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 • Amérique du Sud

Code sujet **22-NSIJ2AS1** 3 heures 30

Proposition de corrigé pédagogique • FormaV

Se repérer dans le corrigé

Exercice 1 — Recherche des k plus proches voisins : programmation, algorithmique et complexité
5 réponses • sujet p. 2–3

Exercice 2 — Réseaux et routage
24 réponses • sujet p. 4–5

Exercice 3 — Bases de données et requêtes SQL
11 réponses • sujet p. 6–7

Exercice 4 — Algorithme de Douglas-Peucker, programmation Python et récursivité
6 réponses • sujet p. 8–10

Exercice 5 — Arbres binaires, programmation orientée objet et récursivité
6 réponses • sujet p. 11–12

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

Recherche des k plus proches voisins : programmation, algorithmique et complexité

Question 1.

Sujet p. 2

> On suppose dans cette question que k = 1. La fonction plus_proche_voisin(t, cible), qui prend en argument le tableau t et l'objet cible, contient un bloc manquant. Écrire sur votre copie uniquement le bloc d'instructions manquant pour que la fonction renvoie l'indice d'un plus proche voisin de cible.

Le tableau est non vide et la fonction a déjà mémorisé la distance du premier élément dans `dmin`, ainsi que son indice dans `idx_ppv`. Pour chaque indice suivant, on calcule la distance à la cible et on remplace le meilleur candidat si cette distance est plus petite.

```
    d = distance(t[idx], cible)
    if d < dmin:
        dmin = d
        idx_ppv = idx
```

En cas d'égalité, garder l'indice déjà mémorisé convient puisque la fonction doit renvoyer l'indice d'un plus proche voisin, pas nécessairement un indice particulier parmi plusieurs ex æquo.

> **RÉPONSE À RETENIR**
>
> Ce bloc permet à la fonction de renvoyer l'indice d'un élément dont la distance à `cible` est minimale.

Question 2.

Sujet p. 2

> On considère que le coût en temps du bloc manquant est constant. Quelle est la complexité de la fonction plus_proche_voisin quand k = 1 ?

Soit `n` la taille du tableau. La fonction examine les éléments d'indice 1 à `n - 1`, soit `n - 1` éléments. Le travail effectué à chaque itération est de coût constant d'après l'hypothèse de l'énoncé. L'initialisation et le retour sont également de coût constant. Le temps d'exécution croît donc proportionnellement à `n`.

> **RÉPONSE À RETENIR**
>
> La complexité temporelle est linéaire : $\Theta(n)$, donc aussi $O(n)$.

Question 3.a.

Sujet p. 2, 3

> Dans la fonction recherche_kppv(t, k, cible), on remarque qu'il y a plusieurs appels identiques à la fonction distance(obj, cible). Comment ne faire qu'un seul appel de cette fonction ?

On calcule la distance une seule fois au début de chaque itération et on la conserve dans une variable locale `d`. On utilise ensuite cette variable pour l'insertion ou pour la comparaison avec la distance du voisin actuellement le plus éloigné. Voici la fonction avec cette modification :

FormaV - document d'entraînement non officiel

```
def recherche_kppv(t, k, cible):
    kppv = []
    n = len(t)
    for idx in range(n):
        obj = t[idx]
        d = distance(obj, cible)
        if len(kppv) < k:
            insertion(kppv, idx, d)
        else:
            i0, d0 = kppv[0]
            if d < d0:
                kppv.pop(0)
                insertion(kppv, idx, d)
    return kppv
```

Ainsi, chaque objet parcouru donne lieu à un seul appel à `distance`, y compris lorsque sa distance sert seulement à vérifier s'il faut le retenir.

> **RÉPONSE À RETENIR**
>
> Calculer `d = distance(obj, cible)` une fois par itération, puis réutiliser `d` dans les deux branches.

Question 3.b.

Sujet p. 2, 3

> Expliquer l'intérêt de maintenir la liste kppv triée.

La liste `kppv` est triée par distances décroissantes. Son premier tuple contient donc la distance la plus grande parmi celles des voisins retenus : c'est le voisin retenu le moins proche de la cible. Lorsque la liste contient déjà `k` éléments, il suffit de comparer la distance du nouvel objet à celle de ce premier tuple. Si le nouvel objet est plus proche, on retire le premier tuple puis on insère le nouveau à sa place dans l'ordre. Le tri permet également de renvoyer directement la liste dans l'ordre décroissant demandé.

> **RÉPONSE À RETENIR**
>
> Le tri permet de consulter immédiatement en `kppv[0]` le voisin retenu le moins proche, de décider s'il faut le remplacer, et de conserver la liste dans l'ordre attendu.

Question 3.c.

Sujet p. 3

> Écrire une fonction insertion(kppv, idx, d) qui insère le tuple (idx, d) dans la liste kppv préalablement triée, en préservant l'ordre décroissant selon l'élément d. On pourra éventuellement utiliser la méthode insert.

On parcourt la liste depuis le début tant que les distances déjà présentes sont supérieures ou égales à `d`. On insère alors le nouveau tuple à la première position où cet ordre décroissant est préservé. En cas d'égalité, le nouveau tuple est placé après ceux qui ont la même distance.

```
def insertion(kppv, idx, d):
    position = 0
    while position < len(kppv) and kppv[position][1] >= d:
        position += 1
    kppv.insert(position, (idx, d))
```

La boucle termine car `position` augmente à chaque itération et ne dépasse pas la longueur de la liste. Par exemple, si deux distances consécutives satisfont `d1 >= d > d2`, le tuple de distance `d` est inséré entre ces deux tuples. La recherche de la position examine au plus tous les éléments de la liste ; l'insertion dans une liste Python peut aussi décaler des éléments.

