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 • Jour 2

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 dans un tableau, programmation, algorithmique et complexité
5 réponses • sujet p. 2–3

Exercice 2 — Réseaux, adressage IP et routage OSPF
24 réponses • sujet p. 4–5

Exercice 3 — Bases de données et requêtes SQL pour la gestion d’un stock de vélos électriques
10 réponses • sujet p. 6–7

Exercice 4 — Programmation en Python, récursivité et simplification d’une ligne polygonale par l’algorithme de Douglas-Peucker
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 dans un tableau, programmation, algorithmique et complexité

Question 1.

Sujet p. 2

> On suppose dans cette question que k = 1. Écrire uniquement le bloc d'instructions manquant dans la fonction plus_proche_voisin(t, cible) afin qu'elle renvoie l'indice d'un plus proche voisin de cible.

La distance minimale et l'indice du meilleur candidat sont déjà initialisés avec l'élément d'indice 0. À chaque itération, on calcule la distance de l'élément courant à la cible et on met à jour le minimum et l'indice si cette distance est strictement plus petite. En cas d'égalité, conserver le premier indice rencontré convient puisqu'il s'agit bien d'un plus proche voisin.

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

RÉPONSE À RETENIR

Le bloc à écrire est celui présenté ci-dessus. Il renvoie l'indice d'un plus proche voisin de `cible`.

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 ?

La boucle effectue exactement `n - 1` itérations, où `n = len(t)`. D'après l'hypothèse, le travail à chaque itération est de coût constant. L'initialisation et le retour sont également de coût constant. Le temps d'exécution est donc proportionnel à `n`.

RÉPONSE À RETENIR

La complexité temporelle est linéaire : $\Theta(n)$, donc aussi $O(n)$, avec $n = \text{len}(t)$.

Question 3.a.

Sujet p. 2, 3

> Dans le code de 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 mémorise dans une variable locale `d`. On utilise ensuite cette valeur dans les deux cas, au lieu de rappeler `distance(obj, cible)`.

FormaV - document d'entraînement non officiel

```
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)
```

RÉPONSE À RETENIR

Calculer `d = distance(obj, cible)` une fois par itération, puis réutiliser `d` pour l'insertion ou la comparaison.

Question 3.b.

Sujet p. 2, 3

> Expliquer l'intérêt de maintenir la liste kppv triée dans l'ordre décroissant des distances.

La liste étant triée par distances décroissantes, son premier tuple correspond au candidat le plus éloigné parmi ceux conservés. Lorsque la liste contient déjà `k` candidats, il suffit donc de comparer la distance du nouvel objet à celle de ce premier candidat : si elle est plus petite, on retire le premier tuple et on insère le nouvel objet à sa place dans l'ordre ; sinon, le nouvel objet n'améliore pas les `k` candidats conservés. Le tri facilite ainsi l'identification et le remplacement éventuel du candidat le moins proche.

RÉPONSE À RETENIR

Le premier élément de `kppv` est le moins proche des candidats conservés. Le tri permet de le repérer immédiatement pour décider s'il faut le remplacer.

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. La méthode insert peut éventuellement être utilisée.

Pour préserver l'ordre décroissant, on avance tant que les distances déjà présentes sont supérieures ou égales à `d`, puis on insère le nouveau tuple à la position obtenue. Les tuples de même distance restent donc dans l'ordre où ils ont été rencontrés. Le sujet ne fournit pas de liste de distances concrètes pour une trace numérique ; voici une trace générale : pour une liste déjà triée, on passe après chaque distance supérieure ou égale à `d`, puis on insère `(idx, d)` avant la première distance strictement inférieure, ou à la fin s'il n'y en a pas.

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

La fonction modifie la liste `kppv` en place.

RÉPONSE À RETENIR

Cette fonction insère `(idx, d)` dans la liste préalablement triée en maintenant l'ordre décroissant des distances.

Exercice 2

4 points

Réseaux, adressage IP et routage OSPF

A

Question 1. — option 1

Sujet p. 4

> Sous Unix, quelle instruction en ligne de commande a pu délivrer l'information sur l'adresse IP de l'interface réseau (carte wifi) de l'ordinateur ? Proposition : ifconfig.

