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 (NSI)

Session 2023 • Centres étrangers • Jour 1

Code sujet **23-NSIJ1ME1** 3 h 30

Proposition de corrigé pédagogique • FormaV

Se repérer dans le corrigé

Exercice 1 — Bases de données et SQL
9 réponses • sujet p. 1–2

Exercice 2 — Réseaux et protocoles de routage
6 réponses • sujet p. 3–4

Exercice 3 — Files
5 réponses • sujet p. 5–6

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

Bases de données et SQL

Question 1.

Sujet p. 1, 2

> Écrire le schéma relationnel de la relation description en précisant les attributs et leurs types probables, la clé primaire et la ou les clé(s) étrangère(s) éventuelle(s).

La relation `description` comporte les attributs visibles dans l'extrait : `id_description`, `resume`, `duree` et `id_emission`. Un identifiant et une durée en minutes peuvent être stockés comme des entiers ; le résumé comme du texte. L'identifiant de description sert de clé primaire. Comme `id_emission` identifie une émission dans la relation `emission`, il peut être défini comme clé étrangère vers cette relation.

RÉPONSE À RETENIR

`description(id_description : INTEGER, resume : TEXT, duree : INTEGER, id_emission : INTEGER)`

Clé primaire : `id_description`. Clé étrangère probable : `id_emission` référence `emission(id_emission)`. Les types indiqués sont des types probables ; un type texte de longueur suffisante convient pour `resume`.

2

Question (a)

Sujet p. 1, 2

> Écrire ce qu'affiche la requête SELECT theme, annee FROM podcast WHERE id_emission = 10081 appliquée aux extraits précédents.

`WHERE` sélectionne les lignes de `podcast` dont `id_emission` vaut 10081 ; `SELECT` ne conserve ensuite que les colonnes `theme` et `annee`. Dans l'extrait, il s'agit des podcasts 1 et 2.

RÉPONSE À RETENIR

theme	annee
Le système d'enseignement supérieur français est-il juste et efficace ?	2022
Trois innovations pour la croissance future (1/3) : la révolution blockchain	2021

Question (b)

Sujet p. 1, 2

> Écrire une requête SQL permettant d'afficher les thèmes des podcasts de l'année 2019.

La condition `WHERE annee = 2019` filtre les podcasts de 2019. Sans `DISTINCT`, chaque ligne correspondante est affichée : le thème des podcasts 4 et 40 apparaît donc deux fois.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

```
SELECT theme
FROM podcast
WHERE annee = 2019;
```

Résultat sur l'extrait fourni :

theme
Le poids de la souveraineté numérique française
Le poids de la souveraineté numérique française
Dans le cloud en Islande, terre des data center

Question (c)

Sujet p. 1, 2

> Écrire une requête SQL affichant la liste des thèmes et des années de diffusion des podcasts dans l'ordre chronologique des années.

La requête sélectionne les thèmes et années, puis `ORDER BY annee` classe les lignes par année croissante, comme le rappelle l'énoncé. L'ordre relatif des lignes ayant la même année n'est pas spécifié.

RÉPONSE À RETENIR

```
SELECT theme, annee
FROM podcast
ORDER BY annee;
```

Un résultat possible sur l'extrait fourni est :

theme	annee
Le poids de la souveraineté numérique française	2019
Le poids de la souveraineté numérique française	2019
Dans le cloud en Islande, terre des data center	2019
Trois innovations pour la croissance future (1/3) : la révolution blockchain	2021
Travailleurs de plateformes : vers un nouveau prolétariat ?	2021
Le système d'enseignement supérieur français est-il juste et efficace ?	2022

3

Question (a)

Sujet p. 1, 2

> Décrire simplement le résultat obtenu avec la requête SQL SELECT DISTINCT theme FROM podcast.

`DISTINCT` élimine les doublons dans les valeurs sélectionnées. La requête affiche donc chaque thème présent dans `podcast` une seule fois, sans imposer d'ordre particulier.

