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

CORRIGÉ NON OFFICIEL

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

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

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

BACCALAURÉAT GÉNÉRAL • CORRIGÉ DÉTAILLÉ

Numérique et sciences informatiques

Session 2023 • Jour 1

Code sujet **23-NSIJ1LI1** 3 heures 30

Proposition de corrigé pédagogique • FormaV

Se repérer dans le corrigé

Exercice 1 — Algorithmique et programmation : tests de palindrome
9 réponses • sujet p. 2–3

Exercice 2 — Bases de données, représentation des données et réseaux
10 réponses • sujet p. 4–7

Exercice 3 — Piles, arbres et algorithmique : notation polonaise inversée
12 réponses • sujet p. 8–10

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

FormaV - document d'entraînement non officiel

Exercice 1

3 points

Algorithmique et programmation : tests de palindrome

Question 1.

Sujet p. 2

> On saisit dans la console tester_palindrome('kayak'). Combien de fois la fonction tester_palindrome est-elle appelée lors de l'exécution de cette instruction ? Compter l'appel initial.

À chaque appel, les deux caractères extérieurs sont comparés. Pour `kayak`, ils sont égaux, et la fonction se rappelle avec la chaîne privée de ces deux caractères : `kayak` devient `aya`, puis `a`. Sur `a`, la condition `len(chaine) < 2` est vérifiée et la fonction renvoie `True` sans nouvel appel récursif.

> **RÉPONSE À RETENIR**
>
> La fonction est appelée **3 fois** : pour `kayak`, `aya` et `a`. L'appel initial est compté.

Question 2.a.

Sujet p. 2

> Justifier que la fonction tester_palindrome est récursive.

Une fonction est récursive lorsqu'elle s'appelle elle-même. À la ligne 8 du code, `tester_palindrome` renvoie le résultat d'un nouvel appel à `tester_palindrome(chaine)`, après que la chaîne a été raccourcie à la ligne 7.

> **RÉPONSE À RETENIR**
>
> `tester_palindrome` est donc une fonction récursive : elle contient un appel à elle-même.

Question 2.b.

Sujet p. 2

> Expliquer pourquoi l'appel à la fonction tester_palindrome se terminera quelle que soit la chaîne de caractères sur laquelle elle s'applique.

À chaque appel récursif, la chaîne est privée de son premier et de son dernier caractère. Sa longueur diminue donc de 2. Tant que sa longueur est au moins 2, cette diminution ne peut se poursuivre qu'un nombre fini de fois : la longueur est un entier non négatif et finit par devenir inférieure à 2. La fonction renvoie alors `True`. Si deux caractères comparés sont différents, elle renvoie `False` encore plus tôt.

> **RÉPONSE À RETENIR**
>
> L'appel se termine pour toute chaîne de caractères : soit une différence est trouvée, soit la chaîne raccourcit jusqu'à avoir une longueur inférieure à 2, ce qui déclenche le cas d'arrêt.

Question 3.a. — ZeroDivisionError

Sujet p. 3

> Parmi les quatre propositions données pour l'erreur générée par tester_palindrome(53235), examiner la proposition « ZeroDivisionError » afin d'indiquer le type d'erreur affiché.

`ZeroDivisionError` signale une tentative de division par zéro. Le code de `tester_palindrome` n'effectue aucune division ; ce n'est donc pas le type d'erreur produit.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

`ZeroDivisionError` n'est pas la bonne réponse.

Question 3.a. — ValueError

Sujet p. 3

> Parmi les quatre propositions données pour l'erreur générée par tester_palindrome(53235), examiner la proposition « ValueError » afin d'indiquer le type d'erreur affiché.

La fonction échoue lorsqu'elle tente d'obtenir la longueur de l'entier `53235`. Ce n'est pas une erreur liée à une valeur qui conviendrait mal à une conversion ou à une opération : l'opération `len` ne peut pas s'appliquer à cet entier.

RÉPONSE À RETENIR

`ValueError` n'est pas la bonne réponse.