`ifconfig` affiche la configuration des interfaces réseau, notamment leurs adresses IP. C'est la commande qui peut fournir l'information demandée.

> **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 sur l'adresse IP de l'interface réseau (carte wifi) de l'ordinateur ? Proposition : ping.

`ping` sert à tester la joignabilité d'une machine sur le réseau ; ce n'est pas la commande adaptée pour afficher la configuration IP de l'interface de l'ordinateur.

> **RÉPONSE À RETENIR**
>
> **Réponse incorrecte : `ping`.**

Question 1. — option 3

Sujet p. 4

> Sous Unix, quelle instruction en ligne de commande a pu délivrer l'information sur l'adresse IP de l'interface réseau (carte wifi) de l'ordinateur ? Proposition : ps.

`ps` affiche les processus en cours d'exécution. Elle ne fournit pas la configuration IP de l'interface réseau.

> **RÉPONSE À RETENIR**
>
> **Réponse incorrecte : `ps`.**

Question 1. — option 4

Sujet p. 4

> Sous Unix, quelle instruction en ligne de commande a pu délivrer l'information sur l'adresse IP de l'interface réseau (carte wifi) de l'ordinateur ? Proposition : ls.

`ls` liste les fichiers d'un répertoire ; elle ne permet pas d'obtenir l'adresse IP de l'interface réseau.

> **RÉPONSE À RETENIR**
>
> **Réponse incorrecte : `ls`.**

Question 2. — option 1

Sujet p. 4

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

DNS sert principalement à associer des noms de domaine à des adresses IP. Ce n'est pas le protocole qui attribue automatiquement une adresse IP à une interface.

> **RÉPONSE À RETENIR**
>
> **Réponse incorrecte : DNS.**

Question 2. — option 2

Sujet p. 4

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

DHCP permet d'attribuer automatiquement des paramètres réseau à un appareil, notamment 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 ? Proposition : TCP.

TCP assure le transport de données entre applications ; il n'attribue pas les adresses IP.

> **RÉPONSE À RETENIR**
>
> **Réponse incorrecte : TCP.**

Question 2. — option 4

Sujet p. 4

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

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

> **RÉPONSE À RETENIR**
>
> **Réponse incorrecte : HTTP.**

Question 3. — option 1

Sujet p. 4

> L'ordinateur a l'adresse IP 192.168.1.14 et le masque 255.255.255.0. Parmi les adresses proposées, quelle est la seule possible pour un autre appareil connecté au même réseau ? Proposition : 192.168.0.14.

Avec le masque 255.255.255.0, les trois premiers octets déterminent l'adresse du réseau. L'adresse 192.168.0.14 appartient au réseau 192.168.0.0, différent du réseau 192.168.1.0 de l'ordinateur.

RÉPONSE À RETENIR

Réponse incorrecte : 192.168.0.14.

Question 3. — option 2

Sujet p. 4

> L'ordinateur a l'adresse IP 192.168.1.14 et le masque 255.255.255.0. Parmi les adresses proposées, quelle est la seule possible pour un autre appareil connecté au même réseau ? Proposition : 192.168.0.1.

Avec le masque 255.255.255.0, l'adresse 192.168.0.1 appartient au réseau 192.168.0.0, différent du réseau 192.168.1.0 de l'ordinateur.

RÉPONSE À RETENIR

Réponse incorrecte : 192.168.0.1.

Question 3. — option 3

Sujet p. 4

> L'ordinateur a l'adresse IP 192.168.1.14 et le masque 255.255.255.0. Parmi les adresses proposées, quelle est la seule possible pour un autre appareil connecté au même réseau ? Proposition : 192.168.1.1.

Le réseau de l'ordinateur est 192.168.1.0/24. L'adresse 192.168.1.1 appartient à ce réseau et est une adresse d'hôte distincte de 192.168.1.14.

RÉPONSE À RETENIR

Bonne réponse : 192.168.1.1.

Question 3. — option 4

Sujet p. 4

> L'ordinateur a l'adresse IP 192.168.1.14 et le masque 255.255.255.0. Parmi les adresses proposées, quelle est la seule possible pour un autre appareil connecté au même réseau ? Proposition : 192.168.1.255.

