

Proposition de correction

Exercice 1

Q1.a

Identifier l'enregistrement par un n° unique

Q1.b

faire la relation entre les tables Benevole et Mission

Q2

peupler la table Benevole avec 2 enregistrements

Q3.a

```
SELECT montant
FROM Collecte
WHERE annee_ventes = 2020
ORDER BY montant
```

Q3.b

```
SELECT Benevole.nom, Benevole.prenom
FROM Benevole, Collecte
WHERE Collecte.annee_ventes = 2020
AND Collecte.montant > 50
AND Benevole.idbenevole = Collecte.idbenevole
ORDER BY Benevole.nom, Benevole.prenom
```

Q3.c

```
UPDATE Collecte
SET montant = 300
WHERE idbenevole = 1
AND annee_ventes = 2020
```

Q4.a

il manque le champ idpresence dans la requete

Q4.b

le champ nom n'existe pas dans la table Presence

il faut rajouter une clé étrangère idbenevole qui pointe sur la table Benevole

Exercice 2

Q1.a

```
def palindrome(mot : str) -> bool:
    """
    @brief  détermine si un mot est un palindrome
    @param mot -- un mot
    @return true si mot est un palindrome, false sinon
    """
    n = len(mot)
    est_palindrome = True
    for k in range(n//2):
        if mot[k] != mot[n-k-1]:
            est_palindrome = False
    return est_palindrome
```

Q1.b

linéaire $O(n)$

Q1.c

une fonction qui s'appelle elle-même

Q1.d

```
def palindrome_recursive(mot):
    if len(mot) <= 1:
        return True
    if mot[0] != mot[len(mot) - 1]:
        return False
    return palindrome_recursive(mot[1:len(mot)-1])
```

Q2.a

```
def test_mot(mot : str, propose : str) -> str:
    """
    @param mot -- le mot à deviner en majuscules
    @param propose -- un mot proposé en majuscules
    @return le mot formé des lettres correctement placées et de tirets ailleurs
    """
    resultat = ""
    if len(mot) == len(propose):
        for c1, c2 in zip(mot, propose):
            if c1.upper() == c2.upper():
                resultat += c1
            else:
```

```

resultat += "-"
return resultat
    
```

Q2.b

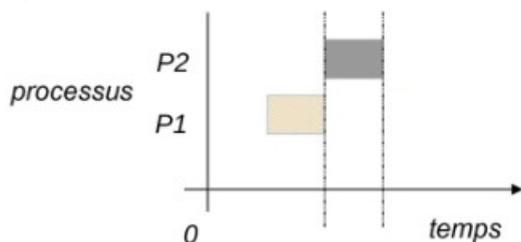
- Le compteur nb_coups n'est jamais incrémenté : la boucle peut devenir infinie si le mot n'est jamais deviné.
- Le mot proposé par le joueur doit être converti en majuscules dans la fonction test_mot()

Exercice 3

Q1.a

	Processus P1	Processus P2	compteur
Intervalle de temps 1	compteur1 ← 100		100
Intervalle de temps 2		compteur2 ← 100 compteur2 ← 100 – 1 compteur ← 99	99
Intervalle de temps 3	compteur1 ← 100 – 1 compteur ← 99		99

Q1.b



Q2.a

le mutex rend les opérations de chaque processus indivisibles

Q2.b

Processus P1	Processus P2
mutex.verrouillage() compteur ← compteur – 1 mutex.deverrouillage()	mutex.verrouillage() compteur ← compteur – 1 mutex.deverrouillage()

Q3

voir annexe

Annexe

DOCUMENT RÉPONSE À RENDRE OBLIGATOIREMENT ET DANS SON INTÉGRALITÉ AVEC LA COPIE

Exercice 3. Question 3)

Compléter le diagramme ci-dessous.

↑ indique la date d'arrivée des processus P1 à P4.

