

# Proposition de correction

## Exercice 1

---

### Q1

MySQL

### Q2.a

titrealbum

### Q2.b

elle garantit l'unicité des enregistrements

### Q2.c

permet de mettre en relation les tables entre elles

### Q2.d

voir Annexe

### Q3.a

```
SELECT * FROM Artiste
```

```
WHERE artiste = 'Black Eyed Peas'
```

### Q3.b

```
SELECT album.albumId, album.titrealbum, artiste.artisteld, artiste.artiste
```

```
FROM album, artiste
```

```
WHERE artiste.artiste = 'Black Eyed Peas'
```

```
AND album.artisteld = artiste.artisteld
```

```
ORDER BY album.albumId
```

### Q3.c

```
INSERT INTO album
```

```
VALUES(2500, 'Translation', 169)
```

### Q4.a

2 Minutes To Midnight

### Q4.b

le critère de recherche porte sur une valeur exacte et non sur une chaîne textuelle qui contient la valeur

## Exercice 2

### Q1.a

- 0000 10010
- 0001 00001

### Q1.b

1. Erreur de double bit : Si deux bits sont inversés dans le même mot de données, la parité ne sera pas altérée.
2. Erreur de nombre impair de bits altérés : Si un nombre impair de bits est altéré, la parité restera inchangée et l'erreur ne sera pas détectée.

### Q2

```
def calcul_parite(Liste_bits : list) -> int:
    """
    @param Liste_bits -- liste constituée d'entiers valant 0 ou 1
    @return 1 ou 0 correspondant à son bit de parité (paire)
    """
    bit_a_un = 0
    for bit in Liste_bits:
        if bit == 1:
            bit_a_un += 1
    return int(bit_a_un % 2)
```

### Q3.a

- La position P1 couvre les bits 1, 3, 5, 7 : [1, 0, 0, 1], ce qui donne 0.
- La position P2 couvre les bits 2, 3, 6, 7 : [0, 0, 0, 1], ce qui donne 0.
- La position P3 couvre les bits 4, 5, 6, 7 : [1, 0, 0, 1], ce qui donne 1.

Après insertion : [0, 0, 1, 1, 0, 0, 1]

### Q3.b

```
def codage_hamming(donnees : list) -> list:
    """
    @param donnees -- liste de quatre bits valant 0 ou 1
    @return liste de sept bits contenant le message encodé selon le codage de Hamming.
    """
    if len(donnees) != 4:
        return []
    p1 = calcul_parite([donnees[0], donnees[1], donnees[3]])
    p2 = calcul_parite([donnees[0], donnees[2], donnees[3]])
    p3 = calcul_parite([donnees[1], donnees[2], donnees[3]])
    return [p1, p2, donnees[0], p3, donnees[1], donnees[2], donnees[3]]
```

### Q4.a

$c1 : [p3, d2, d3, d4] = [1, 0, 0, 1] \rightarrow 0$

$c2 : [p2, d1, d3, d4] = [0, 0, 0, 1] \rightarrow 1$

$c3 : [p1, d1, d2, d4] = [0, 0, 0, 1] \rightarrow 1$

### Q4.b

	Pas de bit altéré	p <sub>1</sub> altéré	p <sub>2</sub> altéré	d <sub>1</sub> altéré	p <sub>3</sub> altéré	d <sub>2</sub> altéré	d <sub>3</sub> altéré	d <sub>4</sub> altéré
[c <sub>1</sub> , c <sub>2</sub> , c <sub>3</sub> ]	[0, 0, 0]	[0, 0, 1]	[0, 1, 0]	[0, 1, 1]	[1, 0, 0]	[1, 0, 1]	[1, 1, 0]	[1, 1, 1]

### Q4.c

en fonction de [c<sub>1</sub>, c<sub>2</sub>, c<sub>3</sub>], il est possible de connaître le bit altéré et de le corriger.

## Exercice 3

### Q1.a

identifiant unique attribué à une interface réseau composé de l'ID constructeur et du n° série de l'interface

### Q1.b

192.168.7.130/24 (routeur B)

### Q1.c

le routeur (passerelle par défaut). Les autres appareils (caméra, imprimante, PC) apparaîtront dans la table ARP uniquement si le téléphone a communiqué avec eux.

### Q2.a

- A → C → D : 2 sauts
- A → B → E → D : 3 sauts

### Q2.b

voir annexe

### Q2.c

voir Annexe

### Q2.d

1. Détection de la Panne : Chaque routeur envoie régulièrement des messages de mise à jour de routage (habituellement toutes les 30 secondes). Si un routeur ne reçoit pas de mises à jour de la part du routeur A, il considère que le routeur A est inaccessible.

2. Propagation de la Mise à Jour : Les routeurs voisins de A mettront la distance vers A à 16 (considérée comme "infinie" dans RIP) dans leurs tables de routage, indiquant que A est inaccessible.
3. Propagation des Changements : Les routeurs voisins de A diffusent leurs tables de routage mises à jour à tous les autres routeurs dans le réseau.
4. Réception des Mises à Jour : Chaque routeur qui reçoit ces mises à jour ajuste sa table de routage. Ils marquent la route vers A comme inaccessible et recalculent les chemins alternatifs pour les destinations qui étaient atteignables via A.
5. Convergence du Réseau : Les routeurs continuent d'échanger des messages de mise à jour jusqu'à ce que toutes les tables de routage soient cohérentes avec la nouvelle topologie du réseau.

### Q2.e

voir annexe

### Q3.a

$G \rightarrow H \rightarrow E$

### Q3.b

$G \rightarrow B \rightarrow A \rightarrow F \rightarrow E$

### Q3.c

- convergence beaucoup plus rapidement en cas de changement dans la topologie du réseau
- gestion de grands réseaux avec des milliers de routeurs
- envoi des paquets de contrôle uniquement lorsqu'il y a un changement dans la topologie

### Q4.d

voir annexe

# Annexe

## DOCUMENT RÉPONSE À RENDRE OBLIGATOIREMENT ET DANS SON INTÉGRALITE AVEC LA COPIE

### Exercice 1. Question 2) d.

**Indiquer** sans justification les valeurs des attributs permettant de compléter l'extrait de la table Titre proposé ci-dessous.

titreId	titre	#albumId	#formatId	#genreId	compositeurs	duree
2	'Paranoïd'	127	1	2	A. F. Iommi, W. Ward, T. Buttler, J. Osbourne	176352

### Exercice 3. Question 2) b.

**Compléter** le tableau des routes et des métriques respectives établies par chacun des routeurs pour rejoindre le routeur D.

Routeur	Destination	Passerelle à utiliser	Nombre de sauts
A	D	C	2
B		E	2
C		D	1
E		D	1
F		C	2
G		B	3
		F	3

**Exercice 3. Question 2) c.**

**Compléter** la table de routage complète du routeur B selon le protocole RIP.

Routeur	Destination	Passerelle à utiliser	Nombre de sauts
B	A	A	1
	C	A	2
	D	E	2
	E	E	1
	F	G	2
	G	G	1

**Exercice 3. Question 2) e.**

**Compléter** la nouvelle table de routage du routeur B après re-convergence suite à la panne du routeur A.

Routeur	Destination	Passerelle à utiliser	Nombre de sauts
B	C	G	3
	C	E	3
	D	E	2
	E	E	1
	F	G	2
	G	G	1

**Exercice 4. Question 3) d.**

**Compléter** la table de routage complète du routeur B après convergence selon le protocole OSPF.

Destination	Passerelle à utiliser	Métrieque
A	A	1
C	C	3
D	C	4
E	A	5
F	A	4
G	G	1
H	G	4