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 2022 • Nouvelle-Calédonie

Code sujet **22-NSIJ1NC1** 3 heures 30

Proposition de corrigé pédagogique • FormaV

Se repérer dans le corrigé

Exercice 1 — Schéma relationnel et requêtes SQL sur la base IMDb
7 réponses • sujet p. 3–4

Exercice 2 — Architecture des ordinateurs et protocoles réseau
11 réponses • sujet p. 5–8

Exercice 3 — Traitement de données de villes à partir d’un fichier CSV en Python
11 réponses • sujet p. 9–10

Exercice 4 — Arbres binaires, parcours et profils des membres d’un club
13 réponses • sujet p. 11–13

Exercice 5 — Programmation orientée objet et gestion de comptes d’une monnaie locale
8 réponses • sujet p. 14–15

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

Schéma relationnel et requêtes SQL sur la base IMDb

Partie 1 : Étude du schéma relationnel

Question 1)

Sujet p. 3

> Indiquer ce qui différencie les concepts de clé primaire et de clé étrangère.

Une clé primaire identifie de manière unique chaque ligne d'une table : ses valeurs ne se répètent pas et ne sont pas absentes. Une clé étrangère est un champ, ou un ensemble de champs, qui référence une clé primaire d'une autre table. Elle sert à relier les tables et à assurer la cohérence des références. Sauf contrainte supplémentaire, une clé étrangère peut être répétée ou absente.

> **RÉPONSE À RETENIR**
>
> La clé primaire identifie sans ambiguïté une ligne de sa table ; la clé étrangère référence une ligne d'une autre table. La clé étrangère peut être répétée, contrairement à la clé primaire.

Question 2)

Sujet p. 3

> Indiquer l'utilité des clés étrangères dans la relation Joue_Role de ce schéma relationnel.

Dans `Joue_Role`, `Idacteur` et `Idfilm` sont des clés étrangères : elles relient chaque rôle à l'acteur qui l'interprète et au film concerné. Les contraintes de clé étrangère vérifient que les lignes référencées existent dans les tables `Acteur` et `Film`.

> **RÉPONSE À RETENIR**
>
> Les clés étrangères de `Joue_Role` relient un rôle à un acteur et à un film existants, et préservent la cohérence des données.

Partie 2 : Accès aux données de la base IMDb

Question 1)

Sujet p. 3

> Écrire une requête SQL permettant d'obtenir la liste des titres des films et de leur année de sortie.

Les deux informations demandées figurent dans la table `Film`. La requête sélectionne donc les champs `titre` et `annee`.

> **RÉPONSE À RETENIR**
>
> ```
> SELECT titre, annee
> FROM Film;
> ```


FormaV - document d'entraînement non officiel

Question 2)

Sujet p. 3

> En 2021, Fathia YOUSSOUF a obtenu le César du meilleur espoir féminin pour le film « Mignonnes ». Écrire une requête permettant de créer l'enregistrement dans la table Acteur de cette actrice selon le modèle fourni : Idacteur 1250, acteur_nom YOUSSOUF, acteur_prenom Fathia.

On insère une ligne dans `Acteur` en indiquant les champs et leurs valeurs correspondantes. L'identifiant est un entier et les valeurs textuelles sont placées entre apostrophes.

RÉPONSE À RETENIR

```
INSERT INTO Acteur (Idacteur, acteur_nom, acteur_prenom)
VALUES (1250, 'YOUSSOUF', 'Fathia');
```

Question 3)

Sujet p. 4

> On cherche à supprimer un acteur de la base. Après l'exécution de la requête DELETE FROM Acteur WHERE acteur_nom='DUPONT' AND acteur_prenom='David', le message indique qu'une contrainte de clé étrangère dans Joue_Role référence Acteur.Idacteur. Indiquer ce qui a pu provoquer cette erreur.

Le message précise que la clé étrangère de `Joue_Role` référence `Acteur.Idacteur` et bloque la suppression. Cela signifie qu'au moins une ligne de `Joue_Role` référence effectivement l'identifiant de David DUPONT. Supprimer cet acteur laisserait une référence sans ligne correspondante dans `Acteur`, ce qui violerait l'intégrité référentielle.

RÉPONSE À RETENIR

Au moins une ligne de `Joue_Role` fait référence à l'identifiant de David DUPONT ; la contrainte de clé étrangère empêche donc sa suppression tant que ces références ne sont pas traitées.

Question 4)

Sujet p. 4

> Écrire une requête permettant d'obtenir le nom et le prénom du réalisateur du film ayant pour titre « Léon ».

Dans `Film`, `Idrealisateur` est une clé étrangère qui référence `Realisateur.Idrealisateur`. On joint les deux tables sur ces identifiants, puis on filtre sur le titre « Léon » et on sélectionne le nom et le prénom du réalisateur.