RÉPONSE À RETENIR

Sur l'extrait fourni, les cinq thèmes distincts sont :

theme
Le système d'enseignement supérieur français est-il juste et efficace ?
Trois innovations pour la croissance future (1/3) : la révolution blockchain
Travailleurs de plateformes : vers un nouveau prolétariat ?
Le poids de la souveraineté numérique française
Dans le cloud en Islande, terre des data center

Question (b)

Sujet p. 1, 2

> Écrire une requête SQL supprimant de la relation podcast la ligne contenant l'id_podcast = 40.

`DELETE` supprime les lignes satisfaisant la condition indiquée. La clause `WHERE` limite ici la suppression au podcast dont l'identifiant vaut 40.

RÉPONSE À RETENIR

```
DELETE FROM podcast
WHERE id_podcast = 40;
```

4

Question (a)

Sujet p. 1, 2

> Une erreur de saisie a été faite dans la relation emission. Écrire une requête SQL permettant de changer le nom de l'animateur de l'émission Le temps du débat en Emmanuel L..

`UPDATE` modifie la table `emission`. La condition sélectionne l'émission désignée par son nom et la valeur à remplacer est celle de l'attribut `animateur`.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

```
UPDATE emission
SET animateur = 'Emmanuel L.'
WHERE nom = 'Le temps du débat';
```

Question (b)

Sujet p. 1, 2

> Écrire une requête SQL permettant d'ajouter l'émission Hashtag sur la radio France inter avec Mathieu V.. On lui donnera un id_emission égal à 12850.

La table `emission` contient les attributs `id_emission`, `nom`, `radio` et `animateur`. La requête fournit une valeur pour chacun, dans le même ordre que les attributs énumérés.

RÉPONSE À RETENIR

```
INSERT INTO emission (id_emission, nom, radio, animateur)
VALUES (12850, 'Hashtag', 'France inter', 'Mathieu V.');
```

Question 5.

Sujet p. 1, 2

> Écrire une requête permettant de lister les thèmes, le nom des émissions et le résumé des podcasts pour lesquels la durée est strictement inférieure à cinq minutes.

La requête doit réunir les thèmes de `podcast`, les noms de `emission` et les résumés de `description`, puis ne retenir que les descriptions dont la durée est strictement inférieure à cinq minutes. On relie les trois tables au moyen de `id_emission`.

La relation `description` de l'extrait ne contient pas de clé reliant une description à un identifiant précis de podcast. La jointure sur `id_emission` associe donc chaque description courte à chaque podcast de la même émission ; les lignes ci-dessous sont le résultat de cette jointure, et non une correspondance individuelle entre un résumé et un podcast.

RÉPONSE À RETENIR

```
SELECT podcast.theme, emission.nom, description.resume
FROM podcast
JOIN emission ON podcast.id_emission = emission.id_emission
JOIN description ON description.id_emission = emission.id_emission
WHERE description.duree < 5;
```

Dans l'extrait, les descriptions 101 et 103 durent chacune 4 minutes et ont l'`id_emission` 10081. Les deux podcasts associés à cette émission se combinent donc avec chacune de ces descriptions :

theme	nom	resume
Le système d'enseignement supérieur français est-il juste et efficace ?	Entendez-vous l'éco ?	Autrefois réservé à une élite, l'enseignement supérieur français s'est profondément démocratisé : donne-t-il pour autant les mêmes chances à chacun ?
Le système d'enseignement supérieur français est-il juste et efficace ?	Entendez-vous l'éco ?	La promesse de la blockchain, c'est la suppression des intermédiaires et la confiance à grande échelle
Trois innovations pour la croissance future (1/3) : la révolution blockchain	Entendez-vous l'éco ?	Autrefois réservé à une élite, l'enseignement supérieur français s'est profondément démocratisé : donne-t-il pour autant les mêmes chances à chacun ?
Trois innovations pour la croissance future (1/3) : la révolution blockchain	Entendez-vous l'éco ?	La promesse de la blockchain, c'est la suppression des intermédiaires et la confiance à grande échelle

Exercice 2

3 points

Réseaux et protocoles de routage

1

Question (a)

Sujet p. 3

> 10100100.10110010.XXXXXXXX.XXXXXXXX est la conversion en binaire de l'adresse 164.178.2.13. Terminer cette conversion en remplaçant les deux octets XXXXXXXX par leur valeur binaire.