> **RÉPONSE À RETENIR**
>
> La fonction insère `(idx, d)` dans `kppv` tout en préservant l'ordre décroissant des distances.

Exercice 2

4 points

Réseaux et routage

Partie A

Question 1. — option 1

Sujet p. 4

> Sous Unix, quelle instruction en ligne de commande a pu délivrer l'information selon laquelle l'adresse IP de l'interface réseau de l'ordinateur est 192.168.1.14 avec le masque 255.255.255.0 ? Option : ifconfig.

`ifconfig` affiche la configuration des interfaces réseau, notamment les adresses IP et les masques.

> **RÉPONSE À RETENIR**
>
> **Bonne réponse : `ifconfig`.**

Question 1. — option 2

Sujet p. 4

> Sous Unix, quelle instruction en ligne de commande a pu délivrer l'information selon laquelle l'adresse IP de l'interface réseau de l'ordinateur est 192.168.1.14 avec le masque 255.255.255.0 ? Option : ping.

`ping` sert à tester la communication avec une autre machine ; il ne donne pas la configuration complète de l'interface locale.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** La commande attendue est `ifconfig`.

Question 1. — option 3

Sujet p. 4

> Sous Unix, quelle instruction en ligne de commande a pu délivrer l'information selon laquelle l'adresse IP de l'interface réseau de l'ordinateur est 192.168.1.14 avec le masque 255.255.255.0 ? Option : ps.

`ps` affiche des informations sur les processus en cours d'exécution, pas sur les interfaces réseau.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** La commande attendue est `ifconfig`.

Question 1. — option 4

Sujet p. 4

> Sous Unix, quelle instruction en ligne de commande a pu délivrer l'information selon laquelle l'adresse IP de l'interface réseau de l'ordinateur est 192.168.1.14 avec le masque 255.255.255.0 ? Option : ls.

`ls` affiche le contenu d'un répertoire, pas la configuration réseau.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** La commande attendue est `ifconfig`.

FormaV - document d'entraînement non officiel

Question 2. — option 1

Sujet p. 4

> Parmi les protocoles proposés, lequel a permis d'attribuer automatiquement l'adresse IP 192.168.1.14 ? Option : DNS.

Le DNS associe principalement les noms de domaine à des adresses IP ; il n'attribue pas automatiquement une adresse IP à l'ordinateur.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** Le protocole d'attribution automatique est DHCP.

Question 2. — option 2

Sujet p. 4

> Parmi les protocoles proposés, lequel a permis d'attribuer automatiquement l'adresse IP 192.168.1.14 ? Option : DHCP.

Le protocole DHCP permet à un appareil de recevoir automatiquement une configuration réseau, dont son adresse IP.

> **RÉPONSE À RETENIR**
>
> **Bonne réponse : DHCP.**

Question 2. — option 3

Sujet p. 4

> Parmi les protocoles proposés, lequel a permis d'attribuer automatiquement l'adresse IP 192.168.1.14 ? Option : TCP.

TCP est un protocole de transport des données ; il ne distribue pas les adresses IP aux appareils.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** Le protocole attendu est DHCP.

Question 2. — option 4

Sujet p. 4

> Parmi les protocoles proposés, lequel a permis d'attribuer automatiquement l'adresse IP 192.168.1.14 ? Option : HTTP.

HTTP est un protocole de communication utilisé notamment pour le Web ; il n'attribue pas les adresses IP.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** Le protocole attendu est DHCP.

Question 3. — option 1

Sujet p. 4

> Parmi les adresses IP proposées, laquelle est la seule possible pour un autre appareil connecté au même réseau que l'ordinateur d'adresse 192.168.1.14 avec le masque 255.255.255.0 ? Option : 192.168.0.14.

Avec le masque 255.255.255.0, les trois premiers octets identifient le réseau. L'ordinateur appartient donc au réseau 192.168.1.0/24. L'adresse 192.168.0.14 appartient à un autre réseau.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

Cette option est fausse. 192.168.0.14 n'est pas dans le réseau 192.168.1.0/24.

Question 3. — option 2

Sujet p. 4

> Parmi les adresses IP proposées, laquelle est la seule possible pour un autre appareil connecté au même réseau que l'ordinateur d'adresse 192.168.1.14 avec le masque 255.255.255.0 ? Option : 192.168.0.1.

Le masque 255.255.255.0 correspond au réseau 192.168.1.0/24 : les trois premiers octets doivent donc être 192.168.1. L'adresse proposée, 192.168.0.1, appartient au réseau 192.168.0.0/24, et non au réseau de l'ordinateur.

RÉPONSE À RETENIR

Cette option est fausse. 192.168.0.1 n'est pas dans le réseau de l'ordinateur ; la seule réponse possible parmi les choix est 192.168.1.1.

Question 3. — option 3

Sujet p. 4

> Parmi les adresses IP proposées, laquelle est la seule possible pour un autre appareil connecté au même réseau que l'ordinateur d'adresse 192.168.1.14 avec le masque 255.255.255.0 ? Option : 192.168.1.1.

Avec le masque 255.255.255.0, le réseau est 192.168.1.0/24. L'adresse 192.168.1.1 appartient à ce réseau, n'est ni l'adresse du réseau ni celle de diffusion, et est distincte de 192.168.1.14.

RÉPONSE À RETENIR

Bonne réponse : 192.168.1.1.

Question 3. — option 4

Sujet p. 4

> Parmi les adresses IP proposées, laquelle est la seule possible pour un autre appareil connecté au même réseau que l'ordinateur d'adresse 192.168.1.14 avec le masque 255.255.255.0 ? Option : 192.168.1.255.

Pour le réseau 192.168.1.0/24, 192.168.1.255 est l'adresse de diffusion. Elle ne peut pas être attribuée à un appareil comme adresse d'hôte.

