Bac Général Maths + NSI (Mathématiques et Numérique et sciences informatiques)

CORRIGÉ NON OFFICIEL

Proposition de corrigé rédigée par FormaV avec l'aide d'outils d'intelligence artificielle. Elle peut contenir des erreurs et ne remplace ni le corrigé ni le barème officiels de l'examen.

Les sujets officiels des sessions passées et nos sujets d'entraînement sont classés sur https://bac-maths-nsi.fr/annales-sujets-et-corriges

Document mis en ligne par FormaV®. Reproduction ou diffusion sans autorisation interdite.

BACCALAURÉAT GÉNÉRAL • CORRIGÉ DÉTAILLÉ

Numérique et sciences informatiques

Session 2023 • Jour 2

Code sujet **23-NSIJ2ME1** 3 h 30

Proposition de corrigé pédagogique • FormaV

Se repérer dans le corrigé

Exercice 1 — Arbres binaires, POO et récursivité — code Morse
8 réponses • sujet p. 1–2

Exercice 2 — Réseaux et protocole de routage
11 réponses • sujet p. 3–4

Exercice 3 — Bases de données et SQL
12 réponses • sujet p. 5–6

Les numéros, parties et pages renvoient au sujet original. Chaque réponse rappelle la consigne, explique la démarche et donne le résultat.

FormaV - document d'entraînement non officiel

Exercice 1

4 points

Arbres binaires, POO et récursivité — code Morse

Question 1.

Sujet p. 1

> Déterminer le code Morse du message « NSI » à l'aide de la figure de l'arbre binaire, en laissant un espace entre le code de chaque lettre. Les points sont représentés par « o » et les tirets par « - ».

On suit les branches depuis la racine, avec une branche gauche correspondant à `o` et une branche droite à `-`. Pour N, le chemin est droite puis gauche, donc `-o`. Pour S, il est gauche trois fois, donc `ooo`. Pour I, il est gauche deux fois, donc `oo`.

> **RÉPONSE À RETENIR**
>
> Le code Morse de `NSI` est `-o ooo oo`.

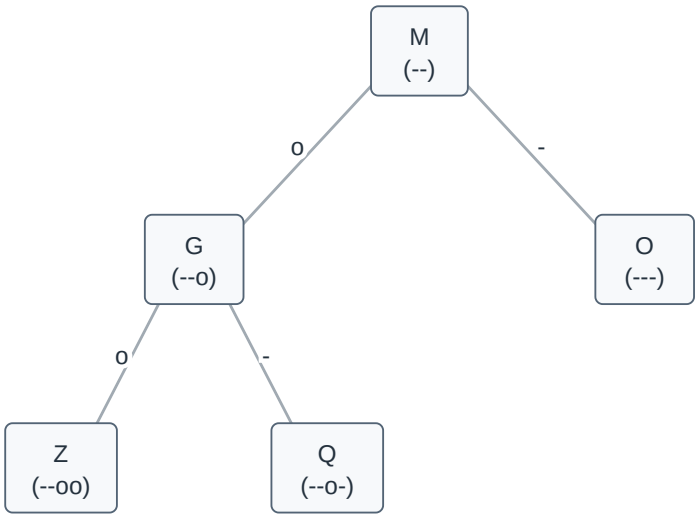
Question 2.

Sujet p. 1

> Représenter le sous-arbre binaire pour les lettres M, G, O, Z et Q à l'aide de l'extrait de la table du code Morse international : G : « --o », M : « -- », O : « --- », Q : « --o- », Z : « --oo ».

Les codes fournis indiquent que M est atteint par le préfixe `--`. Depuis M, la branche gauche (point `o`) mène à G, et la branche droite (tiret `-`) mène à O. Depuis G, la branche gauche mène à Z et la branche droite mène à Q.

Sous-arbre des lettres M, G, O, Z et Q



Les branches gauches correspondent à o et les branches droites à -. M est atteint après le préfixe --.

> **RÉPONSE À RETENIR**
>
> Le sous-arbre a M pour racine, G et O pour fils gauche et droit de M, puis Z et Q pour fils gauche et droit de G.