Question 3.a. — TypeError

Sujet p. 3

> Parmi les quatre propositions données pour l'erreur générée par tester_palindrome(53235), examiner la proposition « TypeError » afin d'indiquer le type d'erreur affiché.

Lors du premier appel, `chaine` vaut l'entier `53235`. La première instruction du corps teste `len(chaine)`, mais un entier n'est pas une séquence dont Python peut ainsi déterminer la longueur. Cette opération provoque une erreur de type.

RÉPONSE À RETENIR

Le type d'erreur affiché est **`TypeError`**.

Question 3.a. — IndexError

Sujet p. 3

> Parmi les quatre propositions données pour l'erreur générée par tester_palindrome(53235), examiner la proposition « IndexError » afin d'indiquer le type d'erreur affiché.

`IndexError` correspond à un accès à un indice inexistant. Dans cet appel, l'erreur se produit avant les accès `chaine[0]` et `chaine[-1]` : c'est le calcul de `len(chaine)` sur un entier qui échoue.

RÉPONSE À RETENIR

`IndexError` n'est pas la bonne réponse.

Question 3.b.

Sujet p. 2, 3

> Proposer une ou plusieurs instructions à écrire entre la ligne 1 et la ligne 2 du code de la fonction tester_palindrome afin d'afficher clairement cette erreur à l'utilisateur.

On peut vérifier dès l'entrée dans la fonction que le paramètre est bien une chaîne. Si ce n'est pas le cas, afficher un message explicite et quitter la fonction avant l'appel à `len` évite que l'erreur initiale se produise. Ces instructions sont à insérer entre la ligne 1 et la ligne 2 :

```
    if not isinstance(chaine, str):
        print("Erreur : le paramètre doit être une chaîne de caractères.")
        return False
```

RÉPONSE À RETENIR

Pour l'appel avec `53235`, ces instructions affichent le message indiquant que le paramètre doit être une chaîne, puis renvoient `False` sans provoquer le `TypeError`.

Question 4.

Sujet p. 3

> Écrire le code d'une fonction itérative (non récursive) est_palindrome qui prend en paramètre une chaîne de caractères et renvoie un booléen égal à True si la chaîne est un palindrome, False sinon.

On compare les caractères symétriques, en partant des deux extrémités et en avançant vers le centre. Dès qu'une paire diffère, la chaîne n'est pas un palindrome. Si toutes les paires comparées sont égales, la boucle s'arrête au centre et la chaîne est un palindrome. Les chaînes vides ou de longueur 1 passent directement à la réponse `True`.

```
def est_palindrome(chaine):
    gauche = 0
    droite = len(chaine) - 1

    while gauche < droite:
        if chaine[gauche] != chaine[droite]:
            return False
        gauche += 1
        droite -= 1

    return True
```

La boucle termine car à chaque tour, `gauche` augmente de 1 et `droite` diminue de 1 ; l'écart entre les indices diminue donc jusqu'à devenir nul ou négatif.

RÉPONSE À RETENIR

Le code ci-dessus définit une fonction itérative `est_palindrome` qui renvoie un booléen : `True` si les caractères symétriques sont tous égaux, et `False` dès qu'une paire diffère.

Exercice 2

5 points

Bases de données, représentation des données et réseaux

Étude du schéma relationnel

Question 1.a.

Sujet p. 4, 5

> Pour chaque attribut de la relation cage, spécifier son type en utilisant le tableau des types fourni.

`num_cage` désigne un numéro : c'est un entier. `taille_cage` et `secteur_cage` contiennent du texte. Comme le sujet ne précise pas de longueur maximale, on les indique en `VARCHAR(t)`, où `t` est à choisir selon les valeurs possibles.

RÉPONSE À RETENIR

Attribut	Type
num_cage	INT
taille_cage	VARCHAR(t)
secteur_cage	VARCHAR(t)

Question 1.b.

Sujet p. 4, 5

> Préciser, pour la relation reservation, le nom de la clé primaire pouvant être utilisée.