RÉPONSE À RETENIR

Cette option est fausse. La seule réponse possible parmi les choix est 192.168.1.1.

Question 4. — option 1

Sujet p. 4

> À son domicile, l'élève consulte une page web qui prétend que l'adresse IP de son ordinateur est 88.168.10.210. Choisir la seule bonne réponse. Option : « C'est une fausse information car son adresse IP est 192.168.1.14. »

192.168.1.14 est l'adresse privée de l'interface de l'ordinateur sur le réseau local. Une adresse IP publique différente peut être utilisée par la box pour communiquer vers Internet.

RÉPONSE À RETENIR

Cette option est fausse. Un site peut voir l'adresse publique utilisée par la connexion, différente de l'adresse locale 192.168.1.14.

Question 4. — option 2

Sujet p. 4

> À son domicile, l'élève consulte une page web qui prétend que l'adresse IP de son ordinateur est 88.168.10.210. Choisir la seule bonne réponse. Option : « C'est sûrement faux car seul le fournisseur d'accès peut avoir connaissance de cette information. »

Le site Web auquel l'ordinateur se connecte peut connaître l'adresse IP publique à l'origine de la connexion. Cette information n'est donc pas réservée au fournisseur d'accès.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** Le site peut voir l'adresse publique de la connexion.

Question 4. — option 3

Sujet p. 4

> À son domicile, l'élève consulte une page web qui prétend que l'adresse IP de son ordinateur est 88.168.10.210. Choisir la seule bonne réponse. Option : « C'est possible et cette adresse serait celle de la box vers Internet. »

La box effectue généralement une traduction d'adresses entre le réseau local et Internet. Le site consulté peut donc voir l'adresse publique utilisée par la box pour cette connexion.

> **RÉPONSE À RETENIR**
>
> **Bonne réponse :** c'est possible ; 88.168.10.210 peut être l'adresse de la box vers Internet.

Question 4. — option 4

Sujet p. 4

> À son domicile, l'élève consulte une page web qui prétend que l'adresse IP de son ordinateur est 88.168.10.210. Choisir la seule bonne réponse. Option : « C'est possible, mais cela signifierait que l'ordinateur est infecté par un malware. »

Le fait qu'un site affiche une adresse IP publique ne signifie pas que l'ordinateur est infecté : il s'agit habituellement de l'adresse publique de la connexion.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** La situation décrite ne permet pas de conclure à une infection par un malware.

Question 5. — option 1

Sujet p. 4

> Est-il possible qu'un ordinateur connecté au réseau du lycée possède la même adresse IP que l'élève à son domicile ? Choisir la seule bonne réponse. Option : « Oui, à condition que les connexions n'aient pas lieu au même moment. »

Des adresses privées peuvent être réutilisées dans des réseaux locaux indépendants, même lorsque ces réseaux fonctionnent au même moment. Il n'est donc pas nécessaire que les connexions aient lieu à des moments différents.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** La réutilisation ne dépend pas du fait que les connexions aient lieu à des moments différents.

Question 5. — option 2

Sujet p. 4

> Est-il possible qu'un ordinateur connecté au réseau du lycée possède la même adresse IP que l'élève à son domicile ? Choisir la seule bonne réponse. Option : « Oui, car les adresses 192.168.x.x ne sont pas routées sur Internet. »

Les adresses privées, comme 192.168.x.x, ne sont pas routées sur l'Internet public. Des réseaux locaux indépendants, par exemple celui du domicile et celui du lycée, peuvent donc réutiliser la même adresse.

> **RÉPONSE À RETENIR**
>
> **Bonne réponse :** oui, car les adresses 192.168.x.x ne sont pas routées sur Internet.

Question 5. — option 3

Sujet p. 4

> Est-il possible qu'un ordinateur connecté au réseau du lycée possède la même adresse IP que l'élève à son domicile ? Choisir la seule bonne réponse. Option : « Oui, à condition d'utiliser un VPN. »

Un VPN n'est pas une condition nécessaire à la réutilisation d'une adresse privée : deux réseaux locaux distincts peuvent déjà utiliser la même adresse.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** La réutilisation est possible sans que l'utilisation d'un VPN soit nécessaire.

Question 5. — option 4

Sujet p. 4

> Est-il possible qu'un ordinateur connecté au réseau du lycée possède la même adresse IP que l'élève à son domicile ? Choisir la seule bonne réponse. Option : « Non, car deux machines sont identifiées de manière unique par leur adresse IP. »

Une adresse IP privée n'identifie pas une machine de manière unique à l'échelle mondiale. Elle peut être réutilisée dans plusieurs réseaux locaux indépendants.

> **RÉPONSE À RETENIR**
>
> **Cette option est fausse.** Deux ordinateurs sur des réseaux locaux distincts peuvent avoir la même adresse privée.

Partie B

Question 1.

Sujet p. 5

> Vérifier que le coût d'une liaison VDSL est égal à 20.

La formule du sujet est $C = \dfrac{10^9}{BP}$, avec BP en bits par seconde. Une liaison VDSL a un débit de $50\,\text{Mb/s} = 50 \times 10^6\,\text{bits/s}$. On obtient donc $C = \dfrac{10^9}{50 \times 10^6} = 20$.

> **RÉPONSE À RETENIR**
>
> Le coût d'une liaison VDSL est bien égal à **20**.