Dans le réseau 192.168.1.0/24, 192.168.1.255 est l'adresse de diffusion (broadcast), et non une adresse attribuable à un appareil ordinaire.

RÉPONSE À RETENIR

Réponse incorrecte : 192.168.1.255.

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. Quelle affirmation est correcte ? Proposition : « 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 sur le réseau local. Cela n'empêche pas la box d'utiliser une adresse IP publique distincte pour communiquer avec Internet.

RÉPONSE À RETENIR

Réponse incorrecte. L'adresse affichée par le site peut être celle de la box vers Internet.

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. Quelle affirmation est correcte ? Proposition : « C'est sûrement faux car seul le fournisseur d'accès peut avoir connaissance de cette information. »

Un serveur web peut observer l'adresse IP publique à partir de laquelle la requête lui parvient. Cette information n'est donc pas réservée au fournisseur d'accès à Internet.

> **RÉPONSE À RETENIR**
>
> **Réponse incorrecte.** Un site consulté peut connaître l'adresse IP publique à partir de laquelle il reçoit la requête.

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. Quelle affirmation est correcte ? Proposition : « C'est possible et cette adresse serait celle de la box vers Internet. »

L'ordinateur utilise l'adresse privée 192.168.1.14 sur le réseau domestique. La box effectue généralement une traduction d'adresses (NAT) et communique avec Internet au moyen d'une adresse publique. Un site web peut afficher cette adresse publique : 88.168.10.210 est donc une adresse publique possible de la box.

> **RÉPONSE À RETENIR**
>
> **Bonne réponse :** « C'est possible et cette adresse serait celle 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. Quelle affirmation est correcte ? Proposition : « C'est possible, mais cela signifierait que l'ordinateur est infecté par un malware. »

Le fait qu'un site affiche une adresse IP publique n'indique pas que l'ordinateur est infecté. Il s'agit normalement de l'adresse publique utilisée par la box pour accéder à Internet.

> **RÉPONSE À RETENIR**
>
> **Réponse incorrecte.** Cela ne signifie pas que l'ordinateur est infecté 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 ? Proposition : « Oui, à condition que les connexions n'aient pas lieu au même moment. »

Les adresses privées peuvent être réutilisées dans des réseaux locaux distincts, y compris au même moment. L'absence de simultanéité n'est donc pas une condition nécessaire.

> **RÉPONSE À RETENIR**
>
> **Réponse incorrecte.** La condition « à condition que les connexions n'aient pas lieu au même moment » n'est pas nécessaire.

FormaV - document d'entraînement non officiel

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 ? Proposition : « Oui, car les adresses 192.168.x.x ne sont pas routées sur Internet. »

192.168.1.14 appartient à la plage d'adresses privées 192.168.0.0/16, qui n'est pas routée sur Internet. Deux réseaux locaux distincts, comme ceux du domicile et du lycée, peuvent donc attribuer indépendamment la même adresse à un appareil.

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 ? Proposition : « Oui, à condition d'utiliser un VPN. »

Un VPN n'est pas nécessaire pour que deux réseaux locaux distincts utilisent la même adresse privée. La réutilisation est possible parce que ces adresses ne sont pas routées sur Internet.

RÉPONSE À RETENIR

Réponse incorrecte. L'utilisation d'un VPN n'est pas 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 ? Proposition : « Non, car deux machines sont identifiées de manière unique par leur adresse IP. »

Une adresse privée n'identifie pas une machine de façon unique à l'échelle mondiale : la même adresse peut être utilisée sur plusieurs réseaux locaux distincts.

RÉPONSE À RETENIR

Réponse incorrecte. Deux machines sur des réseaux locaux différents peuvent partager la même adresse privée.

B

Question 1.

Sujet p. 5

> Vérifier que le coût d'une liaison VDSL est égal à 20, sachant que la bande passante VDSL est de 50 Mb/s et que le coût C d'une liaison est donné par C = 10^9/BP, où BP est exprimée en bits/s.

La bande passante VDSL vaut 50 Mb/s, soit 50 000 000 bits/s. En utilisant la formule du sujet, on obtient :

$$C = \frac{10^9}{BP} = \frac{10^9}{50\,000\,000} = 20.$$

RÉPONSE À RETENIR

Le coût d'une liaison VDSL est bien **20**.