On convertit chacun des deux derniers octets en binaire sur 8 bits. Pour le troisième octet, $2 = 2^1$, donc on écrit `00000010`. Pour le quatrième, $13 = 8 + 4 + 1$, donc on écrit `00001101`.

> **RÉPONSE À RETENIR**
>
> `10100100.10110010.00000010.00001101`

Question (b)

Sujet p. 3

> Donner, en justifiant, l'adresse du réseau à laquelle appartient la machine dont l'adresse complète en notation CIDR est 164.178.2.13/24.

Le préfixe `/24` indique que les 24 premiers bits, soit les trois premiers octets, forment la partie réseau. Il reste 8 bits pour la partie hôte. Pour obtenir l'adresse du réseau, on met ces 8 bits à zéro : le dernier octet devient 0.

> **RÉPONSE À RETENIR**
>
> L'adresse du réseau est `164.178.2.0`.

Question 2.

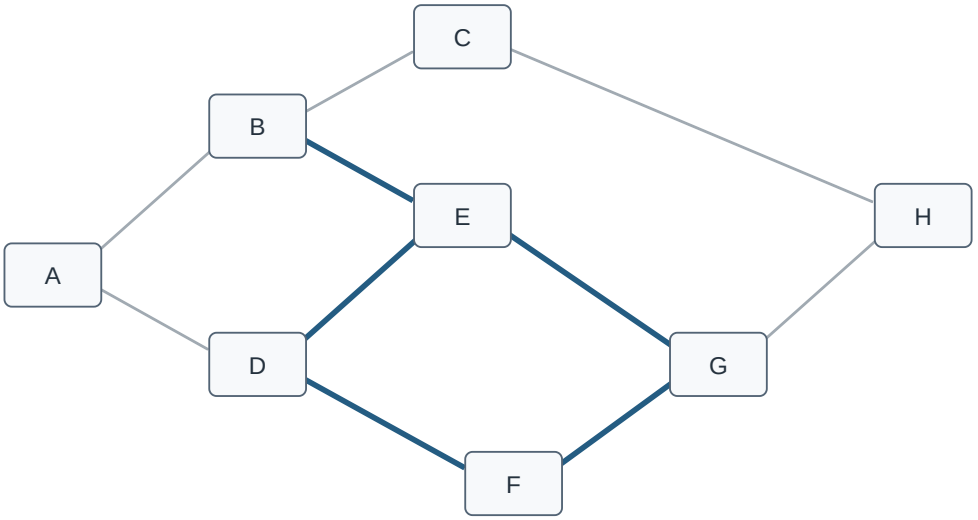
Sujet p. 3

> Donner tous les chemins de parcours optimaux pour un paquet émis par A et à destination de G en suivant le protocole RIP.

RIP minimise le nombre de liaisons traversées. Depuis A, les premiers routeurs possibles sont B et D. En trois sauts, les itinéraires qui atteignent G sont A–B–E–G, A–D–E–G et A–D–F–G. Il n'existe pas d'itinéraire en deux sauts : ni B ni D n'est relié directement à G. Ces trois itinéraires sont donc optimaux et comportent chacun trois sauts. Un chemin passant par C et H, tel que A–B–C–H–G, demande quatre sauts.

FormaV - document d'entraînement non officiel

Chemins optimaux selon RIP



Les arêtes mises en évidence sont celles des trois chemins optimaux.

> **RÉPONSE À RETENIR**
>
> Les chemins optimaux sont **A–B–E–G**, **A–D–E–G** et **A–D–F–G**, chacun en 3 sauts.

