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.

Numérique et sciences informatiques

Session 2023 • La Réunion • Jour 2

Code sujet **23-NSIJ2LR1** 3 heures 30

Proposition de corrigé pédagogique • FormaV

Se repérer dans le corrigé

Exercice 1 — Code Morse : arbres binaires de recherche, programmation orientée objet et récursivité
8 réponses • sujet p. 2–4

Exercice 2 — Réseaux et protocoles de routage : adressage IPv4, RIP et OSPF
11 réponses • sujet p. 4–7

Exercice 3 — Bases de données relationnelles et langage SQL : gestion de véhicules et de locations
12 réponses • sujet p. 8–10

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

Code Morse : arbres binaires de recherche, programmation orientée objet et récursivité

Question 1.

Sujet p. 2

> À l'aide de la figure de l'arbre binaire, déterminer le code morse du message « NSI », en laissant un espace entre le code de chaque lettre.

Dans l'arbre, une branche gauche correspond à « o » et une branche droite à « - ». Le chemin vers N est droite puis gauche, celui vers S est gauche-gauche-gauche, et celui vers I est gauche-gauche.

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

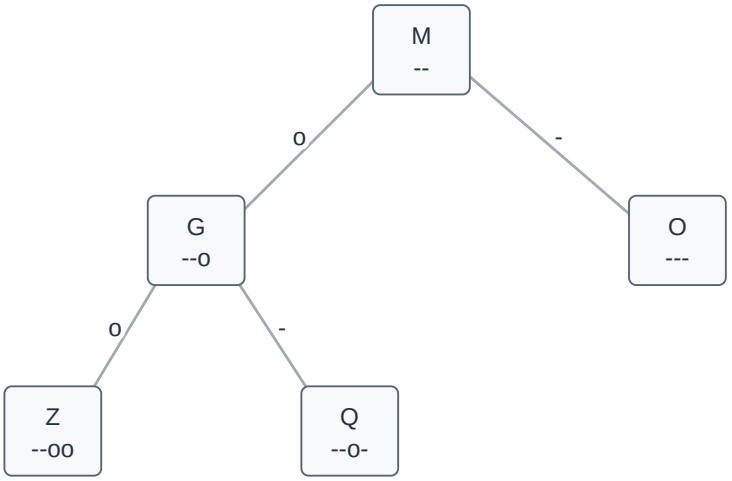
Question 2.

Sujet p. 2, 3

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

Le code de M est « -- ». À partir de ce nœud, une branche gauche ajoute « o » et mène à G (« --o »), tandis qu'une branche droite ajoute « - » et mène à O (« --- »). Depuis G, la branche gauche mène à Z (« --oo ») et la branche droite à Q (« --o- »).

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



M est la racine de ce sous-arbre ; le préfixe « -- » a déjà été parcouru depuis la racine de l'arbre complet.

> **RÉPONSE À RETENIR**
>
> Le sous-arbre demandé est représenté ci-dessous. Les étiquettes des branches indiquent le symbole ajouté au code du nœud parent.

FormaV - document d'entraînement non officiel

Question 3.

Sujet p. 3

> Écrire les instructions à placer en lignes 14 et 15 afin de créer les nœuds pour les lettres K et S.

K se trouve à droite de N, lui-même à gauche de T : le chemin depuis la racine est donc droite-gauche-droite. S se trouve à gauche de I, lui-même à gauche de E : le chemin est gauche-gauche-gauche.

RÉPONSE À RETENIR

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

a

Question 4.a

Sujet p. 3

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

Si le nœud est absent, l'arbre considéré est vide et le caractère ne peut pas s'y trouver : la fonction renvoie alors `False`. Sinon, elle compare la valeur du nœud à `car`, puis cherche dans le sous-arbre droit ou, si nécessaire, dans le sous-arbre gauche.