FormaV - document d'entraînement non officiel

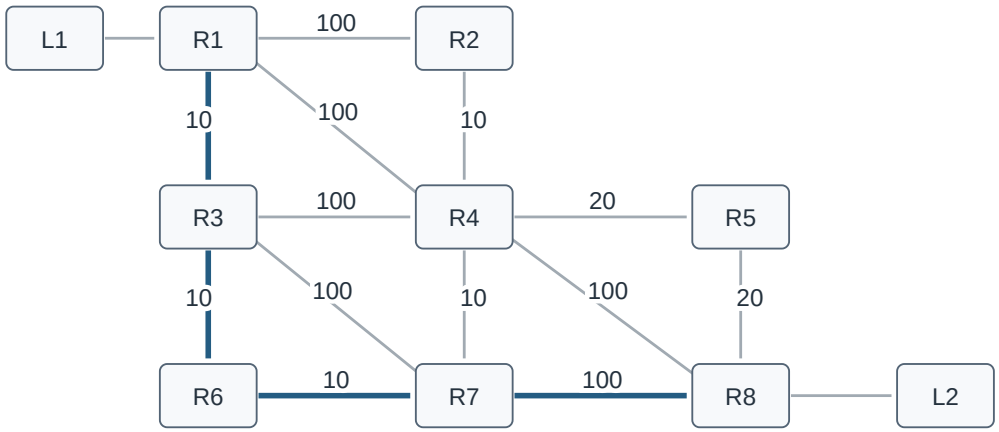
Question 2.a.

Sujet p. 5

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

On applique $C = 10^9 / BP$, avec la bande passante en bits/s. Ainsi, Ethernet à 10 Mb/s a un coût de 100, VDSL à 50 Mb/s un coût de 20, et VDSL2 à 100 Mb/s un coût de 10. On reporte ces valeurs sur chaque liaison selon le tableau du sujet : les liaisons Ethernet coûtent 100, les VDSL coûtent 20 et les VDSL2 coûtent 10.

Réseau et coûts des liaisons



Les coûts sont inscrits sur toutes les liaisons entre routeurs. Le chemin de coût minimal R1–R3–R6–R7–R8 est mis en évidence.

> **RÉPONSE À RETENIR**
>
> Le graphe ci-dessous indique les coûts des liaisons entre routeurs. Les connexions L1–R1 et R8–L2 sont conservées ; le sujet ne leur attribue pas de coût.

Question 2.b.

Sujet p. 5

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

Pour appliquer OSPF, on additionne les coûts des liaisons de chaque chemin entre les routeurs d'accès R1 et R8. Le chemin R1–R3–R6–R7–R8 coûte $10 + 10 + 10 + 100 = 130$. À titre de comparaison, le chemin R1–R4–R8 coûte $100 + 100 = 200$. Les autres itinéraires possibles ne donnent pas de coût inférieur à 130. Les coûts des connexions locales L1–R1 et R8–L2 ne sont pas précisés et ne changent pas la comparaison des itinéraires entre routeurs.

> **RÉPONSE À RETENIR**
>
> Le paquet suit **L1 → R1 → R3 → R6 → R7 → R8 → L2**. Le coût du chemin entre R1 et R8 est **130**.

FormaV - document d'entraînement non officiel

Question 2.c.

> La liaison R1-R4 est remplacée par une liaison de type ADSL dont la bande passante est intermédiaire entre celles des liaisons 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. En l'empruntant puis en continuant par R4–R8, le coût est $c + 100$, car R4–R8 est une liaison Ethernet de coût 100. Le meilleur itinéraire ne passant pas par R1–R4 est R1–R3–R6–R7–R8, de coût 130. Pour que l'itinéraire utilisant R1–R4 soit strictement le moins coûteux, il faut donc $c + 100 < 130$, soit $c < 30$. En supposant, comme pour les coûts calculés dans le sujet, que le coût maximal recherché est un entier, sa valeur maximale est 29. À $c = 30$, il y aurait égalité et OSPF ne garantirait pas le choix de R1–R4. Avec $C = 10^9 / BP$, le coût 29 correspond à $BP = 10^9 / 29$ bits/s.