Une clé primaire identifie de façon unique chaque ligne de la relation. Le numéro propre à chaque réservation est `num_reservation` ; cet attribut peut donc servir de clé primaire.

RÉPONSE À RETENIR

Clé primaire : `num_reservation`.

Question 1.c.

Sujet p. 4, 5

> Indiquer, pour la relation reservation, la ou les clés étrangères (ou secondaires) et en indiquer l'utilité.

Les clés étrangères d'une réservation référencent les identifiants des enregistrements associés dans les autres relations. Elles permettent de relier les données et de garantir que les références correspondent à des clients, animaux et cages existants.

RÉPONSE À RETENIR

- `num_client` référence `client(num_client)`.
- `num_animal` référence `animal(num_animal)`.
- `num_cage` référence `cage(num_cage)`.

FormaV - document d'entraînement non officiel

Requêtes

Question 2.a.

Sujet p. 4, 5

> Indiquer le résultat de l'exécution de la requête : SELECT nom_animal FROM animal WHERE categorie_animal = 'chien';

La requête garde les lignes de la table `animal` dont `categorie_animal` vaut `chien`, puis affiche leur nom. Dans l'extrait fourni, ces lignes correspondent à Api, Rex et Rex. Comme il n'y a pas de `DISTINCT`, les deux occurrences de Rex sont conservées. Sans `ORDER BY`, l'ordre des lignes affichées n'est pas garanti.

RÉPONSE À RETENIR

nom_animal
Api
Rex
Rex

Question 2.b.

Sujet p. 4, 5

> Écrire une requête SQL permettant d'afficher les noms de tous les clients dont l'animal a occupé la cage numéro 23.

On sélectionne les clients pour lesquels il existe au moins une réservation en cage 23. La sous-requête évite de répéter un client ayant plusieurs réservations concernées, tout en conservant deux lignes si deux clients distincts ont le même nom.

```
SELECT client.nom_client
FROM client
WHERE client.num_client IN (
    SELECT reservation.num_client
    FROM reservation
    WHERE reservation.num_cage = 23
);
```

Dans les extraits fournis, les réservations en cage 23 concernent le client 345, dont le nom est Morel.

RÉPONSE À RETENIR

nom_client
Morel

Question 2.c.

Sujet p. 4, 5

> La base de données contient actuellement 491 animaux. Un nouvel animal doit être enregistré : il s'appelle Suki, c'est un chat de petite taille et son propriétaire a déjà été enregistré sous le numéro 342. Écrire la requête SQL permettant d'insérer ces nouvelles données dans la base de données.

La table `animal` comporte un attribut `num_animal`, mais le sujet ne précise pas si cet identifiant est généré automatiquement ni quelle règle attribue les numéros. Si les identifiants suivent la numérotation séquentielle et que le prochain est 492, on peut écrire :

```
INSERT INTO animal (num_animal, nom_animal, categorie_animal, taille_animal, num_client)
VALUES (492, 'Suki', 'chat', 'petit', 342);
```

Si la base génère automatiquement `num_animal`, il faut plutôt omettre cette colonne :

```
INSERT INTO animal (nom_animal, categorie_animal, taille_animal, num_client)
VALUES ('Suki', 'chat', 'petit', 342);
```

Le nombre total de 491 animaux ne suffit pas, à lui seul, à démontrer que le prochain identifiant est 492 ; ce choix suppose donc la règle d'attribution indiquée.

> **RÉPONSE À RETENIR**
>
> L'insertion enregistre Suki comme chat de petite taille, rattaché au client 342. La valeur de `num_animal` dépend de la règle d'identification utilisée par la base.

Programmation Python

Question 3.a.

Sujet p. 6

> À partir de la liste reservation et du code de la fonction mystere fournis, indiquer l'affichage produit par l'exécution de print(mystere(reservation, '2022-08-23')).

La fonction ajoute à sa liste le numéro du client de chaque réservation dont la date de début est égale à `2022-08-23`. Cette date apparaît pour les réservations 44 et 47, concernant respectivement les clients 26 et 345. Aucune autre réservation de l'extrait ne commence à cette date.