FormaV - document d'entraînement non officiel

Question 3.

Sujet p. 2

> Écrire les instructions à placer en lignes 14 et 15 pour créer les nœuds des lettres K et S, à partir de la déclaration de la classe Noeud et de l'extrait de la définition de l'arbre binaire.

Le chemin vers K est droite, droite, gauche : on part de la racine, on atteint T puis M, et on ajoute K comme fils gauche de M. Le chemin vers S est gauche, gauche, gauche : S est le fils gauche de I.

RÉPONSE À RETENIR

```
arbre.droite.droite.gauche = Noeud("K")
arbre.gauche.gauche.gauche = Noeud("S")
```

(a)

Question 4. (a)

Sujet p. 2

> Recopier le code de la fonction est_present(n, car) et compléter les lignes 2, 4 et 7.

Si `n` vaut `None`, le sous-arbre est vide et le caractère ne s'y trouve pas. Sinon, on compare la valeur du nœud avec `car`, puis on cherche dans les sous-arbres droit et gauche, dans cet ordre comme l'indique le squelette.

RÉPONSE À RETENIR

```
def est_present(n, car):
    if n == None:
        return False
    elif n.valeur == car:
        return True
    else:
        return est_present(n.droite, car) or est_present(n.gauche, car)
```

(b)

Question 4. (b)

Sujet p. 2

> La fonction est_present est-elle récursive ? Justifier votre réponse.

Une fonction est récursive si elle s'appelle elle-même. Dans le cas général, `est_present` appelle `est_present` sur les sous-arbres droit et gauche.

RÉPONSE À RETENIR

Oui, `est_present` est récursive.

(c)

Question 4. (c)

Sujet p. 2

> Déterminer quel type de parcours utilise la fonction est_present.

La recherche descend dans un sous-arbre avant de revenir poursuivre la recherche dans l'autre sous-arbre : c'est un parcours en profondeur. Le nœud courant est testé en premier, puis le sous-arbre droit, puis le sous-arbre gauche.

RÉPONSE À RETENIR

La fonction réalise un parcours en profondeur préfixe, dans l'ordre nœud courant, droite, gauche.

(a)

Question 5. (a)

Sujet p. 2

> Recopier et compléter les pointillés de la fonction code_morse(n, car), qui traduit un caractère car présent dans l'arbre n et renvoie son code Morse sous forme d'une chaîne de caractères.

Si le nœud courant contient le caractère recherché, il ne reste aucune branche à parcourir : on renvoie la chaîne vide. Sinon, le squelette teste si le caractère se trouve dans le sous-arbre gauche. Si c'est le cas, on préfixe le code obtenu par `o` ; sinon, comme le caractère est supposé présent dans l'arbre, il se trouve dans le sous-arbre droit et on préfixe par `-`.

RÉPONSE À RETENIR

```
def code_morse(n, car):
    if n.valeur == car:
        return ""
    elif est_present(n.gauche, car):
        return "o" + code_morse(n.gauche, car)
    else:
        return "-" + code_morse(n.droite, car)
```

Cette fonction suppose, comme le précise l'énoncé, que `car` est présent dans l'arbre.

(b)

Question 5. (b)

Sujet p. 1, 2

> Écrire une fonction morse_message qui reçoit un arbre de code Morse et un message sous forme d'une chaîne de caractères, et renvoie le message codé où chaque lettre est séparée par un trait vertical. Exemple : morse_message(arbre, 'PYTHON') renvoie « o--o|-o--|-|oooo|---|-o| ».

On parcourt les caractères du message et on calcule le code de chacun avec `code_morse`. L'exemple du sujet comporte un trait vertical après chaque code, y compris après la dernière lettre : on ajoute donc le séparateur à chaque itération.

RÉPONSE À RETENIR

```
def morse_message(arbre, message):
    resultat = ""
    for car in message:
        resultat += code_morse(arbre, car) + "|"
    return resultat
```

Pour le message `PYTHON`, le résultat est `o--o|-o--|-|oooo|---|-o|`, conformément à l'exemple du sujet.