3

Question (a)

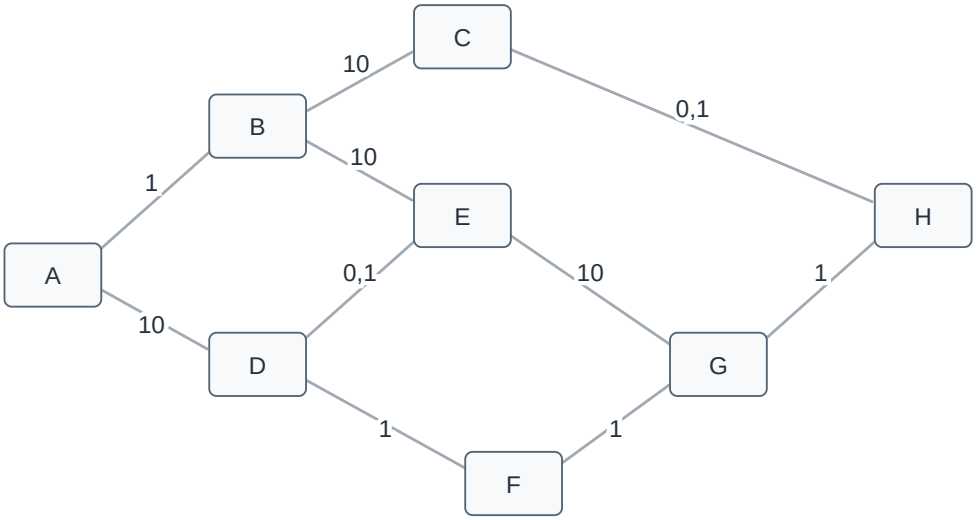
Sujet p. 4

> Dessiner sur votre copie le schéma du réseau en remplaçant le type de connexion par le coût. On se limitera aux noms des routeurs et aux coûts. Le coût d'une liaison est donné par la formule coût = 10^9/BP, où BP est la bande passante en bit/s. Les bandes passantes estimées sont de 10^8 bit/s pour Ethernet, 10^9 bit/s pour FastEthernet et 10^{10} bit/s pour Fibre.

On calcule le coût de chaque liaison avec $\text{coût} = 10^9/\text{BP}$. Ainsi, Ethernet donne $10^9/10^8 = 10$, FastEthernet donne $10^9/10^9 = 1$, et Fibre donne $10^9/10^{10} = 0{,}1$. On reporte ces valeurs sur les liaisons de la figure 2, sans modifier la topologie : A–B est FastEthernet, A–D Ethernet, B–C et B–E Ethernet, C–H Fibre, D–E Fibre, D–F FastEthernet, E–G Ethernet, F–G et G–H FastEthernet.

FormaV - document d'entraînement non officiel

Réseau de routeurs pondéré par les coûts OSPF



Chaque liaison porte le coût correspondant à son type sur la figure 2.

> **RÉPONSE À RETENIR**
>
> Le schéma pondéré obtenu est représenté ci-dessous. Les coûts sont Ethernet : 10, FastEthernet : 1 et Fibre : 0,1.

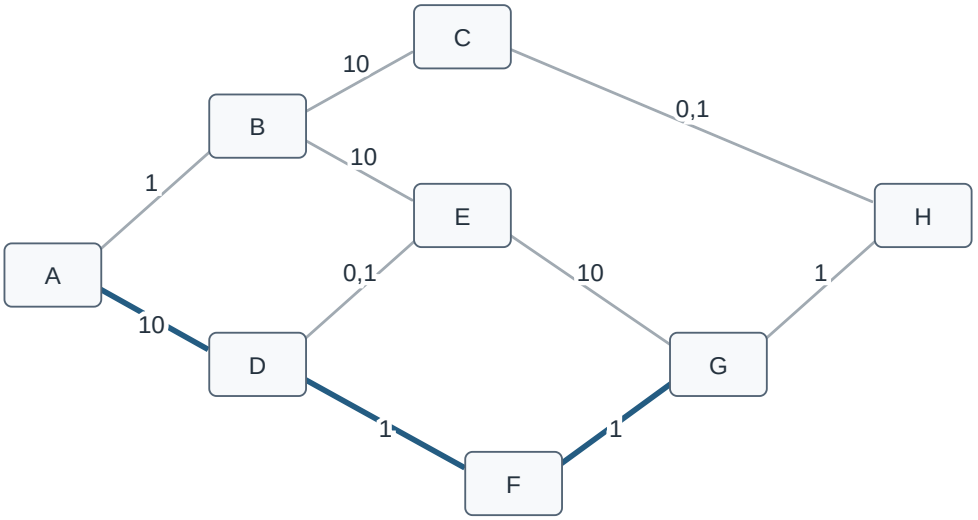
Question (b)

Sujet p. 4

> Donner le chemin de parcours pour un paquet émis par A et à destination de G en respectant le protocole OSPF.