> **RÉPONSE À RETENIR**
>
> L'instruction affiche `[26, 345]`.

Question 3.b.

Sujet p. 7

> Le responsable veut obtenir le nombre de réservations effectuées pour un numéro de client donné. Écrire les lignes de code après la ligne 7 de la fonction nombre_reservation afin de respecter la spécification fournie.

On parcourt la liste une fois et on incrémente un compteur à chaque réservation dont le numéro de client est celui recherché. Les lignes suivantes complètent la fonction après sa ligne 7 :

```
    compteur = 0
    for ligne in table:
        if ligne['num_client'] == numero_client:
            compteur = compteur + 1
    return compteur
```

Exemple avec la liste fournie : pour `numero_client = 345`, les réservations 45, 46, 47 et 48 correspondent ; le compteur passe de 0 à 4 et la fonction renvoie 4. Pour le client 26, elle renvoie 1. Si aucune réservation ne correspond, elle renvoie 0. Le parcours examine chaque réservation une fois, donc le temps de calcul est proportionnel au nombre de lignes de `table`.

> **RÉPONSE À RETENIR**
>
> La fonction renvoie le nombre de réservations du client indiqué.

Protocole OSPF

Question 4.a.

Sujet p. 7

> Pour passer de L1 à L2, le chemin R1 - R2 - R5 utilisant la fibre a le coût le plus faible. Calculer ce coût à l'aide de la formule fournie.

La formule donnée est `C = 10^8 / d`. Sur le trajet R1–R2–R5, la liaison R1–R2 est en Fast-Ethernet, soit `d = 10^8` bit/s et un coût de `10^8 / 10^8 = 1`. La liaison R2–R5 est en fibre, soit `d = 10^9` bit/s et un coût de `10^8 / 10^9 = 0,1`. Le coût du chemin est la somme des coûts de ses deux liaisons.

> **RÉPONSE À RETENIR**
>
> Le coût de R1–R2–R5 est `1 + 0,1 = 1,1`.

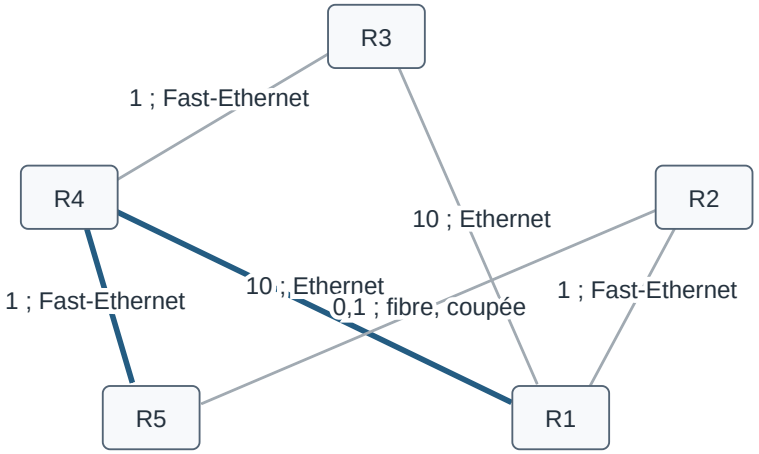
Question 4.b.

Sujet p. 7

> La liaison entre R2 et R5 a été coupée en raison de travaux. Déterminer la route permettant de relier le réseau L1 au réseau L2 selon le protocole OSPF. Justifier.

Après la coupure de R2–R5, ce lien ne peut plus être emprunté. D'après la figure, R1–R4 est une liaison Ethernet de coût `10^8 / 10^7 = 10`, et R4–R5 est une liaison Fast-Ethernet de coût `10^8 / 10^8 = 1`. Ce détour coûte donc `10 + 1 = 11`. L'autre détour possible, R1–R3–R4–R5, coûte `10 + 1 + 1 = 12` : R1–R3 est Ethernet, puis R3–R4 et R4–R5 sont Fast-Ethernet. R2 devient une impasse après la coupure, car il ne reste que sa liaison vers R1. OSPF retient donc le détour de coût 11.