Exercice 2

4 points

Réseaux et protocole de routage

Question 1.

Sujet p. 3

> Le poste PC LG 03 a pour adresse IPv4 192.168.162.4. Recopier et compléter cette adresse IP en binaire : 11000000.10101000.xxxxxxxx.xxxxxxxx.

On convertit chaque octet en binaire sur 8 bits. Les deux premiers octets sont déjà donnés : 192 = `11000000` et 168 = `10101000`. Pour le troisième, 162 = 128 + 32 + 2, donc `10100010`. Enfin, 4 s'écrit `00000100` sur 8 bits.

> **RÉPONSE À RETENIR**
>
> L'adresse complète est `11000000.10101000.10100010.00000100`.

(a)

Question 2. (a)

Sujet p. 3

> Le poste PC Admin 02 du secteur Administration a pour adresse IPv4 192.168.16.12/24. Donner l'adresse du réseau local dédié au secteur Administration et son masque de sous-réseau.

Le préfixe `/24` indique que les 24 premiers bits forment la partie réseau. Pour trouver l'adresse réseau, on met à 0 les bits de la partie hôte, ici le dernier octet : `192.168.16.0`. Le masque comporte 24 bits à 1 puis 8 bits à 0, soit `255.255.255.0`.

> **RÉPONSE À RETENIR**
>
> Adresse du réseau : `192.168.16.0`. Masque de sous-réseau : `255.255.255.0`, soit `/24`.

(b)

Question 2. (b)

Sujet p. 3

> Donner l'adresse de diffusion (appelée aussi broadcast) du réseau du secteur Administration.

Dans un réseau `/24`, le dernier octet correspond à la partie hôte. Pour l'adresse de diffusion, ses 8 bits sont mis à 1, ce qui donne 255.

> **RÉPONSE À RETENIR**
>
> L'adresse de diffusion est `192.168.16.255`.

FormaV - document d'entraînement non officiel

(c)

Question 2. (c)

Sujet p. 3

> Donner le nombre maximal de machines que l'on peut connecter sur ce réseau.

Il reste 32 − 24 = 8 bits pour la partie hôte, ce qui donne $2^8 = 256$ adresses. L'adresse du réseau et l'adresse de diffusion ne peuvent pas être attribuées à des machines ; on les retranche donc.

> **RÉPONSE À RETENIR**
>
> Le nombre maximal de machines est $256 - 2 = 254$.

(a)

Question 3. (a)

Sujet p. 3, 4

> On donne les tables de routage des routeurs R1, R2, R3 et R4 obtenues par le protocole RIP. Donner les tables de routage des routeurs R5 et R6.

Les liaisons entre routeurs visibles sur la figure 1 sont : R6–R4 et R6–R1 ; R4–R3 et R4–R5 ; R1–R5 ; R3–R5 et R3–R2 ; R5–R2. En RIP, la distance est le nombre de liaisons traversées. Une liaison directe donne donc une distance de 1. Lorsque plusieurs routes ont le même nombre minimal de sauts, chacune peut être un prochain saut valide.

Table de routage de R5		
Destination	**Passe par**	**Distance**
R1	R1	1
R2	R2	1
R3	R3	1
R4	R4	1
R6	R1 ou R4	2

Depuis R5, R6 est atteint en deux sauts soit par R1, soit par R4. Depuis R6, R1 et R4 sont voisins directs ; R3 est atteint en deux sauts par R4. Pour atteindre R2, on peut passer en trois sauts par R1–R5, par R4–R5 ou par R4–R3.

FormaV - document d'entraînement non officiel

Table de routage de R6		
Destination	**Passe par**	**Distance**
R1	R1	1
R2	R1 ou R4	3
R3	R4	2
R4	R4	1
R5	R1 ou R4	2

RÉPONSE À RETENIR

Les tables de routage de R5 et de R6 sont indiquées ci-dessus. Les choix multiples correspondent à des routes de même longueur minimale.

(b)

Question 3. (b)

Sujet p. 3, 4

> On souhaite transmettre un paquet de données depuis le poste PC Admin 01 vers le poste PC CLG 01. Déterminer le parcours emprunté par ce paquet en utilisant les tables de routage données précédemment.