RÉPONSE À RETENIR

Le coût maximal entier garantissant que les paquets transitent par R1–R4 est **29**. La bande passante minimale correspondante est $\dfrac{10^9}{29} \approx 34\,482\,759$ bits/s, soit environ **34,5 Mb/s**. Elle est bien intermédiaire entre 10 Mb/s (Ethernet) et 50 Mb/s (VDSL).

FormaV - document d'entraînement non officiel

Exercice 3

4 points

Bases de données et requêtes SQL pour la gestion d'un stock de vélos électriques

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 à mettre à jour est `ModeleVelo`. Il faut affecter la valeur `0` à l'attribut `Stock` et sélectionner le modèle dont le nom est `Bovelo`.

RÉPONSE À RETENIR

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

Question 2. — requête 1

Sujet p. 6, 7

> Ravel, un nouveau fabricant, a livré à l'entreprise 10 vélos du modèle 'Bovelo'. Parmi les cinq requêtes proposées, sélectionner deux requêtes permettant d'intégrer cette nouvelle information et indiquer l'ordre dans lequel elles doivent être saisies. Requête 1 : UPDATE ModeleVelo SET quantite=10 WHERE nomModele='Bovelo';

Cette requête ne convient pas : dans le schéma, l'attribut correspondant au stock se nomme `Stock`, et non `quantite`. De plus, le seul nom du modèle ne permettrait pas de distinguer le modèle de Ravel d'un éventuel modèle homonyme.

RÉPONSE À RETENIR

La requête 1 ne doit pas être sélectionnée.

Question 2. — requête 2

Sujet p. 6, 7

> Ravel, un nouveau fabricant, a livré à l'entreprise 10 vélos du modèle 'Bovelo'. Parmi les cinq requêtes proposées, sélectionner deux requêtes permettant d'intégrer cette nouvelle information et indiquer l'ordre dans lequel elles doivent être saisies. Requête 2 : INSERT INTO ModeleVelo VALUES (6298, 'Bovelo', 3127, 10, 1990);

Cette requête ajoute une ligne à `ModeleVelo` pour le modèle `Bovelo`, avec l'identifiant fabricant `3127` et un stock de `10`. Les valeurs suivent l'ordre des attributs de la relation : `idModele`, `nomModele`, `idFabricant`, `Stock`, `prixUnitaire`. Comme `idFabricant` est une clé étrangère, le fabricant correspondant doit avoir été ajouté auparavant dans `Fabricant`.

RÉPONSE À RETENIR

La requête 2 doit être saisie en deuxième, après la requête 4.

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


FormaV - document d'entraînement non officiel

Question 2. — requête 3

Sujet p. 6, 7

> Ravel, un nouveau fabricant, a livré à l'entreprise 10 vélos du modèle 'Bovelo'. Parmi les cinq requêtes proposées, sélectionner deux requêtes permettant d'intégrer cette nouvelle information et indiquer l'ordre dans lequel elles doivent être saisies. Requête 3 : UPDATE ModeleVelo SET quantite=10 WHERE nomModele='Bovelo' AND nomFabricant='Ravel';

Cette requête ne convient pas : `ModeleVelo` ne comporte ni attribut `quantite` ni attribut `nomFabricant`. Le stock se nomme `Stock`, et le nom du fabricant figure dans la relation `Fabricant`.

> **RÉPONSE À RETENIR**
>
> La requête 3 ne doit pas être sélectionnée.

Question 2. — requête 4

Sujet p. 6, 7

> Ravel, un nouveau fabricant, a livré à l'entreprise 10 vélos du modèle 'Bovelo'. Parmi les cinq requêtes proposées, sélectionner deux requêtes permettant d'intégrer cette nouvelle information et indiquer l'ordre dans lequel elles doivent être saisies. Requête 4 : INSERT INTO Fabricant VALUES (3127, 'Ravel', '89 cours de Vincennes, 75020 Paris');

Cette requête ajoute le fabricant Ravel avec l'identifiant `3127`. Elle doit précéder l'insertion du modèle dans `ModeleVelo`, car son attribut `idFabricant` est une clé étrangère référençant `Fabricant`.