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. 3

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

Une fonction est récursive si elle s'appelle elle-même. Ici, dans le cas général, `est_present` s'appelle sur le sous-arbre droit et, si le résultat est faux, sur le sous-arbre gauche.

RÉPONSE À RETENIR

Oui, `est_present` est récursive. Elle possède un cas d'arrêt : lorsque `n` vaut `None`, elle renvoie `False`.

c

Question 4.c

Sujet p. 3

> Déterminer quel type de parcours utilise la fonction est_present.

La fonction examine d'abord le nœud courant, puis explore le sous-arbre droit avant le sous-arbre gauche. Elle descend dans un sous-arbre avant de passer à l'autre : c'est un parcours en profondeur, dans l'ordre préfixe nœud-droite-gauche. L'opérateur `or` étant évalué avec court-circuit, la recherche dans le sous-arbre gauche n'a lieu que si le caractère n'a pas été trouvé à droite.

> **RÉPONSE À RETENIR**
>
> Il s'agit d'un **parcours en profondeur préfixe**, dans l'ordre nœud, droite, gauche.

a

Question 5.a

Sujet p. 3, 4

> Recopier et compléter les blancs des lignes 10, 11 et 14 de la fonction code_morse.

Lorsque le nœud courant contient le caractère recherché, il n'y a plus de branche à ajouter : le code renvoie la chaîne vide. Si le caractère se trouve dans le sous-arbre droit, on ajoute « - ». Sinon, comme l'énoncé garantit que le caractère est présent dans l'arbre, il se trouve dans le sous-arbre gauche et on ajoute « o ».

> **RÉPONSE À RETENIR**
>
> ```
> def code_morse(n, car):
>     if n.valeur == car:
>         return ""
>     elif est_present(n.droite, car):
>         return "-" + code_morse(n.droite, car)
>     else:
>         return "o" + code_morse(n.gauche, car)
> ```
>
> Cette fonction suppose que `car` est présent dans l'arbre, comme l'indique l'énoncé.

b

Question 5.b

Sujet p. 3, 4

> 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|.

La boucle traite les lettres du message dans l'ordre. Pour chacune, elle calcule le code avec `code_morse`, puis ajoute le séparateur vertical demandé. L'exemple fourni se terminant également par un séparateur, le code en ajoute un après chaque lettre.

RÉPONSE À RETENIR

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

Pour l'exemple de l'énoncé, le résultat est `o--o|-o--|-|oooo|---|-o|`. La fonction suppose que les caractères du message sont présents dans l'arbre.

Exercice 2

4 points

Réseaux et protocoles de routage : adressage IPv4, RIP et OSPF

Question 1.

Sujet p. 5

> Le poste PC LG 03 de la salle informatique du Lycée général a pour adresse IPv4 192.168.162.4. Recopier et compléter cette adresse IP : 11000000.10101000.xxxxxxxx.xxxxxxxx.

On convertit chaque octet séparément en binaire sur 8 bits. Les deux premiers octets sont déjà indiqués : 192 = 11000000 et 168 = 10101000. Ensuite, 162 = 128 + 32 + 2, donc son écriture sur 8 bits est 10100010 ; enfin, 4 s'écrit 00000100 sur 8 bits.

> **RÉPONSE À RETENIR**
>
> **11000000.10101000.10100010.00000100**

a

Question 2.a

Sujet p. 5

> 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 suffixe /24 indique que les 24 premiers bits sont ceux de la partie réseau. Le masque est donc formé de trois octets à 255, suivis d'un octet à 0 : 255.255.255.0. Pour obtenir l'adresse du réseau, on conserve les trois premiers octets de l'adresse donnée et on met à 0 les bits de la partie hôte, ici le dernier octet.

> **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. 5

> Pour le réseau du secteur « Administration », dont le poste PC Admin 02 a pour adresse IPv4 192.168.16.12/24, donner l'adresse de diffusion, appelée aussi broadcast.

Dans un réseau /24, le dernier octet correspond à la partie hôte. Pour l'adresse de diffusion, tous les bits de cette partie sont mis à 1 : le dernier octet vaut donc 255.