PC Admin 01 est relié au switch Admin, puis au routeur R6. La table de R6 indique que, pour rejoindre R3, le paquet passe par R4. La table de R4 indique ensuite que R3 est directement accessible. Sur la figure 1, le switch CLG est relié à R3 et dessert PC CLG 01.

RÉPONSE À RETENIR

Le parcours est **PC Admin 01 → switch Admin → R6 → R4 → R3 → switch CLG → PC CLG 01**. Le parcours entre routeurs compte deux sauts.

(c)

Question 3. (c)

Sujet p. 3, 4

> À la suite d'une panne, le routeur R4 est déconnecté. Déterminer une nouvelle route pouvant être empruntée par les données depuis le secteur Administration vers le Collège en effectuant le moins de sauts possibles.

On retire R4 ainsi que les liaisons qui lui sont associées. Depuis R6, il reste la liaison vers R1 ; depuis R1, on rejoint R5, puis R3. Le chemin R6–R1–R5–R3 comporte trois sauts. Il est minimal : sans R4, il faut passer par R1, puis R5, pour atteindre R3.

RÉPONSE À RETENIR

Une route comportant le moins de sauts est **Administration → R6 → R1 → R5 → R3 → Collège**, soit trois sauts entre routeurs.

(a)

Question 4. (a)

Sujet p. 4

> Le coût d'une liaison est défini par la relation coût = $10^8/d$, où d représente le débit en bit/s. Recopier et compléter le tableau : Ethernet (coût 10), Fast-Ethernet (débit 10^8) et Fibre (débit 10^9).

On applique la relation `coût = 10⁸ / débit`. Pour Ethernet, le débit vaut `10⁸ / 10 = 10⁷` bit/s. Pour Fast-Ethernet, le coût vaut `10⁸ / 10⁸ = 1`. Pour la fibre, il vaut `10⁸ / 10⁹ = 0,1`.

RÉPONSE À RETENIR

Liaison	Débit	Coût
Ethernet	10^7 bit/s	10
Fast-Ethernet	10^8 bit/s	1
Fibre	10^9 bit/s	0,1

(b)

Question 4. (b)

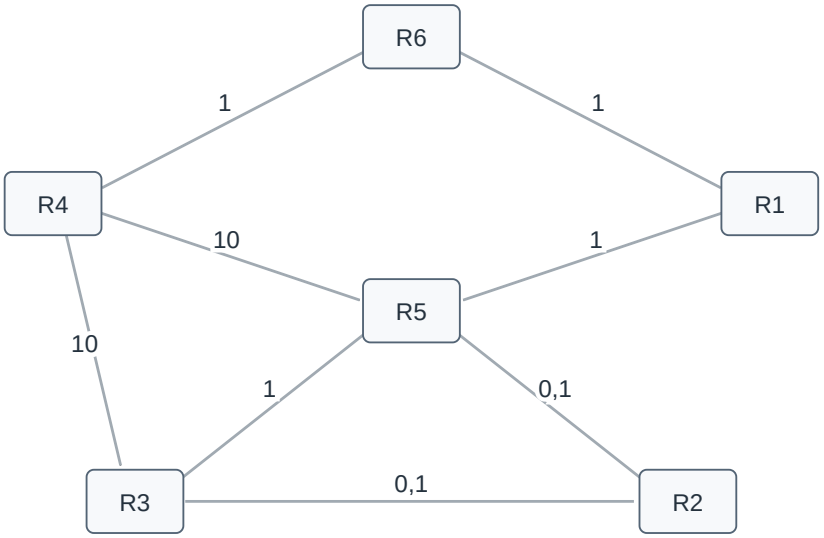
Sujet p. 4

> Recopier le schéma du réseau et indiquer le coût de chacune des liaisons connues.

D'après le tableau de la question 4(a), une liaison Ethernet a un coût de 10, une liaison Fast-Ethernet un coût de 1 et une liaison fibre un coût de 0,1. Le coût de la liaison R6–R4 se déduit de la question 4(c) : il vaut 1. La figure 2 imprime « R4 » sur le nœud à droite de R6, alors que la figure 1 et les tables RIP désignent ce routeur comme R1. Pour représenter le réseau cohérent avec l'ensemble du sujet, le schéma ci-dessous nomme ce nœud R1.