FormaV - document d'entraînement non officiel

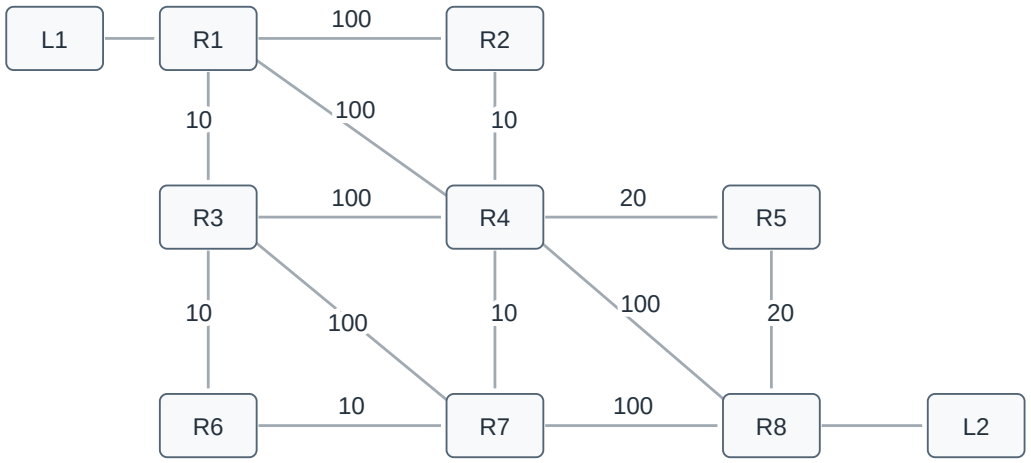
Question 2.a.

Sujet p. 5

> Recopier le graphe du réseau sur votre copie en inscrivant les coûts des liaisons.

On applique la formule $C = 10^9 / BP$. Ethernet à 10 Mb/s donne un coût de 100 ; VDSL à 50 Mb/s donne 20 ; VDSL2 à 100 Mb/s donne 10. On reporte ces coûts sur les liaisons routeur-routeur du tableau du sujet.

Réseau pondéré : coûts des liaisons



Les liaisons L1–R1 et R8–L2 relient les réseaux locaux aux routeurs ; le sujet ne leur attribue pas de coût.

RÉPONSE À RETENIR

Les coûts à inscrire sont : R1–R2 100 ; R1–R3 10 ; R1–R4 100 ; R2–R4 10 ; R3–R4 100 ; R3–R6 10 ; R3–R7 100 ; R4–R5 20 ; R4–R7 10 ; R4–R8 100 ; R5–R8 20 ; R6–R7 10 ; R7–R8 100.

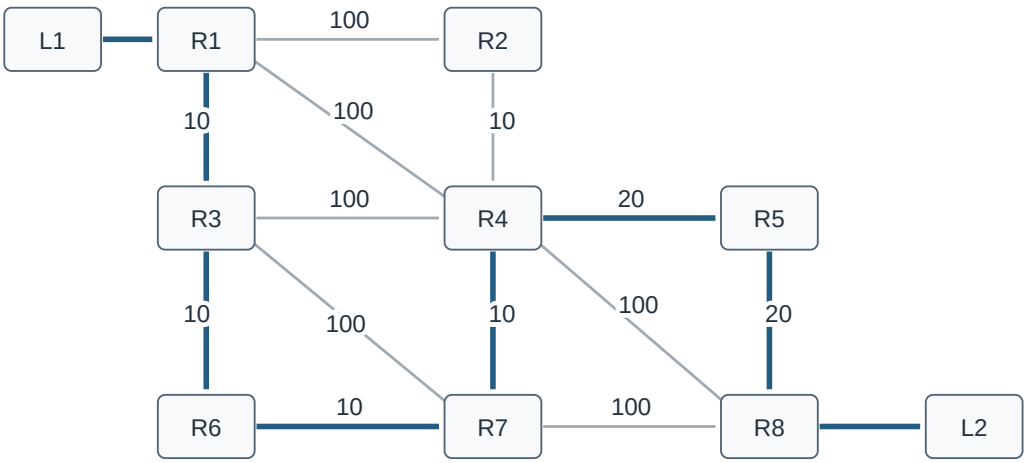
Question 2.b.

Sujet p. 5

> Déterminer le chemin parcouru par un paquet partant du réseau L1 et arrivant au réseau L2, en utilisant le protocole OSPF.

En partant de R1, les coûts minimaux successifs sont : R3 à 10, R6 à 20, R7 à 30, R4 à 40, R5 à 60, puis R8 à 80. Le chemin correspondant coûte $10 + 10 + 10 + 10 + 20 + 20 = 80$. Par comparaison, R1–R4–R5–R8 coûte $100 + 20 + 20 = 140$, et R1–R3–R6–R7–R8 coûte $10 + 10 + 10 + 100 = 130$. Les liaisons entre L1 et R1, puis entre R8 et L2, complètent le trajet ; aucun coût ne leur est indiqué.

Chemin OSPF minimal de L1 à L2



Le chemin mis en évidence a un coût total de 80 entre R1 et R8 ; aucun coût n'est indiqué pour les liaisons d'accès aux réseaux locaux.

> **RÉPONSE À RETENIR**
>
> Le paquet suit **L1 → R1 → R3 → R6 → R7 → R4 → R5 → R8 → L2**. Le coût minimal entre R1 et R8 est **80**.

Question 2.c.

Sujet p. 5

> La liaison R1-R4 est remplacée par une liaison de type ADSL avec une bande passante intermédiaire entre celles de type Ethernet et VDSL. Quel devrait être le coût maximal de cette liaison pour que des paquets issus du réseau L1 à destination du réseau L2 transitent par celle-ci ? En déduire la bande passante minimale de cette liaison.

Notons c le coût de la nouvelle liaison R1–R4. Le chemin de coût minimal empruntant cette liaison est R1–R4–R5–R8, de coût $c + 20 + 20 = c + 40$. Le meilleur chemin ne l'empruntant pas coûte 80. Pour que l'itinéraire par R1–R4 soit de coût minimal, il faut $c + 40 \leq 80$, donc $c \leq 40$. Avec $C = 10^9 / BP$, la limite $C = 40$ correspond à $BP = 10^9 / 40 = 25\,000\,000$ bits/s. À cette limite, les deux itinéraires sont à égalité ; le sujet ne précise pas lequel serait retenu en cas d'égalité.