> **RÉPONSE À RETENIR**
>
> L'adresse de diffusion (broadcast) est **192.168.16.255**.

FormaV - document d'entraînement non officiel

c

Question 2.c

Sujet p. 5

> Pour ce réseau, donner le nombre maximal de machines que l'on peut connecter.

Un réseau /24 laisse 8 bits pour la partie hôte, soit $2^8 = 256$ adresses. Deux sont réservées : celle dont la partie hôte est entièrement à 0 (adresse du réseau) et celle dont elle est entièrement à 1 (adresse de diffusion). On peut donc connecter $256 - 2 = 254$ machines.

> **RÉPONSE À RETENIR**
>
> Le nombre maximal de machines connectables est **254**.

a

Question 3.a

Sujet p. 4, 5

> À partir des tables de routage des routeurs R1, R2, R3 et R4 obtenues avec le protocole RIP, donner les tables de routage des routeurs R5 et R6.

On lit les liaisons entre routeurs sur le schéma : R5 est directement relié à R1, R2, R3 et R4 ; R6 est directement relié à R1 et R4. Une liaison directe compte pour un saut. Pour les autres destinations, on retient une route de nombre minimal de sauts. Lorsque plusieurs routes minimales existent, le prochain routeur indiqué peut varier.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

Destination depuis R5	Passe par	Distance
R1	R1	1
R2	R2	1
R3	R3	1
R4	R4	1
R6	R1	2

Destination depuis R6	Passe par	Distance
R1	R1	1
R2	R1	3
R3	R4	2
R4	R4	1
R5	R1	2

Les choix indiqués sont des routes minimales. Par exemple, de R5 à R6, on peut passer par R1 ou R4 en 2 sauts. De R6 à R2, les routes R6–R1–R5–R2, R6–R4–R5–R2 et R6–R4–R3–R2 comptent toutes 3 sauts. De R6 à R5, le premier saut peut également être R1 ou R4.

b

Question 3.b

Sujet p. 4, 5, 6

> 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.

Le poste PC Admin 01 est raccordé au routeur R6 et PC CLG 01 au routeur R3, comme le montre le schéma. Dans la table de R6, la destination R3 est atteinte en passant par R4, en 2 sauts. Depuis R4, la table indique ensuite R3 en 1 saut.

RÉPONSE À RETENIR

Le parcours est **PC Admin 01 → R6 → R4 → R3 → PC CLG 01**.

c

Question 3.c

Sujet p. 4, 6

> À la suite d'une panne, le routeur R4 est déconnecté. Déterminer une nouvelle route pouvant être empruntée par

FormaV - document d'entraînement non officiel

> les données depuis le secteur « Administration » vers le Collège en effectuant le moins de sauts possible.

Une fois R4 déconnecté, on cherche un chemin minimal de R6 (routeur du secteur Administration) à R3 (routeur du Collège) qui ne passe pas par R4. Le chemin R6–R1–R5–R3 emprunte 3 liaisons. Il est minimal : R6 ne peut rejoindre R3 en 1 saut, et ses voisins restants sont R1 ; depuis R1, il faut encore passer par R5 avant d'atteindre R3.

RÉPONSE À RETENIR

Une route minimale possible est **R6 → R1 → R5 → R3**, soit **3 sauts**.

a

Question 4.a

Sujet p. 6

> Le routeur R4 étant réparé et reconnecté au réseau, on applique le protocole de routage OSPF. Le coût d'une liaison est défini par coût = 10^8/d, où d représente le débit en bit/s. Recopier et compléter le tableau : Ethernet (coût 10, débit à compléter), Fast-Ethernet (débit 10^8, coût à compléter) et Fibre (débit 10^9, coût à compléter).