RÉPONSE À RETENIR

```
SELECT Realisateur.realisateur_nom, Realisateur.realisateur_prenom
FROM Film
JOIN Realisateur
  ON Film.Idrealisateur = Realisateur.Idrealisateur
WHERE Film.titre = 'Léon';
```


FormaV - document d'entraînement non officiel

Question 5)

> Écrire une requête permettant d'obtenir le nom des acteurs du film ayant pour titre « Intouchables ».

La table `Joue_Role` relie les films et les acteurs. On la joint à `Film` avec `Idfilm`, puis à `Acteur` avec `Idacteur`. Le filtre conserve les rôles associés au film « Intouchables ». La requête renvoie le nom demandé pour chaque ligne de rôle correspondante.

RÉPONSE À RETENIR

```
SELECT Acteur.acteur_nom
FROM Film
JOIN Joue_Role
  ON Film.Idfilm = Joue_Role.Idfilm
JOIN Acteur
  ON Joue_Role.Idacteur = Acteur.Idacteur
WHERE Film.titre = 'Intouchables';
```

Exercice 2

4 points

Architecture des ordinateurs et protocoles réseau

Partie 1 : Choix d'un serveur

Question 1)

Sujet p. 5

> Indiquer la quantité de mémoire vive du Raspberry Pi.

La quantité de mémoire vive est indiquée dans les caractéristiques du Raspberry Pi 4B, à la ligne « RAM ».

> **RÉPONSE À RETENIR**
>
> Le Raspberry Pi possède **8 Gio de mémoire vive (RAM)**.

Question 2)

Sujet p. 5

> Indiquer la fonction principale du composant SSD (Solid State Drive).

Un SSD est un support de stockage persistant : il conserve les données même lorsque l'ordinateur est éteint. Il peut notamment contenir le système d'exploitation, les logiciels et les fichiers.

> **RÉPONSE À RETENIR**
>
> La fonction principale du SSD est le **stockage des données** de l'ordinateur.

Question 3)

Sujet p. 5

> Indiquer deux avantages d'un ordinateur monocarte (SoC) par rapport à un mini-ordinateur.

Les caractéristiques indiquent une alimentation de 15 W pour le Raspberry Pi, contre 65 W pour le mini-PC : le Raspberry Pi nécessite donc une alimentation de puissance nominale plus faible. Un ordinateur monocarte est également compact et peu encombrant.

> **RÉPONSE À RETENIR**
>
> Deux avantages du Raspberry Pi par rapport au mini-PC sont son **faible encombrement** et la **puissance d'alimentation plus faible indiquée** (15 W contre 65 W).

Partie 2 : Paramétrage du serveur web

Question 1)

Sujet p. 6

> Avec le masque 255.255.255.0 et les adresses IP indiquées sur la figure 3, indiquer l'adresse IP du « Réseau services ».

Avec le masque 255.255.255.0, les trois premiers octets identifient le réseau et le dernier identifie une machine. Les interfaces indiquées dans le « Réseau services » commencent par 192.168.254 ; l'adresse de réseau s'obtient en mettant à 0 la partie machine.

RÉPONSE À RETENIR

L'adresse IP du « Réseau services » est **192.168.254.0**.

Question 2)

Sujet p. 6

Indiquer le nombre d'adresses IP encore disponibles qui peuvent être attribuées au serveur web.

Avec le masque 255.255.255.0, le réseau comprend 256 adresses, de 192.168.254.0 à 192.168.254.255. L'adresse .0 identifie le réseau et l'adresse .255 est l'adresse de diffusion : il reste donc 254 adresses attribuables à des machines. Les deux adresses déjà utilisées par les interfaces des routeurs sont .1 et .2, donc il en reste 254 − 2 = 252.

RÉPONSE À RETENIR

Il reste **252 adresses IP** attribuables à des machines sur le « Réseau services » ; le serveur web pourra recevoir l'une d'elles.

Question 3)

Sujet p. 6

Indiquer l'adresse IP de la passerelle donnant au serveur web l'accès à Internet.

La passerelle du serveur web est l'interface du routeur reliée au réseau services, qui permet ensuite de rejoindre Internet. Sur la figure 3, cette interface appartient au routeur 2 et porte l'adresse 192.168.254.2.

RÉPONSE À RETENIR

L'adresse IP de la passerelle du serveur web est **192.168.254.2**.

Partie 3 : Analyse d'une trame réseau

Question 1)

Sujet p. 7

En analysant les échanges de trame présentés en figure 4, indiquer l'adresse physique / MAC (Media Access Control) du serveur DNS.