FormaV - document d'entraînement non officiel

Réseau des routeurs et coûts OSPF



Les coûts correspondent aux types de liaison inscrits dans la figure 2 ; R6–R4 vaut 1, d'après la question 4(c). Le nœud imprimé R4 à droite de R6 dans cette figure est représenté comme R1, conformément à la figure 1 et aux tables de routage.

> **RÉPONSE À RETENIR**
>
> Les coûts des liaisons sont indiqués sur le schéma ; celui de R6–R4 est déduit de la question 4(c).

(c)

Question 4. (c)

Sujet p. 4

> On sait que le parcours R6 – R4 – R5 – R2 a un coût de 11,1. Déterminer le type de liaison entre R6 et R4.

Dans le parcours R6–R4–R5–R2, la liaison R4–R5 est Ethernet, donc son coût est 10, et la liaison R5–R2 est en fibre, donc son coût est 0,1. Si le coût de R6–R4 vaut x, alors `x + 10 + 0,1 = 11,1`, d'où `x = 1`. D'après le tableau de la question 4(a), ce coût correspond à Fast-Ethernet.

> **RÉPONSE À RETENIR**
>
> La liaison R6–R4 est de type **Fast-Ethernet**.

(d)

Question 4. (d)

Sujet p. 3, 4

> On souhaite acheminer un paquet de données depuis le secteur Administration vers le réseau du Lycée Général en utilisant le protocole OSPF. Déterminer la route empruntée par un paquet de données pour aller du poste PC Admin 01 vers le poste PC LG 01.

PC Admin 01 est rattaché à R6 et PC LG 01 à R2. Depuis R6, le chemin par R1 puis R5 jusqu'à R2 coûte `1 + 1 + 0,1 = 2,1`. Le chemin R6–R4–R5–R2 coûte `1 + 10 + 0,1 = 11,1`. L'autre chemin par R1, R5, R3 et R2 coûte `1 + 1 + 1 + 0,1 = 3,1`. Les autres chemins passant par R4 empruntent au moins une liaison Ethernet de coût 10 ; ils sont donc plus coûteux que 2,1. Le trajet par R1 et R5 est ainsi le chemin de coût minimal.

> **RÉPONSE À RETENIR**
>
> La route est **PC Admin 01 → switch Admin → R6 → R1 → R5 → R2 → switch LG → PC LG 01**. Son coût entre routeurs est **2,1**.

FormaV - document d'entraînement non officiel

Exercice 3

4 points

Bases de données et SQL

(a)

Question 1. (a)

Sujet p. 5

> Écrire le résultat de la requête : SELECT id_vehicule FROM vehicule WHERE type = 'citadine';

La requête conserve les lignes de `vehicule` dont `type` vaut `'citadine'`, puis affiche leur `id_vehicule`. Dans l'extrait, ces lignes correspondent aux véhicules 1, 2 et 4.

RÉPONSE À RETENIR

id_vehicule
1
2
4

(b)

Question 1. (b)

Sujet p. 5

> Écrire une requête permettant d'afficher les immatriculations des véhicules diesel.

On sélectionne l'attribut `immatriculation` et on filtre les véhicules dont l'attribut `carburant` vaut `'diesel'`. L'extrait contient trois véhicules diesel.

RÉPONSE À RETENIR

```
SELECT immatriculation
FROM vehicule
WHERE carburant = 'diesel';
```

Résultat sur l'extrait fourni :

immatriculation
AB - 135 - YZ
CL - 142 - CE
JL - 526 - LM

FormaV - document d'entraînement non officiel

(c)

Question 1. (c)

Sujet p. 5

> Écrire le résultat de la requête : SELECT immatriculation, modele FROM vehicule ORDER BY marque;

La requête affiche les deux attributs demandés et trie les lignes selon `marque`. Avec un ordre alphabétique usuel, les marques apparaissent dans l'ordre Citroën, Peugeot, Renault, Tesla. Comme aucun critère de tri supplémentaire n'est donné pour départager deux véhicules de même marque, leur ordre relatif n'est pas garanti.