> **RÉPONSE À RETENIR**
>
> Le coût maximal est **40**. La bande passante minimale correspondante est **25 Mb/s**. À ce seuil, l'itinéraire par R1–R4 a le même coût minimal que l'itinéraire concurrent ; pour qu'il soit strictement moins coûteux, il faudrait une bande passante supérieure à 25 Mb/s.

Exercice 3

4 points

Bases de données et requêtes SQL

Question 1.

Sujet p. 6

> Quand l'entreprise vend le dernier exemplaire en stock d'un modèle, l'attribut Stock de la table ModeleVelo doit être mis à jour pour contenir la valeur 0. L'entreprise vient de vendre le dernier exemplaire en stock du modèle 'Bovelo'. Compléter la requête SQL UPDATE ... SET ... WHERE ... afin de mettre à jour la relation ModeleVelo.

La relation `ModeleVelo` contient l'attribut `Stock`. Il faut affecter la valeur 0 à cet attribut et sélectionner le modèle concerné à l'aide de son nom.

RÉPONSE À RETENIR

```
UPDATE ModeleVelo
SET Stock = 0
WHERE nomModele = 'Bovelo';
```

La requête met à jour le stock du modèle nommé « Bovelo » pour indiquer que le dernier exemplaire a été vendu.

Question 2.

Sujet p. 6, 7

> Ravel, un nouveau fabricant, a livré à l'entreprise 10 vélos du modèle 'Bovelo'. Parmi les cinq requêtes numérotées proposées, sélectionner les deux qui permettent d'intégrer cette nouvelle information et indiquer l'ordre dans lequel elles doivent être saisies.

Il faut d'abord insérer Ravel dans `Fabricant`, puisque la clé étrangère `ModeleVelo.idFabricant` doit référencer un fabricant déjà présent. Ensuite, on insère dans `ModeleVelo` le modèle `Bovelo`, associé à l'identifiant `3127`, avec un stock de 10.

RÉPONSE À RETENIR

Les deux requêtes à choisir sont les requêtes **4 puis 2**.

```
INSERT INTO Fabricant
VALUES (3127, 'Ravel', '89 cours de Vincennes, 75020 Paris');

INSERT INTO ModeleVelo
VALUES (6298, 'Bovelo', 3127, 10, 1990);
```

Dans la deuxième insertion, les valeurs correspondent dans l'ordre à `idModele`, `nomModele`, `idFabricant`, `Stock` et `prixUnitaire`.

Question 2. — Requête 1

Sujet p. 7

> Option proposée : UPDATE ModeleVelo SET quantite=10 WHERE nomModele='Bovelo';

Cette requête ne convient pas : la relation `ModeleVelo` contient l'attribut `Stock`, et non `quantite`. Elle ne permet donc pas de mettre à jour le stock selon le schéma fourni.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

Ne pas sélectionner la requête 1.

Question 2. — Requête 2

Sujet p. 7

> Option proposée : INSERT INTO ModeleVelo VALUES (6298, 'Bovelo', 3127, 10, 1990);

Cette insertion ajoute le modèle `Bovelo` avec l'identifiant fabricant `3127` et un stock de 10. Elle doit être précédée de l'insertion de Ravel dans `Fabricant`, afin que la clé étrangère soit valide.

RÉPONSE À RETENIR

Sélectionner la requête 2 en deuxième position, après la requête 4.

Question 2. — Requête 3

Sujet p. 7

> Option proposée : UPDATE ModeleVelo SET quantite=10 WHERE nomModele='Bovelo' AND nomFabricant='Ravel';

Cette requête ne convient pas : `quantite` n'est pas un attribut de `ModeleVelo`, et `nomFabricant` n'est pas un attribut de cette relation. Le fabricant y est représenté par la clé étrangère `idFabricant`.

RÉPONSE À RETENIR

Ne pas sélectionner la requête 3.

Question 2. — Requête 4

Sujet p. 7

> Option proposée : INSERT INTO Fabricant VALUES (3127, 'Ravel', '89 cours de Vincennes, 75020 Paris');

Cette requête ajoute le fabricant Ravel dans la relation `Fabricant`, avec l'identifiant `3127`. Cette ligne doit exister avant l'insertion du modèle qui référence cet identifiant.

RÉPONSE À RETENIR

Sélectionner la requête 4 en première position, avant la requête 2.

Question 2. — Requête 5

Sujet p. 7

> Option proposée : UPDATE ModeleVelo SET quantite=10 WHERE idFabricant=3127;

Cette requête ne convient pas : elle utilise l'attribut `quantite`, absent de `ModeleVelo`. De plus, une mise à jour filtrée seulement par `idFabricant` ne permettrait pas d'ajouter le nouveau modèle Bovelo.

RÉPONSE À RETENIR

Ne pas sélectionner la requête 5.

Question 3.a.

Sujet p. 6, 7

> Écrire une requête SQL permettant d'obtenir les noms des modèles en rupture de stock et l'identifiant de leur

FormaV - document d'entraînement non officiel

> fabricant respectif.

Une rupture de stock correspond à `Stock = 0`. Le nom du modèle et l'identifiant de son fabricant figurent tous deux dans `ModeleVelo`, donc aucune jointure n'est nécessaire.

RÉPONSE À RETENIR

```
SELECT nomModele, idFabricant
FROM ModeleVelo
WHERE Stock = 0;
```

Cette requête renvoie le nom et l'identifiant du fabricant pour chaque modèle en rupture de stock. Le sujet ne fournit pas de données de tables permettant de lister les lignes renvoyées.