La trame 1 demande l'adresse MAC correspondant à 192.168.0.253. Dans la réponse ARP de la trame 2, cette adresse IP est associée à 8A:FD:54:49:D0:CC. La figure 3 identifie 192.168.0.253 comme l'adresse du serveur DNS.

RÉPONSE À RETENIR

L'adresse MAC du serveur DNS est **8A:FD:54:49:D0:CC**.

Question 2)

Sujet p. 7

En utilisant les détails de la trame n° 4 (figure 5), indiquer le protocole utilisé pour cette trame au niveau de la couche transport.

Dans le détail de la trame 4, la rubrique « Transport » indique le protocole UDP. Le champ « Protocol: 17 » de la rubrique Internet est également cohérent avec UDP.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

Le protocole utilisé à la couche transport est **UDP**.

Question 3)

Sujet p. 7

> À partir du détail de la trame n° 4 (figure 5), indiquer l'adresse IP du serveur web.

Dans le détail de la trame 4, la réponse DNS associe le nom www.monsite.sme à l'adresse IPv4 192.168.254.201. Cette adresse est celle du serveur web hébergeant le site.

RÉPONSE À RETENIR

L'adresse IP du serveur web est **192.168.254.201**.

Partie 4 : Protocole de routage

Question 1)

Sujet p. 8

> Donner la table de routage du routeur R4.

La figure 6 montre que R4 est directement relié à R2 et à R6. Ces deux destinations sont donc à un saut. R1 et R3 sont accessibles en passant par R2, et R5 en passant par R6 : ces destinations sont à deux sauts. Comme dans la table de R2 donnée dans le sujet, on ne fait pas figurer le routeur lui-même comme destination.

RÉPONSE À RETENIR

Destination	Routeur suivant	Distance
R1	R2	2
R2	R2	1
R3	R2	2
R5	R6	2
R6	R6	1

Question 2)

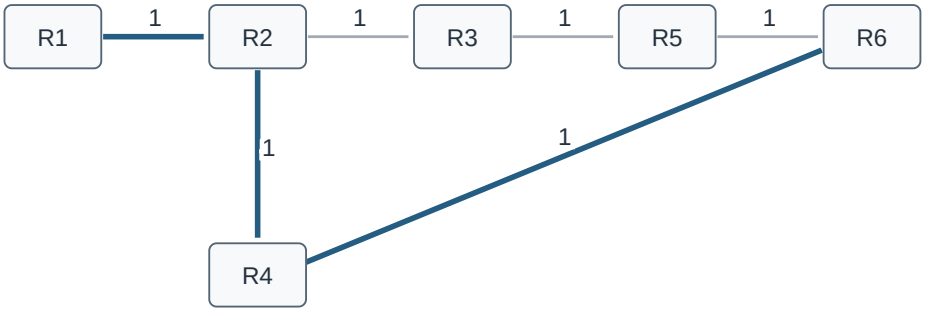
Sujet p. 8

> Le routeur R1 doit transmettre un message au routeur R6 en effectuant un nombre minimal de sauts. Indiquer le trajet parcouru.

La figure 6 montre deux itinéraires possibles vers R6 : R1 → R2 → R4 → R6 comporte 3 liaisons, tandis que R1 → R2 → R3 → R5 → R6 en comporte 4. Le premier trajet comporte donc le nombre minimal de sauts.

FormaV - document d'entraînement non officiel

Réseau de routeurs et trajet minimal de R1 à R6



Chaque liaison représente un saut. Le trajet minimal mis en évidence est R1 → R2 → R4 → R6.

> **RÉPONSE À RETENIR**
>
> Le trajet minimal est **R1 → R2 → R4 → R6**, soit **3 sauts**.

Exercice 3

4 points

Traitement de données de villes à partir d'un fichier CSV en Python

Partie 1

Question 1.a) – Réponse A

Sujet p. 9

> En appliquant le Programme 1, indiquer ce que signifie « utf-8 » à la ligne 8. Proposition A : type de codage de caractères informatiques.

Dans l'appel à `open`, l'argument `encoding` précise comment interpréter les caractères du fichier lors de sa lecture. `utf-8` est un codage de caractères.

RÉPONSE À RETENIR

La proposition A est vraie : `utf-8` désigne un type de codage de caractères informatiques.

Question 1.a) – Réponse B

Sujet p. 9

> En appliquant le Programme 1, indiquer ce que signifie « utf-8 » à la ligne 8. Proposition B : nom du dossier contenant le fichier.

L'argument `encoding` concerne le codage du texte ; il ne précise pas le dossier contenant le fichier.

RÉPONSE À RETENIR

La proposition B est fausse.