> **RÉPONSE À RETENIR**
>
> La requête 4 doit être saisie en premier, puis la requête 2.
>
> ```
> INSERT INTO Fabricant
> VALUES (3127, 'Ravel', '89 cours de Vincennes, 75020 Paris');
> ```

Question 2. — requête 5

Sujet p. 6, 7

> Ravel, un nouveau fabricant, a livré à l'entreprise 10 vélos du modèle 'Bovelo'. Parmi les cinq requêtes proposées, sélectionner deux requêtes permettant d'intégrer cette nouvelle information et indiquer l'ordre dans lequel elles doivent être saisies. Requête 5 : UPDATE ModeleVelo SET quantite=10 WHERE idFabricant=3127;

Cette requête ne convient pas : elle utilise l'attribut `quantite`, qui n'existe pas dans `ModeleVelo`. De plus, elle ne cible pas le modèle `Bovelo` et pourrait modifier le stock d'autres modèles du fabricant.

> **RÉPONSE À RETENIR**
>
> La requête 5 ne doit pas être sélectionnée.

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 fabricant respectif.

Le nom du modèle, l'identifiant de son fabricant et son stock figurent dans `ModeleVelo`. Une rupture de stock correspond à un stock égal à `0` : on sélectionne donc les deux attributs demandés parmi les lignes vérifiant cette condition.

RÉPONSE À RETENIR

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

La requête renvoie le nom et l'identifiant du fabricant de chaque modèle en rupture de stock. Aucune ligne de données n'étant fournie, les valeurs obtenues ne peuvent pas être énumérées.

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 représente une date sous la forme 'AAAA-MM-JJ' et ces dates peuvent être comparées à l'aide des opérateurs usuels (=, !=, >, <, <=, >=).

Chaque commande correspond à une ligne de `Commande`. On compte les numéros de commande dont la date est supérieure ou égale au `2022-01-01` ; l'opérateur `>=` inclut bien cette date.

RÉPONSE À RETENIR

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

Cette requête renvoie le nombre de commandes passées depuis le 1er janvier 2022 inclus. Les données des commandes ne sont pas fournies, donc le résultat numérique ne peut pas être calculé.

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 renseigné dans `ModeleVelo`, tandis que le nom du fabricant se trouve dans `Fabricant`. On relie ces relations grâce à `idFabricant` et on ne conserve que les modèles dont le stock est strictement positif. `DISTINCT` évite de répéter un nom si le fabricant possède plusieurs modèles en stock.

RÉPONSE À RETENIR

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

La requête renvoie les noms des fabricants qui possèdent au moins un modèle ayant un stock positif. Les données des tables ne sont pas fournies, donc les noms obtenus ne peuvent pas être énumérés.

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 première jointure associe chaque commande à son client, puis la seconde associe cette commande au modèle commandé. La condition conserve les commandes portant sur un modèle nommé `Bovelo`. Enfin, `DISTINCT` élimine les répétitions dans les noms renvoyés.

RÉPONSE À RETENIR

La requête renvoie les noms distincts des clients ayant passé au moins une commande portant sur un modèle nommé `Bovelo`. Elle renvoie des noms, et non des identifiants : deux clients différents ayant le même nom ne figurent donc qu'une fois dans le résultat.

Exercice 4

4 points

Programmation en Python, récursivité et simplification d'une ligne polygonale par l'algorithme de Douglas-Peucker

Question 1.a.

Sujet p. 8

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

On importe `sqrt` depuis le module `math`. La fonction peut alors être appelée directement sous le nom `sqrt`.

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, selon la formule donnée dans le sujet.

On récupère les coordonnées des deux points, puis on applique la formule de la distance euclidienne donnée dans le sujet.

RÉPONSE À RETENIR

```
def distance_points(a, b):
    xa, ya = a
    xb, yb = b
    return sqrt((xb - xa)**2 + (yb - ya)**2)
```

Question 2.

Sujet p. 8

> À l'aide des fonctions distance_points et distance_point_droite(p, a, b), 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 produit une erreur lorsque A et B sont égaux.

La fonction `distance_point_droite` provoque une erreur lorsque `a` et `b` sont égaux. On teste donc d'abord ce cas : la distance demandée est alors la distance entre `p` et `a`. Si les deux points sont distincts, on appelle la fonction de distance à la droite.