Question 3.b.

Sujet p. 6, 7

> Écrire une requête SQL permettant d'obtenir le nombre de commandes passées depuis le '2022-01-01' inclus. Le type DATE permet de représenter une date par une chaîne de caractères au format 'AAAA-MM-JJ' et ces dates peuvent être comparées à l'aide des opérateurs usuels (=, !=, >, <, <=, >=).

La date est incluse dans le décompte à partir du `2022-01-01`. Il faut donc sélectionner les commandes dont la date est supérieure ou égale à cette valeur, puis compter leurs numéros de commande.

RÉPONSE À RETENIR

```
SELECT COUNT(numeroCommande)
FROM Commande
WHERE date >= '2022-01-01';
```

La requête renvoie le nombre de commandes passées depuis le 1er janvier 2022 inclus. Aucune donnée de tables n'étant fournie, la valeur du compte ne peut pas être déterminée ici.

Question 3.c.

Sujet p. 6, 7

> Écrire une requête SQL permettant d'obtenir les noms des fabricants dont le stock de vélos est strictement positif.

Le stock est indiqué pour chaque modèle dans `ModeleVelo`, tandis que le nom du fabricant se trouve dans `Fabricant`. On relie ces tables grâce à `idFabricant` et on conserve les modèles dont le stock est strictement positif. `DISTINCT` évite de répéter un fabricant ayant plusieurs modèles en stock.

RÉPONSE À RETENIR

```
SELECT DISTINCT Fabricant.nom
FROM Fabricant
JOIN ModeleVelo ON Fabricant.idFabricant = ModeleVelo.idFabricant
WHERE ModeleVelo.Stock > 0;
```

Cette requête renvoie les noms des fabricants qui ont au moins un modèle dont le stock est strictement positif. Le sujet ne fournit pas de données de tables permettant de lister les noms obtenus.

Question 4.

Sujet p. 6, 7

> Que permet d'obtenir la requête SQL suivante ? SELECT DISTINCT Client.nom FROM Client JOIN Commande ON Client.idClient = Commande.idClient JOIN ModeleVelo ON ModeleVelo.idModele = Commande.idModele WHERE ModeleVelo.nomModele = 'Bovelo';

La requête relie chaque client à ses commandes, puis chaque commande au modèle commandé. La condition `WHERE` ne conserve que les commandes concernant un modèle nommé `'Bovelo'`. `DISTINCT` supprime les éventuelles répétitions d'un même nom.

RÉPONSE À RETENIR

La requête renvoie les **noms distincts des clients ayant passé au moins une commande pour un modèle nommé « Bovelo »**. Elle renvoie les noms, et non les prénoms ou les identifiants des clients. Le sujet ne fournit pas de données de tables permettant d'indiquer les noms effectivement renvoyés.

Exercice 4

4 points

Algorithme de Douglas-Peucker, programmation Python et récursivité

Question 1.a.

Sujet p. 8

> Écrire une instruction qui permet d'importer la fonction sqrt du module math.

La fonction `sqrt` appartient au module standard `math`. Pour pouvoir l'appeler directement sous le nom `sqrt`, on l'importe explicitement depuis ce module.

RÉPONSE À RETENIR

```
from math import sqrt
```

Question 1.b.

Sujet p. 8

> En supposant l'import réalisé, écrire une fonction distance_points(a, b) qui prend en argument deux tuples a et b représentant les coordonnées de deux points et renvoie la distance qui les sépare. La distance entre deux points A(xA, yA) et B(xB, yB) est donnée par la formule √((xB − xA)² + (yB − yA)²).

Si `a` et `b` contiennent les coordonnées (x_A, y_A) et `(x_B, y_B)`, on calcule les écarts entre les coordonnées, puis la racine carrée de la somme de leurs carrés. Comme demandé, la fonction utilise le nom `sqrt` importé à la question 1.a.

RÉPONSE À RETENIR

```
def distance_points(a, b):
    return sqrt((b[0] - a[0])**2 + (b[1] - a[1])**2)
```

Question 2.

Sujet p. 8

> On dispose d'une fonction distance_point_droite(p, a, b) qui prend en argument les tuples représentant les coordonnées respectives des points P, A et B, et renvoie la distance du point P à la droite (AB). L'exécution de cette fonction produit une erreur dans le cas où les points A et B sont égaux. À l'aide des fonctions distance_points et distance_point_droite, écrire une fonction distance(p, a, b) qui renvoie la distance entre le point P et la droite (AB) si les points A et B sont distincts, et la distance AP sinon.

La fonction `distance_point_droite` ne peut pas être appelée lorsque les points `a` et `b` sont égaux. On teste donc cette égalité en premier. Si les points sont confondus, la distance demandée est la distance entre `p` et `a` ; sinon, on calcule la distance de `p` à la droite passant par `a` et `b`.

RÉPONSE À RETENIR

```
def distance(p, a, b):
    if a == b:
        return distance_points(p, a)
    return distance_point_droite(p, a, b)
```


FormaV - document d'entraînement non officiel

Question 3.

Sujet p. 9

> Recopier et compléter le code de la fonction le_plus_loin(ligne). Elle prend en argument une liste de tuples représentant les coordonnées des points d'une ligne polygonale et doit renvoyer un tuple composé de l'indice du point de coordonnées p de la ligne polygonale d'extrémités deb et fin pour lequel distance(p, deb, fin) est la plus grande, et de la valeur correspondante de cette distance.