Question 1.a) – Réponse C

Sujet p. 9

> En appliquant le Programme 1, indiquer ce que signifie « utf-8 » à la ligne 8. Proposition C : nom du fichier.

Le nom du fichier est donné par l'argument `filename`, ici `villes56.csv`. L'argument `encoding` indique le codage des caractères.

RÉPONSE À RETENIR

La proposition C est fausse.

Question 1.a) – Réponse D

Sujet p. 9

> En appliquant le Programme 1, indiquer ce que signifie « utf-8 » à la ligne 8. Proposition D : taille du fichier.

L'argument `encoding` décrit le codage des caractères, et non la taille du fichier.

RÉPONSE À RETENIR

La proposition D est fausse.

FormaV - document d'entraînement non officiel

Question 1.b)

Sujet p. 9

> Indiquer ce que signifie « mode = 'r' » à la ligne 8 du Programme 1.

Dans l'appel à `open`, le mode `'r'` indique que le fichier est ouvert en lecture.

> **RÉPONSE À RETENIR**
>
> `mode='r'` signifie que le fichier est ouvert pour être lu.

Question 2.a)

Sujet p. 10

> Après exécution du Programme 1, indiquer le type de la variable listeVilles.

`readCSV` construit une liste à partir des lignes fournies par `csv.DictReader`. Chaque ligne est convertie en dictionnaire, puis la liste obtenue est affectée à `listeVilles`.

> **RÉPONSE À RETENIR**
>
> `listeVilles` est une liste Python, plus précisément une liste de dictionnaires.

Question 2.b)

Sujet p. 10

> Indiquer ce que renvoie l'instruction listeVilles[1], en précisant son type.

Les indices d'une liste Python commencent à 0 : l'indice 1 correspond donc à la deuxième ville, Vannes. Dans le résultat du programme 1, les valeurs lues dans le CSV sont des chaînes de caractères.

> **RÉPONSE À RETENIR**
>
> `listeVilles[1]` renvoie un dictionnaire de type `dict` :
>
> ```
> {'dep': '56', 'nom': 'Vannes', 'nb_hab_1999': '51759', 'nb_hab_2012': '53000',
> 'densite': '1625', 'superficie': '32.3', 'alt_min': '0', 'alt_max': '56'}
> ```

Question 2.c)

Sujet p. 9, 10

> Écrire une instruction permettant d'obtenir le nombre d'habitants à Pontivy en 2012.

On sélectionne le dictionnaire dont la clé `'nom'` a pour valeur `'Pontivy'`, puis on récupère la valeur associée à `'nb_hab_2012'`. Dans la liste produite par le programme 1, cette valeur est une chaîne de caractères.

> **RÉPONSE À RETENIR**
>
> ```
> [d['nb_hab_2012'] for d in listeVilles if d['nom'] == 'Pontivy'][0]
> ```
>
> Cette instruction renvoie `'13700'`.

FormaV - document d'entraînement non officiel

Partie 2

Question 1)

Sujet p. 9, 10

> On admet qu'après un traitement, les données figurant dans les colonnes 2 à 7 du tableau listeVilles sont des entiers ou des flottants et que listeVilles résulte de l'exécution du Programme 1. Indiquer ce que renvoie l'instruction [(d['nom'], d['alt_max']) for d in listeVilles if d['alt_max']>100] appliquée à listeVilles.

Après le traitement indiqué, les valeurs de `alt_max` sont numériques. La condition `d['alt_max'] > 100` retient Pontivy, dont l'altitude maximale vaut 192, et Saint-Avé, dont elle vaut 136. Chaque paire construite entre parenthèses est un tuple.

RÉPONSE À RETENIR

L'instruction renvoie :

```
[('Pontivy', 192), ('Saint-Avé', 136)]
```

Question 2)

Sujet p. 9, 10

> Écrire en langage Python une fonction variations qui prend en paramètre listeV, du même type que la variable listeVilles, et renvoie la différence totale entre le nombre d'habitants en 2012 et le nombre d'habitants en 1999, cumulée pour l'ensemble des villes.

On initialise un total à zéro, puis on ajoute, pour chaque ville, le nombre d'habitants en 2012 moins celui de 1999. Les champs de population étant numériques après le traitement mentionné dans l'énoncé, cette soustraction est possible. La fonction effectue un seul parcours de la liste.

```
def variations(listeV):
    total = 0
    for d in listeV:
        total += d['nb_hab_2012'] - d['nb_hab_1999']
    return total
```

RÉPONSE À RETENIR

Pour les neuf villes, les différences sont :

`-1124 + 1241 + 599 - 101 + 1190 + 199 + 1601 + 945 + 1802 = 6352`.

Appliquée aux données du tableau, la fonction renvoie donc **6352**.