RÉPONSE À RETENIR

immatriculation	modele
ML - 128 - VM	C3
CL - 142 - CE	C5
AB - 135 - YZ	208
JL - 526 - LM	508
EC - 246 - TP	Zoe
LC - 231 - MG	Model X

L'ordre des deux Citroën entre elles et celui des deux Peugeot entre eux peuvent varier.

(d)

Question 1. (d)

Sujet p. 5

> Écrire une requête permettant de calculer le nombre de véhicules dans la table vehicule.

`COUNT(*)` compte toutes les lignes de la relation `vehicule`. L'extrait comporte six lignes.

RÉPONSE À RETENIR

```
SELECT COUNT(*)
FROM vehicule;
```

Résultat sur l'extrait :

COUNT(*)
6

(e)

Question 1. (e)

Sujet p. 5

> Écrire une requête permettant d'afficher les noms et prénoms des utilisateurs par ordre alphabétique du nom.

On sélectionne `nom` et `prenom`, puis `ORDER BY nom` trie les lignes par ordre alphabétique du nom.

RÉPONSE À RETENIR

```
SELECT nom, prenom
FROM utilisateur
ORDER BY nom;
```

Résultat sur l'extrait fourni :

nom	prenom
DUBOIS	Louise
DUPONT	Paul
DURAND	Kevin
MARTIN	Jeanne
MOREAU	Violette
PETIT	Pierre

(a)

Question 2. (a)

Sujet p. 5, 6

> Citer, en justifiant, les clés étrangères de la table location.

Une clé étrangère référence la clé primaire d'une autre relation. Dans `location`, `id_utilisateur` référence `utilisateur.id_utilisateur` et `id_vehicule` référence `vehicule.id_vehicule`. Ces deux attributs sont donc les clés étrangères de `location`.

RÉPONSE À RETENIR

Les clés étrangères sont `id_utilisateur` et `id_vehicule`.

(b)

Question 2. (b)

Sujet p. 5, 6

> Écrire un schéma relationnel des relations vehicule, utilisateur et location.

On indique les clés primaires par **PK** et les clés étrangères par **FK**. Les références des clés étrangères sont précisées après une flèche.

RÉPONSE À RETENIR

`vehicule`(**id_vehicule** PK, immatriculation, marque, modele, type, carburant)

`utilisateur`(**id_utilisateur** PK, nom, prenom, permis, adresse, ville)

`location`(**id_location** PK, id_utilisateur FK → `utilisateur.id_utilisateur`, id_vehicule FK → `vehicule.id_vehicule`, date_debut, date_fin)

(c)

Question 2. (c)

Sujet p. 5, 6

> Expliquer pourquoi la requête INSERT INTO location VALUES (1, 132, 4, '2022-05-10', '2022-05-12') produit une erreur.

La première valeur de la ligne à insérer est `id_location = 1`. Or une location d'identifiant 1 existe déjà dans l'extrait. Comme `id_location` est la clé primaire, ses valeurs doivent être uniques : l'insertion viole donc la contrainte de clé primaire. Les identifiants utilisateur 132 et véhicule 4 existent bien dans les extraits ; le doublon de clé primaire suffit à expliquer l'erreur.

RÉPONSE À RETENIR

L'insertion échoue parce que l'identifiant de location 1 est déjà utilisé alors qu'une clé primaire doit être unique.

(a)

Question 3. (a)

Sujet p. 5, 6

> Une erreur s'est glissée dans la relation utilisateur : Louise DUBOIS habite au numéro 50 de la rue de la Liberté, et non au 52. Écrire une requête permettant de modifier cette information.

Dans l'extrait, Louise DUBOIS a pour identifiant 133. La condition sur cet identifiant cible sa ligne afin de ne modifier que son adresse.