Les extrémités de la ligne sont `ligne[0]` et `ligne[n - 1]`. La boucle examine les points intérieurs, d'indice compris entre 1 et `n - 2`, et conserve l'indice associé à une distance maximale. En cas d'égalité, le test `>=` retient le dernier point intérieur ayant cette distance maximale. Cela fonctionne aussi si toutes les distances sont nulles. La fonction est appelée ici sur une ligne ayant au moins trois points, comme dans la branche récursive de la question 5 lorsque `n > 2`.

Trace générale : pour chaque indice intérieur, on calcule `d = distance(ligne[idx], deb, fin)` ; si `d >= dmax`, on met à jour `dmax` et `indice_max`. Le sujet ne donne pas les coordonnées numériques des sommets : on ne peut donc pas calculer de distances numériques sans inventer des données.

RÉPONSE À RETENIR

```
def le_plus_loin(ligne):
    n = len(ligne)
    deb = ligne[0]
    fin = ligne[n - 1]
    dmax = 0
    indice_max = 0
    for idx in range(1, n - 1):
        p = ligne[idx]
        d = distance(p, deb, fin)
        if d >= dmax:
            dmax = d
            indice_max = idx
    return indice_max, dmax
```

Question 4.

Sujet p. 9

> Écrire une fonction extrait(tab, i, j) qui renvoie la copie du tableau tab des cases d'indice i inclus à j inclus, pour 0 ≤ i ≤ j < len(tab). L'appel extrait([7, 4, 9, 12], 2, 3) renverra ainsi [9, 12].

En Python, la borne de fin d'une tranche est exclue. Pour inclure l'indice `j`, on utilise donc la borne `j + 1`. La tranche produit une nouvelle liste. Avec l'exemple du sujet, les indices 2 et 3 de `[7, 4, 9, 12]` donnent `[9, 12]`.

RÉPONSE À RETENIR

```
def extrait(tab, i, j):
    return tab[i:j + 1]
```

Question 5.

Sujet p. 9, 10

> Recopier et compléter le code de la fonction simplifie(ligne, seuil), qui implémente l'algorithme de Douglas-Peucker. Si la ligne ne contient qu'un ou deux sommets, l'algorithme se termine. Sinon, on sélectionne le point le plus éloigné de la droite formée par les extrémités de la ligne (ou des extrémités lorsqu'elles sont confondues). Si la

> distance maximale dmax est inférieure ou égale au seuil, on ne conserve que les extrémités ; sinon, on applique récursivement l'algorithme aux deux parties de la ligne formées jusqu'au point sélectionné, puis on renvoie la concaténation des séquences simplifiées obtenues.

Si la ligne contient au plus deux sommets, elle est déjà simplifiée. Sinon, on calcule l'indice du point le plus éloigné et sa distance maximale. Lorsque cette distance est inférieure ou égale au seuil, seules les extrémités sont conservées. Dans le cas contraire, on simplifie récursivement les deux sous-lignes qui se rejoignent au point d'indice `indice_max`.

Les deux sous-résultats contiennent ce point commun ; la concaténation `gauche + droite[1:]` évite de le répéter. Dans l'exemple du sujet, la ligne `[A, B, C, D, E, F, G]` est appelée récursivement sur `[A, B, C]` et `[C, D, E, F, G]`, et le résultat final indiqué est `[A, C, E, G]`.

La récursion termine : dans le cas récursif, l'indice retenu est intérieur à la ligne, de sorte que chacune des deux sous-lignes est strictement plus courte que la ligne initiale. Le seuil est supposé donné comme dans l'énoncé.

RÉPONSE À RETENIR

```
def simplifie(ligne, seuil):
    n = len(ligne)
    if n <= 2:
        return ligne
    else:
        indice_max, dmax = le_plus_loin(ligne)
        if dmax <= seuil:
            return [ligne[0], ligne[n - 1]]
        else:
            gauche = simplifie(extrait(ligne, 0, indice_max), seuil)
            droite = simplifie(extrait(ligne, indice_max, n - 1), seuil)
            return gauche + droite[1:]
```

Exercice 5

4 points

Arbres binaires, programmation orientée objet et récursivité

Question 1.

Sujet p. 11

> Déterminer la plus grande somme racine-feuille de l'arbre représenté.

On énumère les chemins racine-feuille et on additionne les étiquettes de chaque chemin :

- 2-7-4-2 : 2+7+4+2=15 ;
- 2-7-4-3 : 2+7+4+3=16 ;
- 2-7-1-5 : 2+7+1+5=15 ;
- 2-5-3-1 : 2+5+3+1=11 ;
- 2-5-8 : 2+5+8=15.

La plus grande des sommes obtenues est 16.

RÉPONSE À RETENIR

La plus grande somme racine-feuille est **16**, obtenue sur le chemin 2-7-4-3.

Question 2.a.

Sujet p. 11, 12

> Écrire une suite d'instructions utilisant la classe Noeud permettant de représenter l'arbre ci-dessous.

On crée un objet `Noeud` pour chaque étiquette, puis on relie les objets avec les méthodes de modification des sous-arbres. La racine 2 a pour enfants gauche et droit 7 et 5. Le nœud 7 a pour enfants 4 et 1 ; le nœud 5 a seulement un enfant droit, 8.

```
racine = Noeud(2)
n7 = Noeud(7)
n5 = Noeud(5)
n4 = Noeud(4)
n1 = Noeud(1)
n8 = Noeud(8)

racine.modifier_sag(n7)
racine.modifier_sad(n5)
n7.modifier_sag(n4)
n7.modifier_sad(n1)
n5.modifier_sad(n8)
```

Les sous-arbres non renseignés restent à `None`, comme à la création de chaque nœud.

RÉPONSE À RETENIR

Ces instructions représentent l'arbre de la question 2, avec le nœud 2 pour racine.

FormaV - document d'entraînement non officiel

Question 2.b.

Sujet p. 11, 12

> Que renvoie l'appel de la méthode niveau sur l'arbre ci-dessus ?