Question 3)

Sujet p. 9, 10

> Écrire en langage Python une fonction surmax qui prend en paramètre listeV, du même type que la variable listeVilles, et renvoie la ville ayant la superficie la plus grande. On admet que deux villes n'ont pas la même superficie.

On prend le premier dictionnaire comme meilleure candidate, puis on parcourt les autres villes. Chaque fois qu'une superficie est supérieure à celle de la meilleure candidate, on remplace celle-ci. La liste fournie contient des villes et les superficies ont été converties en nombres après le traitement indiqué. La fonction renvoie le dictionnaire de la ville retenue.

FormaV - document d'entraînement non officiel

```
def surmax(listeV):
    meilleure = listeV[0]
    for d in listeV[1:]:
        if d['superficie'] > meilleure['superficie']:
            meilleure = d
    return meilleure
```

RÉPONSE À RETENIR

La plus grande superficie est **52.29** : elle correspond à Guidel. La fonction renvoie donc le dictionnaire de Guidel, dont le champ `'nom'` vaut `'Guidel'`.

Exercice 4

4 points

Arbres binaires, parcours et profils des membres d'un club

Question 1.a)

Sujet p. 11, 12

> On définit la hauteur d'un arbre non vide comme la longueur, en nombre d'arêtes, du plus long chemin allant d'une feuille à la racine. Un arbre vide a une hauteur égale à −1 et un arbre qui n'a qu'un seul nœud a une hauteur de 0. Indiquer la hauteur de l'arbre P.

La hauteur est le nombre d'arêtes du plus long chemin entre la racine et une feuille. Dans l'arbre P, un chemin le plus long est AnneB → Sophia → AstridM → KevinH (ou Nico) : il comporte 3 arêtes.

RÉPONSE À RETENIR

La hauteur de l'arbre P est **3**.

Question 1.b)

Sujet p. 12

> Recopier et compléter le programme implémentant la fonction récursive hauteur, qui prend en argument un arbre binaire A et renvoie la hauteur de cet arbre. On pourra utiliser la fonction max renvoyant la valeur maximale entre deux valeurs. Compléter les tests et expressions manquants des lignes 2, 6 et 7.

Pour un arbre vide, la hauteur vaut −1 : c'est le cas de base. Pour un arbre non vide, on calcule les hauteurs des sous-arbres gauche et droit, puis on ajoute 1 à la plus grande. Une feuille a ainsi une hauteur de 1 + max(−1, −1) = 0. À chaque appel récursif, on considère un sous-arbre plus petit ; on finit donc par atteindre des arbres vides.

Sur P, les feuilles FredB, Marc, KevinH et Nico ont une hauteur de 0. Pedro et Malik2 ont une hauteur de 1 ; AstridM et Sophia ont une hauteur de 2 ; AnneB a une hauteur de 3.

```
def hauteur(A):
    if estvide(A):
        return -1
    else:
        g = hauteur(gauche(A))
        d = hauteur(droit(A))
        return 1 + max(g, d)
```

RÉPONSE À RETENIR

Les compléments sont : ligne 2, `estvide(A)` ; ligne 6, `d = hauteur(droit(A))` ; ligne 7, `1 + max(g, d)`.

Question 1.c)

Sujet p. 12

> Indiquer le type de la valeur renvoyée par la fonction hauteur.

La fonction renvoie −1 pour un arbre vide. Sinon, elle renvoie 1 plus le maximum de deux hauteurs, qui sont des entiers. La valeur renvoyée est donc toujours un entier.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

Le type de la valeur renvoyée par `hauteur` est **int** (entier).

Question 2.a)

Sujet p. 11, 12

La fonction membres(A,L) ajoute l'étiquette de chaque nœud à la liste L, en traitant la racine puis le sous-arbre gauche et le sous-arbre droit. Écrire le contenu de L_membres après l'exécution de L_membres = [] puis membres(P, L_membres), où P désigne l'arbre donné.

La fonction ajoute d'abord la racine à la liste, puis parcourt le sous-arbre gauche et enfin le sous-arbre droit. En suivant cet ordre sur P, on visite AnneB, Pedro, FredB, Sophia, Malik2, Marc, AstridM, KevinH et Nico.

RÉPONSE À RETENIR

```
L_membres = ['AnneB', 'Pedro', 'FredB', 'Sophia', 'Malik2', 'Marc', 'AstridM',
'KevinH', 'Nico']
```

Question 2.b)

Sujet p. 12

Indiquer le nom du type de parcours d'arbre binaire réalisé par la fonction membres.

La fonction visite la racine avant ses sous-arbres, puis le sous-arbre gauche avant le sous-arbre droit. C'est un parcours préfixe, aussi appelé parcours en profondeur préordre.