La relation donnée est $\text{coût} = 10^8/d$, où d est le débit en bit/s. Pour Ethernet, le coût est 10 : $d = 10^8/10 = 10^7$ bit/s. Pour Fast-Ethernet, $d = 10^8$ bit/s, donc le coût vaut 1. Pour la fibre, $d = 10^9$ bit/s, donc le coût vaut 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

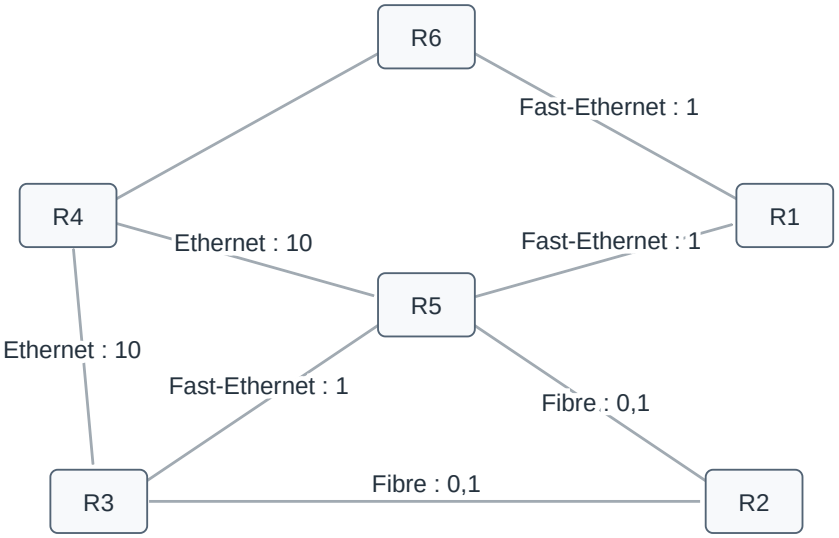
Sujet p. 6, 7

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

On reporte sur chaque liaison le coût correspondant à son type, obtenu à la question 4.a : Ethernet vaut 10, Fast-Ethernet vaut 1 et fibre vaut 0,1. Le schéma indique le type de toutes les liaisons sauf R6–R4 ; son type, et donc son coût, est déterminé à la question 4.c. Les liaisons et leurs types ci-dessous correspondent au schéma du sujet.

FormaV - document d'entraînement non officiel

Réseau des routeurs et coûts des liaisons connues



Les arêtes reproduisent les liaisons du schéma du sujet ; la liaison R6–R4 est présente mais son coût n'est pas encore indiqué.

> **RÉPONSE À RETENIR**
>
> Les coûts des liaisons connues sont indiqués sur le schéma ci-dessous. La liaison R6–R4 est représentée sans coût, car son type n'est pas encore connu à cette question.

c

Question 4.c

Sujet p. 6, 7

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

Le coût du parcours R6–R4–R5–R2 est la somme des coûts de ses trois liaisons. R4–R5 est de type Ethernet, donc son coût est 10 ; R5–R2 est en fibre, donc son coût est 0,1. Si x est le coût de R6–R4, alors $x + 10 + 0,1 = 11,1$, d'où $x = 1$. D'après le tableau de la question 4.a, un coût de 1 correspond à Fast-Ethernet.