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 pour lequel la distance 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]`. On parcourt les indices intérieurs de 1 à `n - 2` inclus et on mémorise toute distance strictement supérieure au maximum courant. En cas d'égalité, le premier indice maximal rencontré est conservé. Le code ci-dessous suppose une ligne d'au moins deux sommets ; il est appelé par `simplifie` uniquement lorsque la ligne comporte plus de deux sommets.

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 [9, 12].

En Python, la borne de fin d'une tranche est exclue. Pour inclure l'indice `j`, on utilise donc `j + 1`. La tranche renvoie une nouvelle liste, c'est-à-dire une copie des éléments demandés.

Pour l'exemple du sujet, les éléments d'indices 2 et 3 de `[7, 4, 9, 12]` sont 9 et 12.

RÉPONSE À RETENIR

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

Ainsi, `extrait([7, 4, 9, 12], 2, 3)` renvoie `[9, 12]`. La copie contient `j - i + 1` éléments.

Question 5.

Sujet p. 9, 10

> Recopier et compléter le code de la fonction simplifie(ligne, seuil), qui implémente l'algorithme de Douglas-Peucker. Compléter les valeurs de retour pour le cas où la ligne contient au plus deux sommets et pour le cas où dmax ≤ seuil, puis écrire le bloc correspondant au cas récursif.

Si la ligne contient au plus deux sommets, on la renvoie sans modification. Sinon, on recherche le point intérieur le plus éloigné des extrémités (ou de la droite qu'elles définissent). Si sa distance est inférieure ou égale au seuil, on ne conserve que les extrémités. Dans le cas contraire, on simplifie récursivement les deux sous-lignes, qui ont en commun le sommet de séparation, puis on concatène leurs résultats en ne gardant ce sommet qu'une seule fois.

Dans l'exemple du sujet, la ligne `[A, B, C, D, E, F, G]` est séparée en `[A, B, C]` et `[C, D, E, F, G]` ; le résultat final indiqué est `[A, C, E, G]`.

On suppose le seuil non négatif, ce qui est naturel puisqu'il s'agit d'une distance. Dans le cas récursif, `dmax > seuil` implique alors `dmax > 0` : le point sélectionné est intérieur et les deux sous-lignes sont strictement plus courtes que la ligne initiale. Cela garantit la terminaison.

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:]
```

Le retrait du premier élément de `droite` évite de répéter le sommet commun. Sur l'exemple fourni, les appels récursifs aboutissent à `[A, C, E, G]`.

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 binaire représenté dans le sujet.

On énumère les chemins racine-feuille et on additionne les étiquettes de leurs nœuds :

- 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 de ces sommes 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 fournie pour représenter l'arbre ci-dessous : sa racine est étiquetée 2, son fils gauche 7 et son fils droit 5 ; le nœud 7 a pour fils gauche 4 et fils droit 1 ; le nœud 5 a pour fils droit 8.

On crée un objet `Noeud` pour chacun des six nœuds, puis on relie les objets selon les positions des fils dans l'arbre. Le nœud 5 n'a pas de fils gauche ; ses autres fils absents restent à `None`, valeur donnée par le constructeur.

```
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)
```

RÉPONSE À RETENIR

`racine` référence l'arbre demandé : le nœud 2 a pour fils 7 et 5 ; le nœud 7 a pour fils 4 et 1 ; le nœud 5 a pour seul fils le nœud droit 8.

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 ?

La méthode `niveau` renvoie 0 pour une feuille. Les nœuds 4, 1 et 8 ont donc le niveau 0. Le nœud 7, dont les deux fils sont des feuilles, a le niveau 1 ; le nœud 5, qui a un seul fils, a aussi le niveau 1. Pour la racine, la méthode renvoie alors 1 + max(1, 1) = 2.

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

Question 3.

Sujet p. 11, 12

> En s'inspirant du code de la méthode niveau, 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 est son étiquette. Si le nœud a un seul fils, chaque chemin passe par ce fils : on ajoute donc l'étiquette du nœud à la somme renvoyée par cet enfant. S'il a deux fils, on choisit le plus grand des deux résultats récursifs, puis on ajoute l'étiquette du nœud. Cette définition traite aussi les étiquettes négatives, car elle ne compare que les branches qui existent.