Réseau après la coupure de R2–R5



Le chemin mis en évidence est R1–R4–R5, de coût 10 + 1 = 11. Le lien R2–R5, de coût 0,1, est coupé.

> **RÉPONSE À RETENIR**
>
> La route de coût minimal est L1 → R1 → R4 → R5 → L2. Son coût entre routeurs est 11.

FormaV - document d'entraînement non officiel

Exercice 3

4 points

Piles, arbres et algorithmique : notation polonaise inversée

Question 1.

Sujet p. 8

> Donner la valeur de l'expression suivante écrite en notation polonaise inversée : 15 5 − 4 12 + ×.

En notation polonaise inversée, chaque opérateur s'applique aux deux valeurs qui le précèdent immédiatement. On évalue donc successivement les opérations :

1. $15 - 5 = 10$, ce qui laisse $10\ 4\ 12\ +\ \times$.
2. $4 + 12 = 16$, ce qui laisse $10\ 16\ \times$.
3. $10 \times 16 = 160$.

RÉPONSE À RETENIR

La valeur de l'expression est **160**.

Question 2. — un parcours en largeur

Sujet p. 8

> Parmi les quatre parcours proposés, examiner le parcours en largeur pour déterminer lequel permet d'obtenir la succession de symboles 20 11 3 + 5 2 + / − correspondant à la notation polonaise inversée.

Le parcours en largeur visite les nœuds niveau par niveau, de gauche à droite. Sur l'arbre donné, il produit la succession $-,\ 20,\ /,\ +,\ +,\ 11,\ 3,\ 5,\ 2$. Elle ne correspond pas à la NPI demandée, qui commence par 20.

RÉPONSE À RETENIR

Le parcours en largeur **ne convient pas**.

Question 2. — un parcours en profondeur préfixe

Sujet p. 8

> Parmi les quatre parcours proposés, examiner le parcours en profondeur préfixe pour déterminer lequel permet d'obtenir la succession de symboles 20 11 3 + 5 2 + / − correspondant à la notation polonaise inversée.

Le parcours en profondeur préfixe visite d'abord la racine, puis le sous-arbre gauche et le sous-arbre droit. Il produit ici $-,\ 20,\ /,\ +,\ 11,\ 3,\ +,\ 5,\ 2$. La racine $-$ apparaît en premier, contrairement à la succession demandée.

RÉPONSE À RETENIR

Le parcours en profondeur préfixe **ne convient pas**.

Question 2. — un parcours en profondeur infixe

Sujet p. 8

> Parmi les quatre parcours proposés, examiner le parcours en profondeur infixe pour déterminer lequel permet d'obtenir la succession de symboles 20 11 3 + 5 2 + / − correspondant à la notation polonaise inversée.

Le parcours en profondeur infixe visite le sous-arbre gauche, puis la racine, puis le sous-arbre droit. Il produit ici $20,\ -,\ 11,\ +,\ 3,\ /,\ 5,\ +,\ 2$, ce qui ne correspond pas à la succession demandée.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

Le parcours en profondeur infixe **ne convient pas**.

Question 2. — un parcours en profondeur postfixe (ou suffixe)

Sujet p. 8

> Parmi les quatre parcours proposés, examiner le parcours en profondeur postfixe (ou suffixe) pour déterminer lequel permet d'obtenir la succession de symboles 20 11 3 + 5 2 + / − correspondant à la notation polonaise inversée.

Le parcours en profondeur postfixe visite le sous-arbre gauche, puis le sous-arbre droit, puis la racine. Le sous-arbre gauche fournit 20. Le sous-arbre droit fournit $11,\ 3,\ +,\ 5,\ 2,\ +,\ /$. La racine $-$ est visitée en dernier.

RÉPONSE À RETENIR

On obtient $20,\ 11,\ 3,\ +,\ 5,\ 2,\ +,\ /,\ -$, la succession demandée. Il faut choisir le **parcours en profondeur postfixe (ou suffixe)**.