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

d

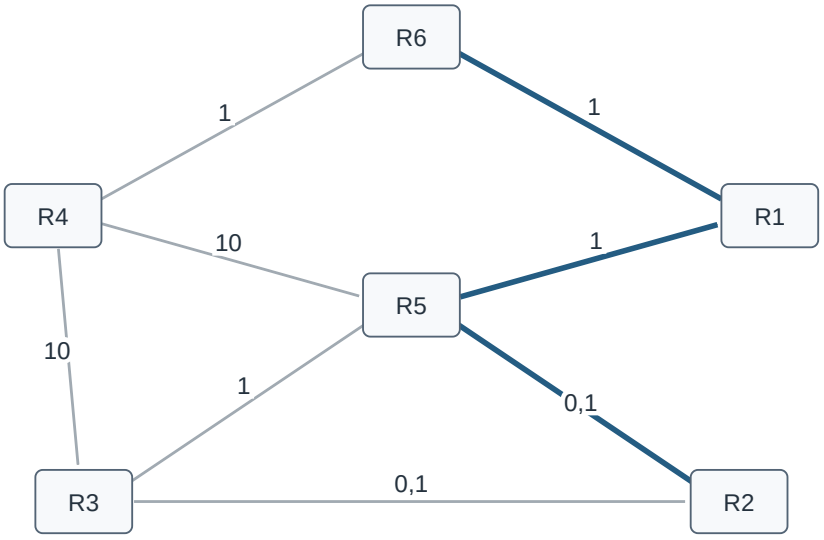
Question 4.d

Sujet p. 4, 6, 7

> En utilisant le protocole OSPF, déterminer la route empruntée par un paquet de données pour aller du poste PC Admin 01 vers le poste ayant pour adresse PC LG 01.

PC Admin 01 rejoint le réseau des routeurs par R6 et PC LG 01 par R2. Depuis R6, le chemin R6–R1–R5–R2 a pour coût $1 + 1 + 0,1 = 2,1$. Les chemins passant par R4 sont plus coûteux : R6–R4–R5–R2 coûte $1 + 10 + 0,1 = 11,1$, et R6–R4–R3–R2 coûte également $1 + 10 + 0,1 = 11,1$. Le chemin passant par R1 et R5 est donc le chemin de coût minimal.

Route OSPF de R6 vers R2



Le chemin OSPF mis en évidence est R6–R1–R5–R2 ; son coût est 1 + 1 + 0,1 = 2,1.

RÉPONSE À RETENIR

La route empruntée est **PC Admin 01 → R6 → R1 → R5 → R2 → PC LG 01**. Son coût entre routeurs est **2,1**.

Exercice 3

4 points

Bases de données relationnelles et langage SQL : gestion de véhicules et de locations

a

Question 1.a

Sujet p. 8, 9

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

La clause `WHERE type = 'citadine'` conserve les lignes dont le type est « citadine », puis `SELECT id_vehicule` n'affiche que leur identifiant. Dans l'extrait, les véhicules concernés ont les identifiants 1, 2 et 4. Comme la requête ne contient pas de `ORDER BY`, l'ordre d'affichage n'est pas garanti.

RÉPONSE À RETENIR

id_vehicule
1
2
4

b

Question 1.b

Sujet p. 8, 9

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

On filtre les lignes avec `WHERE carburant = 'diesel'`, puis on sélectionne uniquement l'attribut `immatriculation`.

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

RÉPONSE À RETENIR

Résultat sur les véhicules visibles dans l'extrait. L'ordre n'est pas garanti faute de `ORDER BY`.

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

c

Question 1.c

Sujet p. 8, 9

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

La requête affiche les deux colonnes demandées et trie les lignes selon la marque. Dans l'extrait, les marques se rangent dans l'ordre Citroën, Peugeot, Renault, Tesla. Pour deux véhicules de même marque, aucun ordre n'est imposé entre eux puisqu'aucun critère supplémentaire ne départage les ex æquo.

RÉPONSE À RETENIR

Un résultat possible sur l'extrait est le suivant ; l'ordre des lignes de même marque n'est pas garanti.

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

d

Question 1.d

Sujet p. 8, 9

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

`COUNT(*)` compte toutes les lignes de la table, sans filtrage. La requête permettant de compter les véhicules est :

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

L'extrait montre six véhicules, mais les points de suspension indiquent que la table complète contient éventuellement d'autres lignes.

RÉPONSE À RETENIR

La requête renvoie le nombre total de véhicules de la table complète. Le compte des seules lignes visibles dans l'extrait est 6 ; le nombre total de lignes de la table complète ne peut pas être déterminé à partir de l'extrait.

e

Question 1.e

Sujet p. 8, 9

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

On sélectionne les attributs `nom` et `prenom`, puis on trie les résultats par ordre alphabétique croissant du nom avec `ORDER BY nom`.