RÉPONSE À RETENIR

Il s'agit d'un **parcours préfixe** (racine, gauche, droite).

Question 3.a)

Sujet p. 11, 12

Indiquer les étiquettes des feuilles de l'arbre P.

Une feuille est un nœud qui ne possède aucun enfant. Sur P, les nœuds sans enfant sont FredB, Marc, KevinH et Nico.

RÉPONSE À RETENIR

Les feuilles de P sont **FredB, Marc, KevinH et Nico**.

Question 3.b) – Réponse A

Sujet p. 11, 12

À partir des propositions, indiquer le profil des membres dont les pseudonymes sont les étiquettes des feuilles de l'arbre P. Proposition A : membre or.

Une feuille n'a aucun enfant dans l'arbre de parrainage : son membre n'a parrainé personne. Il ne peut donc pas être or, profil qui correspond à deux personnes parrainées.

RÉPONSE À RETENIR

La proposition A (« membre or ») est **fausse**.

FormaV - document d'entraînement non officiel

Question 3.b) – Réponse B

Sujet p. 11, 12

> À partir des propositions, indiquer le profil des membres dont les pseudonymes sont les étiquettes des feuilles de l'arbre P. Proposition B : membre argent.

Une feuille n'a aucun enfant dans l'arbre de parrainage : son membre n'a parrainé personne. Il ne peut donc pas être argent, profil qui correspond à une personne parrainée.

> **RÉPONSE À RETENIR**
>
> La proposition B (« membre argent ») est **fausse**.

Question 3.b) – Réponse C

Sujet p. 11, 12

> À partir des propositions, indiquer le profil des membres dont les pseudonymes sont les étiquettes des feuilles de l'arbre P. Proposition C : membre bronze.

Les feuilles n'ont aucun enfant dans l'arbre ; les membres correspondants n'ont donc parrainé personne. C'est précisément la définition du profil bronze.

> **RÉPONSE À RETENIR**
>
> La proposition C (« membre bronze ») est **vraie**.

Question 3.b) – Réponse D

Sujet p. 11, 12

> À partir des propositions, indiquer le profil des membres dont les pseudonymes sont les étiquettes des feuilles de l'arbre P. Proposition D : on ne peut pas savoir.

Pour les feuilles, le nombre d'enfants est nul. Leur profil est donc déterminé : elles correspondent à des membres bronze.

> **RÉPONSE À RETENIR**
>
> La proposition D (« on ne peut pas savoir ») est **fausse** : les membres concernés sont bronze.

Question 3.c)

Sujet p. 11, 13

> Écrire la fonction profil prenant en argument un arbre binaire non vide A et renvoyant le profil du membre dont le pseudonyme est l'étiquette de la racine de A, sous la forme d'une chaîne de caractères : « membre_or », « membre_argent » ou « membre_bronze ». Par exemple, profil(P) doit renvoyer « membre_or », correspondant au profil d'AnneB, racine de P.

On compte les sous-arbres non vides de la racine : deux enfants correspondent au profil or, un au profil argent et aucun au profil bronze. L'argument étant un arbre non vide, la fonction peut accéder à ses sous-arbres. La racine de P, AnneB, a deux enfants ; `profil(P)` renvoie donc `'membre_or'`.

FormaV - document d'entraînement non officiel

```
def profil(A):
    nb_enfants = 0
    if not estvide(gauche(A)):
        nb_enfants += 1
    if not estvide(droit(A)):
        nb_enfants += 1

    if nb_enfants == 2:
        return 'membre_or'
    elif nb_enfants == 1:
        return 'membre_argent'
    else:
        return 'membre_bronze'
```

RÉPONSE À RETENIR

La fonction renvoie le profil de la racine selon le nombre de ses enfants. En particulier, `profil(P)` renvoie `'membre_or'`.

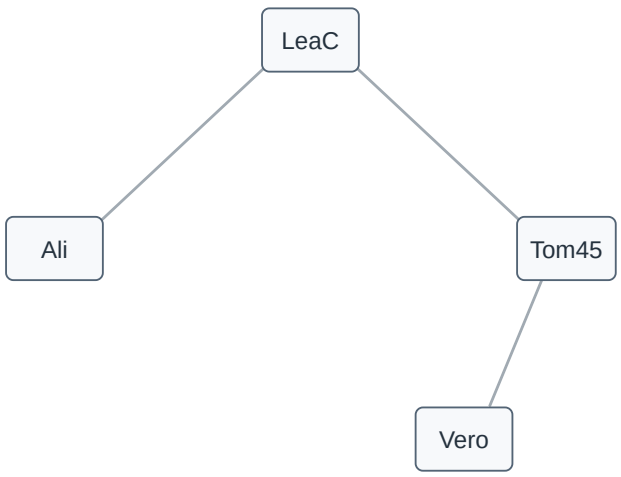
Question 4)