Question 3.a.

Sujet p. 9

> Des deux acronymes LIFO et FIFO, indiquer lequel permet de décrire, de manière générale, la structure de pile. Donner la signification des quatre lettres qui composent cet acronyme.

Dans une pile, le dernier élément empilé est le premier dépilé : c'est le principe « dernier entré, premier sorti ».

RÉPONSE À RETENIR

L'acronyme est **LIFO**, pour *Last In, First Out*, c'est-à-dire « dernier entré, premier sorti ».

Question 3.b.

Sujet p. 9, 10

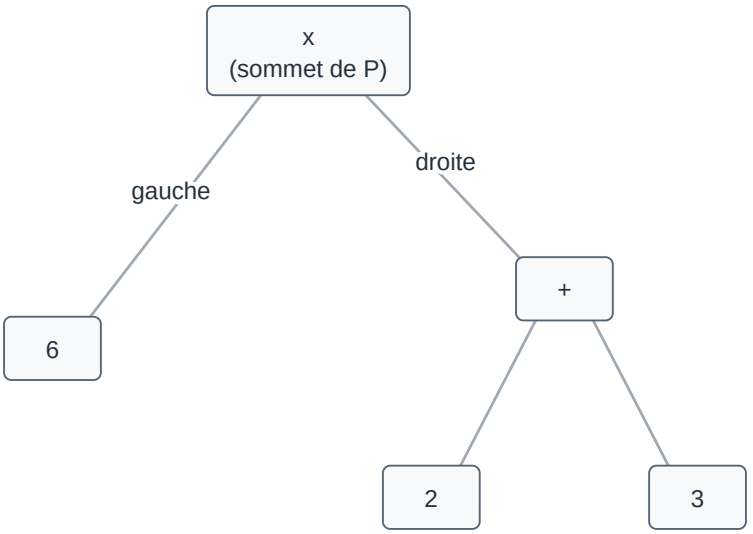
> On applique l'algorithme créer_arbre à la liste [6, 2, 3, +, x, 94, 1, -, +]. À la fin du quatrième tour de la boucle pour, l'état de la pile est représenté sur la page 9. Indiquer le nombre d'éléments contenus dans la pile P et dessiner l'état de la pile à la fin du cinquième tour de la boucle pour.

À la fin du quatrième tour, la pile représentée page 9 contient deux éléments : au sommet, l'arbre de racine $+$ et de feuilles 2 et 3 ; en dessous, l'arbre réduit à la feuille 6.

Au cinquième tour, l'élément lu est l'opérateur de multiplication x. L'algorithme dépile d'abord l'arbre $+$, qui devient le sous-arbre droit, puis la feuille 6, qui devient le sous-arbre gauche. Il empile ensuite l'arbre obtenu : la pile ne contient alors plus qu'un seul élément.

FormaV - document d'entraînement non officiel

Arbre unique dans la pile après le cinquième tour



À la fin du cinquième tour, cet arbre est l'unique élément de la pile P.

> **RÉPONSE À RETENIR**
>
> À la fin du quatrième tour, **P contient 2 éléments**. À la fin du cinquième tour, **P contient 1 élément** : l'arbre de racine x, dont le fils gauche est 6 et le fils droit est l'arbre $+$ de feuilles 2 et 3.

Question 3.c.

