)
9     while not est_vide(stock):
10        enfiler(sequence, defiler(stock))
```

4. a.

Script Python

```
1 def tour_de_jeu(sequence):
2     affich_seq(sequence)
3     stock = creer_file_vide()
4     while not est_vide(sequence):
5         c_joueur = saisie_joueur()
6         c_seq = defiler(sequence)
7         if c_joueur == c_seq:
8             enfiler(stock, c_seq)
9         else:
10            vider(sequence)
11     while not est_vide(stock):
12        enfiler(sequence, defiler(stock))
```

5. b. Question bizarre...

Script Python

```
1 def tour_de_jeu_modifie(sequence):
2     while True:
3         affich_seq(sequence)
4         stock = creer_file_vide()
5         while not est_vide(sequence):
6             c_joueur = saisie_joueur()
7             c_seq = defiler(sequence)
8             if c_joueur == c_seq:
9                 enfiler(stock, c_seq)
10            else:
11                vider(sequence)
12                vider(stock)
13            while not est_vide(stock):
14                enfiler(sequence, defiler(stock))
```

ou bien

Script Python

```
1
```

```
2 def tour_de_jeu_modifie(sequence):
3     affich_seq(sequence)
4     stock = creer_file_vide()
5     while not est_vide(sequence):
6         c_joueur = saisie_joueur()
7         c_seq = defiler(sequence)
8         if c_joueur == c_seq:
9             enfile( stock, c_seq)
10        else:
11            vider(sequence)
12            print("Perdu ! On rejoue !")
13            tour_de_jeu_modifie(sequence)
14    while not est_vide(stock):
15        enfile( sequence, defiler(stock))
tour_de_jeu_modifie(sequence)
```