```
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()
    )
```

Sur un arbre fini, chaque appel récursif porte sur un sous-arbre plus petit ; les appels atteignent donc une feuille et se terminent. Pour l'arbre de la question 1, la méthode renvoie 16.

> **RÉPONSE À RETENIR**
>
> La méthode `pgde_somme` renvoie la plus grande somme des étiquettes d'un chemin allant du nœud courant jusqu'à une feuille.

Question 4.a.

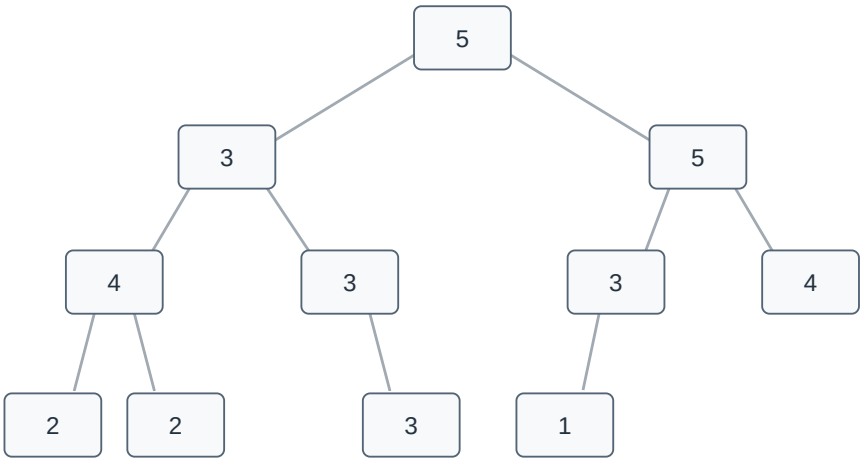
Sujet p. 12

> Un arbre magique est un arbre binaire dont toutes les sommes des chemins racine-feuille sont égales. Recopier et compléter l'arbre représenté dans le sujet pour qu'il soit magique.

Le chemin déjà entièrement étiqueté passant par les nœuds 5, 5, 3 et 1 a pour somme 14. Pour que l'arbre soit magique, chaque autre chemin racine-feuille doit donc également avoir pour somme 14.

Notons `u` l'étiquette manquante du fils gauche de la racine. Le chemin passant ensuite par 4 et 2 donne 5 + u + 4 + 2 = 14, donc `u = 3`. Le chemin passant par l'autre fils de 4 impose 5 + 3 + 4 + v = 14, donc cette étiquette manquante vaut `v = 2`. Le chemin passant par le fils droit de u puis par 3 impose 5 + 3 + w + 3 = 14, donc `w = 3`. Enfin, le fils droit du nœud 5 de droite doit valoir 4, car 5 + 5 + 4 = 14.

Arbre magique complété



Les branches et les fils gauche/droit correspondent à la figure du sujet. Chacun des cinq chemins racine-feuille a pour somme 14.

> **RÉPONSE À RETENIR**
>
> On complète les quatre cercles vides, dans l'ordre de gauche à droite sur la figure, par **3, 2, 3 et 4**. Tous les chemins racine-feuille ont alors pour somme 14.

Question 4.b.

Sujet p. 12

> Un arbre est magique si ses sous-arbres sont magiques et 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, car elle possède un seul chemin racine-feuille. Lorsqu'un nœud a un seul enfant, son sous-arbre est magique si et seulement si le sous-arbre de cet enfant l'est. Lorsqu'il a deux enfants, il faut que les deux sous-arbres soient magiques et que leurs plus grandes sommes racine-feuille soient égales : l'étiquette du nœud courant s'ajoute à chacune de ces sommes et ne change donc pas leur égalité. La méthode ci-dessous utilise `pgde_somme` définie à la question 3.

```
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()
    )
```

La méthode s'applique à un arbre non vide, comme ceux représentés dans le sujet.

RÉPONSE À RETENIR

`est_magique` renvoie `True` exactement lorsque l'arbre est magique au sens de la question ; sinon, elle renvoie `False`.

Sujet : https://www.sujetdebac.fr/annales-pdf/2022/spe-numerique-informatique-2022-amerique-sud-2-sujet-officiel.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.