Sujet p. 13

La fonction membres_profiles produit un tableau dont chaque élément est un tuple contenant le pseudonyme d'un membre et son profil. Après l'exécution de L2 = [] puis membres_profiles(arbre2, L2), on obtient L2 = [('LeaC', 'membre_or'), ('Ali', 'membre_bronze'), ('Tom45', 'membre_argent'), ('Vero', 'membre_bronze')]. Dessiner un arbre possible pouvant correspondre à arbre2.

La liste est produite par un parcours préfixe : racine, sous-arbre gauche, sous-arbre droit. On place donc LeaC à la racine, Ali à gauche et Tom45 à droite. Ali, de profil bronze, est une feuille. Tom45, de profil argent, a un seul enfant, Vero, qui est une feuille de profil bronze. LeaC a deux enfants et est donc or. Cet arbre respecte l'ordre et les profils indiqués.

Un arbre possible pour arbre2



Le parcours préfixe donne LeaC, Ali, Tom45, Vero ; leurs profils sont respectivement or, bronze, argent et bronze.

RÉPONSE À RETENIR

Un arbre possible est représenté ci-dessous. Vero est ici placé comme enfant gauche de Tom45.

Question 5)

Sujet p. 11, 13

> Chaque année, les membres reversent une cotisation selon leur profil : membre or, 20 € ; membre argent, 30 € ; membre bronze, 40 €. Écrire une fonction cotisation en langage Python ou en pseudo-code qui prend en argument un arbre binaire modélisant les parrainages comme explicité au début de l'exercice et renvoie le montant total des cotisations reçues par le club. On pourra utiliser la fonction membres_profiles de la question précédente.

On utilise `membres_profiles` pour obtenir chaque membre et son profil, puis on additionne la cotisation correspondante. Chaque nœud est traité une fois ; le temps de calcul est proportionnel au nombre de nœuds de l'arbre.

Sur P, les profils dans l'ordre du parcours sont : AnneB or, Pedro argent, FredB bronze, Sophia or, Malik2 argent, Marc bronze, AstridM or, KevinH bronze et Nico bronze. Le total vaut 3 × 20 + 2 × 30 + 4 × 40 = 60 + 60 + 160 = 280 €.

```
def cotisation(A):
    liste = []
    membres_profiles(A, liste)

    total = 0
    for pseudonyme, profil_membre in liste:
        if profil_membre == 'membre_or':
            total += 20
        elif profil_membre == 'membre_argent':
            total += 30
        else:
            total += 40

    return total
```

RÉPONSE À RETENIR

`cotisation(P)` renvoie **280**, soit **280 €**.

Exercice 5

4 points

Programmation orientée objet et gestion de comptes d'une monnaie locale

Partie 1 : Création de la classe compte

Question 1)

Sujet p. 14

> Écrire sur la copie la méthode crediter.

La méthode `crediter` ajoute le montant reçu au solde actuel. Elle met donc à jour l'attribut `self.solde` en lui ajoutant `montant`.

RÉPONSE À RETENIR

```
def crediter(self, montant):
    self.solde = self.solde + montant
```

Après l'appel, le solde augmente de `montant`.

Question 2)

Sujet p. 14

> Écrire sur la copie la méthode debiter.

Débiter un compte consiste à retirer le montant demandé du solde existant. La méthode soustrait donc `montant` à `self.solde`. Aucune vérification de provision n'est demandée pour cette méthode.

RÉPONSE À RETENIR

```
def debiter(self, montant):
    self.solde = self.solde - montant
```

Question 3)

Sujet p. 14

> Écrire sur la copie la méthode est_positif.

La méthode renvoie `True` lorsque le solde est positif ou nul, et `False` lorsqu'il est négatif. La comparaison `self.solde >= 0` effectue exactement ce test.

RÉPONSE À RETENIR

```
def est_positif(self):
    return self.solde >= 0
```

Partie 2 : Utilisation de la classe compte

Question 1)

Sujet p. 15

> Écrire la ligne de code en langage Python permettant de créer le compte cpt_0123456A, dont le numéro est « 0123456A », le titulaire est « MARTIN Dominique » et l'adresse est « 12 rue des sports ».

Le constructeur de `Compte` reçoit, dans cet ordre, le numéro, le nom de l'adhérent et l'adresse. Il initialise automatiquement le solde à 0 sou.