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

RÉPONSE À RETENIR

Résultat pour les utilisateurs visibles dans l'extrait :

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

a

Question 2.a

Sujet p. 8, 9

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

Les clés étrangères de `location` sont `id_utilisateur` et `id_vehicule`. `id_utilisateur` référence la clé primaire `utilisateur.id_utilisateur` : une location est associée à un utilisateur. `id_vehicule` référence la clé primaire `vehicule.id_vehicule` : une location concerne un véhicule.

RÉPONSE À RETENIR

`location.id_utilisateur` est une clé étrangère vers `utilisateur(id_utilisateur)`, et `location.id_vehicule` est une clé étrangère vers `vehicule(id_vehicule)`.

b

Question 2.b

Sujet p. 8, 9

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

Voici le schéma relationnel, avec les clés primaires et les clés étrangères. Les attributs et les clés primaires sont ceux indiqués dans le sujet ; les deux références de `location` relient cette relation aux deux autres.

RÉPONSE À RETENIR

```
vehicule(
    id_vehicule [clé primaire],
    immatriculation,
    marque,
    modele,
    type,
    carburant
)

utilisateur(
    id_utilisateur [clé primaire],
    nom,
    prenom,
    permis,
    adresse,
    ville
)

location(
    id_location [clé primaire],
    id_utilisateur [clé étrangère vers utilisateur.id_utilisateur],
    id_vehicule [clé étrangère vers vehicule.id_vehicule],
    date_debut,
    date_fin
)
```

c

Question 2.c

Sujet p. 8, 9

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

La première valeur insérée est `1` : elle attribue donc la valeur 1 à `id_location`. Or l'extrait montre déjà une location d'identifiant 1. Comme `id_location` est la clé primaire, cette valeur doit être unique : l'insertion viole la contrainte de clé primaire. Les identifiants 132 pour l'utilisateur et 4 pour le véhicule figurent bien dans les extraits fournis.

RÉPONSE À RETENIR

L'erreur vient du doublon de la clé primaire `id_location = 1`, déjà attribuée à une location existante.

a

Question 3.a

Sujet p. 8, 9

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

Il faut modifier l'attribut `adresse` de la ligne de Louise DUBOIS. L'identifiant 133, visible dans l'extrait, permet de cibler précisément son enregistrement. On remplace le numéro 52 par 50 en conservant le reste de l'adresse.

RÉPONSE À RETENIR

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

b

Question 3.b

Sujet p. 8, 9

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

On insère les valeurs dans l'ordre des attributs de la relation `utilisateur` donné dans le sujet : identifiant, nom, prénom, permis, adresse, ville. L'identifiant demandé est 137.

RÉPONSE À RETENIR

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

c

Question 3.c

Sujet p. 8, 10

> 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.

Pierre PETIT a pour identifiant 132 dans l'extrait, et le véhicule Tesla a pour identifiant 3. En respectant l'ordre des attributs de la table `location`, on insère l'identifiant de location demandé, les deux clés étrangères, puis les dates 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. 8, 9, 10

> 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.

Il faut relier les trois relations : `location` à `vehicule` par `id_vehicule`, et `location` à `utilisateur` par `id_utilisateur`. On filtre ensuite les locations dont la date de début est le 21 juin 2022.

```
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';
```

RÉPONSE À RETENIR

Sur les seules locations visibles dans l'extrait fourni, les identifiants 2 et 3 ont commencé le 21 juin 2022. Les véhicules et utilisateurs associés sont :

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

Sujet : https://www.sujetdebac.fr/annales-pdf/2023/spe-numerique-informatique-2023-la-reunion-2-sujet-officiel.pdf

© FormaV EI • Proposition de correction, distincte d'un corrigé officiel.

FormaV - document d'entraînement non 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.