OSPF minimise la somme des coûts des liaisons. Le chemin A–D–F–G coûte $10 + 1 + 1 = 12$. Le chemin concurrent A–B–C–H–G coûte $1 + 10 + 0{,}1 + 1 = 12{,}1$. Les itinéraires passant par E coûtent davantage : A–D–E–G coûte $10 + 0{,}1 + 10 = 20{,}1$, et A–B–E–G coûte $1 + 10 + 10 = 21$. Le coût minimal est donc 12.

Chemin OSPF le moins coûteux de A à G



Chemin mis en évidence : A–D–F–G, de coût 10 + 1 + 1 = 12.

> **RÉPONSE À RETENIR**
>
> Le chemin retenu par OSPF est **A–D–F–G**, de coût total **12**.

Question (c)

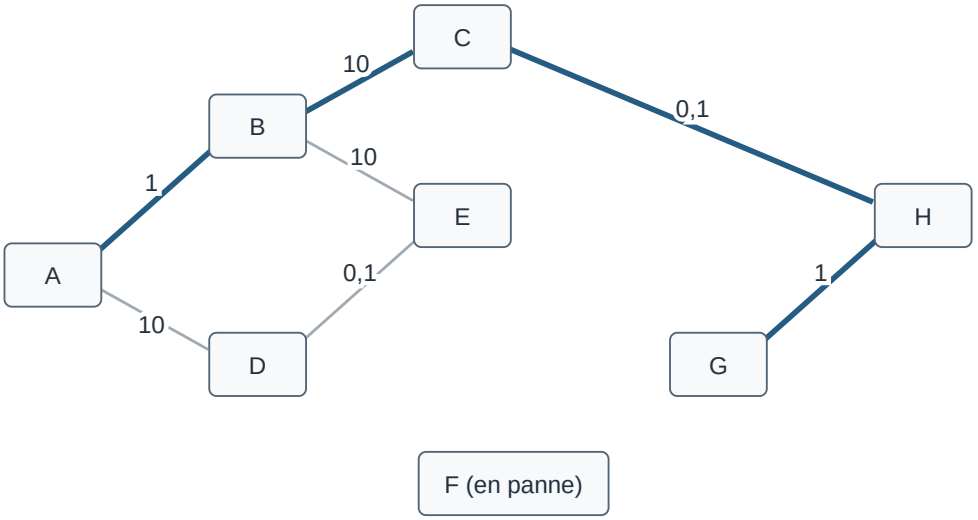
Sujet p. 4

> Donner le chemin de parcours pour un paquet émis par A et à destination de G en respectant le protocole OSPF si le routeur F est en panne.

Si F est en panne, on retire ce routeur ainsi que ses liaisons D–F et F–G. Le chemin le moins coûteux restant est A–B–C–H–G, de coût $1 + 10 + 0{,}1 + 1 = 12{,}1$. À titre de comparaison, A–D–E–G coûte $10 + 0{,}1 + 10 = 20{,}1$, et A–B–E–G coûte $1 + 10 + 10 = 21$.

FormaV - document d'entraînement non officiel

Chemin OSPF de A à G lorsque F est en panne



Les liaisons incidentes à F sont retirées. Le chemin mis en évidence est A–B–C–H–G.

> **RÉPONSE À RETENIR**
>
> Lorsque F est en panne, le chemin retenu par OSPF est **A–B–C–H–G**, de coût total **12,1**.

FormaV - document d'entraînement non officiel

Exercice 3

6 points

Files

Question 1.