RÉPONSE À RETENIR

```
UPDATE utilisateur
SET adresse = '50 rue de la Liberté'
WHERE id_utilisateur = 133;
```

(b)

Question 3. (b)

Sujet p. 5, 6

> Un nouveau client souhaite louer une voiture. Ses informations sont : LEFEBVRE Gabriel, numéro de permis 124689, adresse 30 ruelle des champs, ville Amiens. Écrire une requête permettant d'ajouter ce nouvel utilisateur. On lui affectera un id_utilisateur de 137.

On insère les six valeurs dans l'ordre des attributs de `utilisateur` : identifiant, nom, prénom, permis, adresse et ville. La nature du type de l'attribut `permis` n'est pas précisée ; ci-dessous, le numéro est donné comme une valeur numérique, conformément à sa présentation dans l'extrait.

RÉPONSE À RETENIR

```
INSERT INTO utilisateur
VALUES (137, 'LEFEBVRE', 'Gabriel', 124689, '30 ruelle des champs', 'Amiens');
```

(c)

Question 3. (c)

Sujet p. 5, 6

> Pierre PETIT souhaite louer le véhicule de marque Tesla entre le 21 juin 2022 et le 23 juin 2022. Écrire une requête permettant d'ajouter cette information à la table location. On lui donnera un id_location égal à 4.

Dans les données fournies, Pierre PETIT a l'identifiant 132 et le véhicule de marque Tesla a l'identifiant 3. La nouvelle ligne de `location` reçoit l'identifiant 4 et les dates indiquées, au format `AAAA-MM-JJ`.

RÉPONSE À RETENIR

```
INSERT INTO location
VALUES (4, 132, 3, '2022-06-21', '2022-06-23');
```

Question 4.

Sujet p. 5, 6

> Écrire une requête permettant de lister les modèles et les immatriculations des véhicules dont les locations ont débuté le 21 juin 2022, ainsi que les noms et prénoms des utilisateurs correspondants.

La relation `location` donne la date de début et les identifiants du véhicule et de l'utilisateur. On la joint à `vehicule` avec `id_vehicule` et à `utilisateur` avec `id_utilisateur`. Le filtre retient les locations commencées le 21 juin 2022, puis la sélection affiche les deux informations demandées sur le véhicule et le nom et le prénom de l'utilisateur.

RÉPONSE À RETENIR

```
SELECT vehicule.modele, vehicule.immatriculation,
       utilisateur.nom, utilisateur.prenom
FROM location
JOIN vehicule ON location.id_vehicule = vehicule.id_vehicule
JOIN utilisateur ON location.id_utilisateur = utilisateur.id_utilisateur
WHERE location.date_debut = '2022-06-21';
```

Dans l'extrait, les locations commencées à cette date sont les locations 2 et 3. Le résultat est :

modele	immatriculation	nom	prenom
Zoe	EC - 246 - TP	MARTIN	Jeanne
C5	CL - 142 - CE	DURAND	Kevin

Sujet : https://aeif.fr/wp-content/uploads/2025/07/Mayotte_mars_2023_2.pdf
© FormaV EI • Proposition de correction, distincte d'un corrigé officiel.

Copyright © 2027 FormaV®. Tous droits réservés.

FormaV® est une marque déposée auprès de l'INPI sous le numéro 5227897.

Ce document a été élaboré par FormaV® avec le plus grand soin afin d'accompagner chaque apprenant vers la réussite de ses examens. Son contenu (textes, graphiques, méthodologies, tableaux, exercices, concepts, mises en forme) constitue une œuvre protégée par le droit d'auteur.

Toute copie, partage, reproduction, diffusion ou mise à disposition, même partielle, gratuite ou payante, est strictement interdite sans accord préalable et écrit de FormaV®, conformément aux articles L.111-1 et suivants du Code de la propriété intellectuelle. Dans une logique anti-plagiat, FormaV® se réserve le droit de vérifier toute utilisation illicite, y compris sur les plateformes en ligne ou sites tiers.

En utilisant ce document, vous vous engagez à respecter ces règles et à préserver l'intégrité du travail fourni. La consultation de ce document est strictement personnelle.

Merci de respecter le travail accompli afin de permettre la création continue de ressources pédagogiques fiables et accessibles.