Selon la méthode fournie, le niveau d'une feuille vaut 0. Les nœuds 4, 1 et 8 sont des feuilles, donc leur niveau vaut 0. Les nœuds 7 et 5 ont chacun un enfant et leur niveau vaut donc 1. Enfin, la racine 2 a pour niveau 1+max(1,1)=2.

> **RÉPONSE À RETENIR**
>
> L'appel de `niveau` sur la racine renvoie **2**.

Question 3.

Sujet p. 11, 12

> S'inspirer du code de la méthode niveau pour écrire une méthode récursive pgde_somme qui renvoie la plus grande somme racine-feuille d'un arbre.

Pour une feuille, la plus grande somme racine-feuille vaut son étiquette. Si le nœud n'a qu'un enfant, tout chemin racine-feuille passe par cet enfant. S'il a deux enfants, on choisit celui dont la plus grande somme racine-feuille est la plus élevée. On ajoute dans tous les cas l'étiquette du nœud courant. Il faut traiter séparément les cas d'un ou de deux enfants : une branche absente ne doit pas être considérée comme une somme nulle, notamment si les étiquettes sont négatives.

Cette méthode est à ajouter à la classe `Noeud`.

```
def pgde_somme(self):
    if self.sag is None and self.sad is None:
        return self.etiquette
    if self.sag is None:
        return self.etiquette + self.sad.pgde_somme()
    if self.sad is None:
        return self.etiquette + self.sag.pgde_somme()
    return self.etiquette + max(
        self.sag.pgde_somme(),
        self.sad.pgde_somme()
    )
```

Les appels récursifs portent sur des sous-arbres strictement plus petits. Comme l'arbre est fini, ils atteignent une feuille et la méthode termine.

> **RÉPONSE À RETENIR**
>
> `pgde_somme` renvoie la plus grande somme racine-feuille de l'arbre considéré.

Question 4.a.

Sujet p. 12

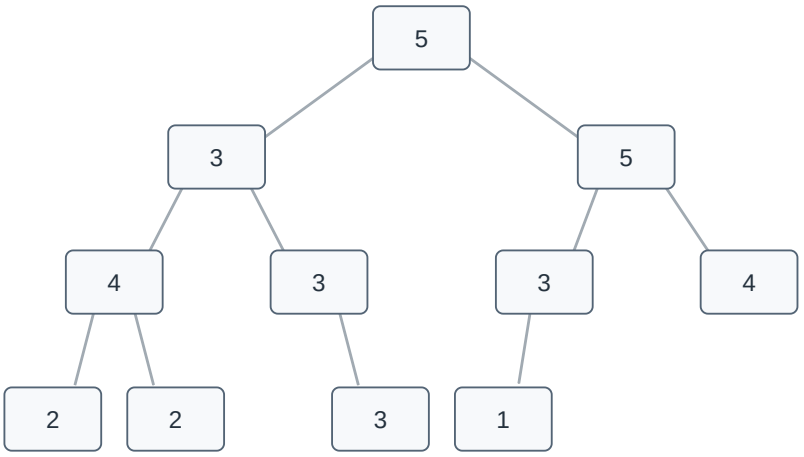
> On appelle arbre magique un arbre binaire dont toutes les sommes des chemins racine-feuille sont égales. Recopier et compléter l'arbre ci-dessous pour qu'il soit magique.

On conserve les étiquettes déjà inscrites dans l'arbre. On complète les quatre cases vides ainsi : le fils gauche de la racine vaut 3 ; le fils droit du nœud 4 vaut 2 ; le fils droit du nœud 3 situé à gauche de la racine vaut 3 ; le fils droit du nœud 5 situé à droite de la racine vaut 4.

Vérification des sommes des chemins racine-feuille : 5+3+4+2=14, 5+3+4+2=14, 5+3+3+3=14, 5+5+3+1=14 et 5+5+4=14. Tous les chemins ont donc la même somme.

FormaV - document d'entraînement non officiel

Une complétion possible de l’arbre magique



Les branches absentes dans l’arbre du sujet restent absentes. Chaque chemin racine-feuille a une somme égale à 14.

> **RÉPONSE À RETENIR**
>
> Une complétion valable donne une somme de **14** sur chaque chemin racine-feuille.

Question 4.b.

Sujet p. 12

> Un arbre est magique si ses sous-arbres sont magiques et qu’ils ont de plus la même plus grande somme racine-feuille. Écrire une méthode récursive est_magique qui renvoie True si l’arbre est magique et False sinon.

Une feuille est magique : elle possède un seul chemin racine-feuille. Pour un nœud avec un seul enfant, l’arbre est magique si et seulement si le sous-arbre de cet enfant l’est. Pour un nœud avec deux enfants, les deux sous-arbres doivent être magiques et leurs plus grandes sommes racine-feuille, calculées depuis leurs propres racines, doivent être égales. L’étiquette du nœud courant s’ajoute aux sommes des deux côtés et ne change pas leur égalité.

Cette méthode est à ajouter à la classe `Noeud`.

```
def est_magique(self):
    if self.sag is None and self.sad is None:
        return True
    if self.sag is None:
        return self.sad.est_magique()
    if self.sad is None:
        return self.sag.est_magique()
    return (
        self.sag.est_magique()
        and self.sad.est_magique()
        and self.sag.pgde_somme() == self.sad.pgde_somme()
    )
```

Chaque appel récursif concerne un sous-arbre strictement plus petit ; la récursion atteint donc une feuille et termine.

> **RÉPONSE À RETENIR**
>
> La méthode renvoie `True` si l'arbre est magique et `False` sinon.

Sujet : https://aeif.fr/wp-content/uploads/2022/11/2022_Amerique_Sud_J2.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.