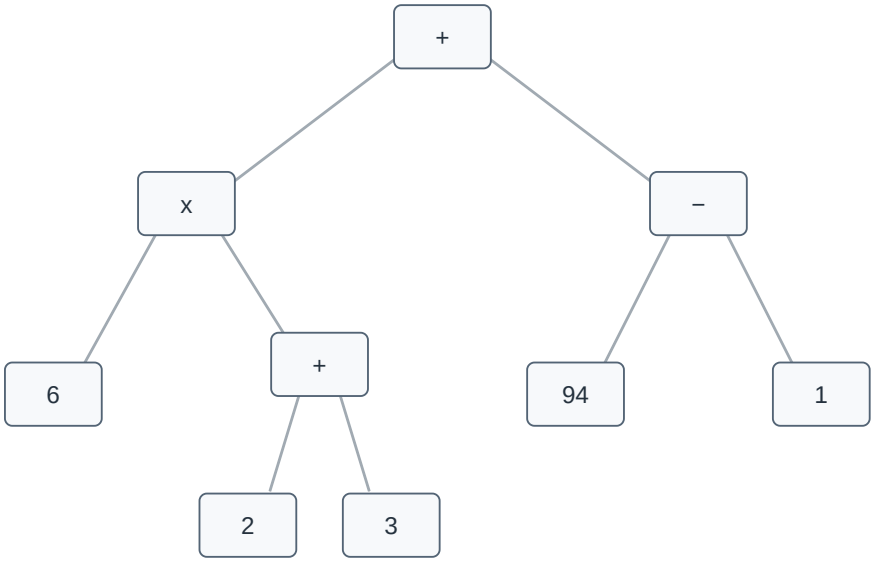
Sujet p. 9, 10

> Dessiner l'arbre binaire représentant l'expression mathématique dont la notation polonaise inversée est la liste [6, 2, 3, +, x, 94, 1, -, +].

On lit la liste de gauche à droite. Les symboles $2,\ 3,\ +$ forment un arbre de racine $+$, avec 2 et 3 comme fils gauche et droit. L'opérateur x combine ensuite cet arbre avec la feuille 6 : 6 est le fils gauche et l'arbre $+$ le fils droit. Puis $94,\ 1,\ -$ forment un arbre de racine $-$. Enfin, le dernier $+$ combine les arbres de multiplication et de soustraction, respectivement à gauche et à droite.

FormaV - document d'entraînement non officiel

Arbre binaire de l’expression en NPI



Pour chaque opérateur, le fils gauche correspond au premier opérande et le fils droit au second.

> **RÉPONSE À RETENIR**
>
> L’arbre binaire associé à $[6,\ 2,\ 3,\ +,\ x,\ 94,\ 1,\ -,\ +]$ est celui représenté ci-dessous. Le symbole x désigne la multiplication.

Question 4. — instruction 1

Sujet p. 10

> Écrire le code de l’instruction 1, qui complète le code de la fonction evaluer fourni.

Une feuille est reconnue ici par le fait que sa racine n’est aucun des quatre opérateurs. Le code fourni utilise l’écriture Python ” $*$ ” pour l’opérateur de multiplication ; il faut donc utiliser cette même écriture dans le test.

> **RÉPONSE À RETENIR**
>
> Instruction 1 : `racine(arb) not in ["+", "-", "*", "/"]`

Question 4. — instruction 2

Sujet p. 10

> Écrire le code de l’instruction 2, qui complète le code de la fonction evaluer fourni.

Il faut examiner la valeur de la racine : pour une feuille, c’est la valeur à renvoyer ; pour un arbre dont la racine est un opérateur, cette valeur détermine l’opération à effectuer.

> **RÉPONSE À RETENIR**
>
> Instruction 2 : `racine(arb)`

FormaV - document d'entraînement non officiel

Question 4. — instruction 3

Sujet p. 10

> Écrire le code de l'instruction 3, qui complète le code de la fonction evaluer fourni.

Dans chacune des opérations, l'opérande gauche est la valeur obtenue en évaluant récursivement le sous-arbre gauche.

> **RÉPONSE À RETENIR**
>
> Instruction 3 : `gauche(arb)`

Question 4. — instruction 4

Sujet p. 10

> Écrire le code de l'instruction 4, qui complète le code de la fonction evaluer fourni.

L'opérande droit doit également être évalué récursivement. Il faut donc appeler `evaluer` sur le sous-arbre droit.

> **RÉPONSE À RETENIR**
>
> Instruction 4 : `evaluer(droit(arb))`

Sujet : https://www.sujetdebac.fr/annales-pdf/2023/spe-numerique-informatique-2023-liban-1-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.