RÉPONSE À RETENIR

```
cpt_0123456A = Compte("0123456A", "MARTIN Dominique", "12 rue des sports")
```

Question 2)

Sujet p. 15

> Écrire la ligne de code en langage Python permettant de déposer 200 € sur le compte de monsieur Martin.

On appelle la méthode `crediter` sur le compte de Monsieur Martin avec `200` comme montant. Le sou étant à parité avec l'euro, le dépôt de 200 € correspond à 200 sous.

RÉPONSE À RETENIR

```
cpt_0123456A.crediter(200)
```

Le solde du compte passe de 0 à 200 sous.

Question 3)

Sujet p. 15

> Monsieur Martin souhaite transférer 50 sous de son compte « cpt_0123456A » vers un autre compte « cpt_2468794E ». Écrire la méthode transferer(self, autre_compte, montant), qui transfère le montant passé en paramètre vers un autre compte. On suppose que le compte courant est suffisamment approvisionné. Utiliser les autres méthodes de la classe Compte sans modifier directement les attributs.

Le transfert retire d'abord le montant du compte courant, puis crédite l'autre compte de ce même montant. On utilise les méthodes `debiter` et `crediter`, sans modifier directement les attributs. L'énoncé précise que le compte courant est suffisamment approvisionné.

RÉPONSE À RETENIR

```
def transferer(self, autre_compte, montant):
    self.debiter(montant)
    autre_compte.crediter(montant)
```

Par exemple, `cpt_0123456A.transferer(cpt_2468794E, 50)` débite 50 sous du compte de Monsieur Martin et crédite de 50 sous l'autre compte.

Partie 3 : Gestion des comptes

Question 1)

Sujet p. 15

> Écrire la fonction rechercher_debiteurs qui prend en argument liste_comptes et renvoie une liste de dictionnaires pour les adhérents dont le solde est négatif. La première clé de chaque dictionnaire est le nom de l'adhérent et la seconde clé est le solde de son compte en négatif. Exemple : [{'Nom': 'DUPONT Thomas', 'solde': -60}, {'Nom': 'CARNEIRO Sarah', 'solde': -75}].

On parcourt chaque compte de `liste_comptes`. La méthode `est_positif` permet de repérer les soldes positifs ou nuls ; on conserve donc les comptes pour lesquels elle renvoie `False`, c'est-à-dire ceux dont le solde est négatif. Pour chaque compte retenu, on ajoute un dictionnaire avec les clés dans l'ordre demandé : `'Nom'`, puis `'solde'`. Le solde est enregistré avec sa valeur négative.

Le parcours examine chaque compte une fois : le temps d'exécution est proportionnel au nombre de comptes. Si aucun compte n'est débiteur, la fonction renvoie une liste vide.

RÉPONSE À RETENIR

```
def rechercher_debiteurs(liste_comptes):
    liste_debiteurs = []
    for compte in liste_comptes:
        if not compte.est_positif():
            liste_debiteurs.append({
                'Nom': compte.adherent,
                'solde': compte.solde
            })
    return liste_debiteurs
```

Pour des comptes dont les noms et soldes sont ceux de l'exemple du sujet, la liste renvoyée contient `{'Nom': 'DUPONT Thomas', 'solde': -60}` et `{'Nom': 'CARNEIRO Sarah', 'solde': -75}`.

Question 2)

Sujet p. 15

> Écrire la fonction urgent_debiteur qui prend en argument la liste de dictionnaires et renvoie le nom de l'adhérent le plus endetté. On suppose qu'aucun adhérent n'a la même dette.

Les soldes de la liste sont négatifs. Le membre le plus endetté est donc celui dont le solde est le plus petit : un solde de `-75` correspond à une dette plus importante qu'un solde de `-60`. On mémorise l'indice du plus petit solde rencontré et on le met à jour au cours du parcours. L'énoncé exclut les dettes égales, donc le membre trouvé est unique.

Si la liste est vide, aucun nom ne peut être renvoyé ; le code choisit alors de renvoyer `None`. Pour une liste non vide, il effectue un seul parcours.

RÉPONSE À RETENIR

```
def urgent_debiteur(liste_debiteurs):
    if len(liste_debiteurs) == 0:
        return None

    indice_plus_endette = 0
    for i in range(1, len(liste_debiteurs)):
        if liste_debiteurs[i]['solde'] < liste_debiteurs[indice_plus_endette]
['solde']:
            indice_plus_endette = i

    return liste_debiteurs[indice_plus_endette]['Nom']
```

Avec l'exemple du sujet, la comparaison des soldes `-60` et `-75` retient `-75`, car `-75 < -60`. La fonction renvoie donc `'CARNEIRO Sarah'`.

Sujet : https://aeif.fr/wp-content/uploads/2022/12/22_NSIJ1NC1.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.