Sujet p. 5

> Recopier et compléter sur votre copie les pointillés des lignes 3 et 4 de la fonction ajout(f), qui permet de tirer au hasard une couleur et de l'ajouter à une séquence. Le tuple nommé couleur contient les quatre couleurs ('bleu', 'rouge', 'jaune', 'vert'). La fonction ajout prend en paramètre la séquence f et renvoie la séquence f modifiée, qui intègre la couleur ajoutée au format chaîne de caractères.

Le tuple `couleur` contient quatre éléments, indexés de 0 à 3. Comme les deux bornes de `randint` sont incluses, on tire un indice entre 0 et 3, puis on ajoute à la file la couleur qui se trouve à cet indice.

```
def ajout(f):
    couleur = ('bleu', 'rouge', 'jaune', 'vert')
    indice = randint(0, 3)
    enfiler(f, couleur[indice])
    return f
```

Par exemple, si `randint` renvoie `2`, l'élément ajouté est `'jaune'`, troisième élément du tuple.

> **RÉPONSE À RETENIR**
>
> Ligne 3 : `randint(0, 3)`. Ligne 4 : `enfiler(f, couleur[indice])`.

Question 2.

Sujet p. 5

> Écrire la fonction vider qui prend en paramètre une séquence f et la vide sans la renvoyer.

On retire les éléments un à un tant que la file n'est pas vide. Chaque appel à `defiler` retire un élément de la tête. La boucle termine après un nombre d'itérations égal à la taille initiale de la file ; si elle est déjà vide, le corps de la boucle n'est pas exécuté.

```
def vider(f):
    while not est_vide(f):
        defiler(f)
```

Par exemple, sur une file contenant la séquence d'exemple du sujet, `[rouge, bleu, rouge, jaune, bleu]`, les cinq éléments sont retirés et la file devient vide.

> **RÉPONSE À RETENIR**
>
> Cette fonction vide la file `f` et ne renvoie aucune valeur.

FormaV - document d'entraînement non officiel

Question 3.

Sujet p. 5, 6

> Recopier et compléter sur votre copie les pointillés des lignes 4 à 10 de la fonction affiche_seq(sequence) présentée avec le nom affiche_sequence(sequence) dans le code. Elle ajoute une nouvelle couleur à sequence, puis affiche les couleurs de la séquence, une par une, avec une pause de 0,5 seconde entre chaque affichage, en utilisant time.sleep(0.5). Après son exécution, la file sequence ne devra pas être vidée de ses éléments.

Après l'ajout d'une couleur, on défile les éléments de `sequence`, on les affiche et on les place dans `stock`. Une fois la première file parcourue, on transfère les éléments de `stock` vers `sequence` : l'ordre est ainsi conservé et la séquence n'est pas perdue.

```
def affiche_sequence(sequence):
    stock = creer_file_vide()
    ajout(sequence)
    while not est_vide(sequence):
        c = defiler(sequence)
        affichage(c)
        time.sleep(0.5)
        enfiler(stock, c)
    while not est_vide(stock):
        enfiler(sequence, defiler(stock))
```

Trace avec la séquence d'exemple du sujet : si `sequence` contient `[rouge, bleu, rouge, jaune, bleu]` avant l'appel et que le tirage aléatoire ajoute `'vert'`, les couleurs affichées sont rouge, bleu, rouge, jaune, bleu, vert. Après le transfert depuis `stock`, `sequence` contient de nouveau ces six éléments dans le même ordre. Le choix de `'vert'` est un cas possible du tirage, pas une valeur imposée.

> **RÉPONSE À RETENIR**
>
> Ligne 5 : `c = defiler(sequence)` ; ligne 6 : `affichage(c)` ; ligne 8 : `enfiler(stock, c)` ; ligne 9 : `while not est_vide(stock):` ; ligne 10 : `enfiler(sequence, defiler(stock))`.

4

Question (a)

Sujet p. 5, 6

> Afin d'obtenir la fonction tour_de_jeu(sequence) correspondant au comportement décrit, recopier le script, compléter les pointillés et choisir les syntaxes correspondant aux ZONES A, B, C, D, E et F, puis les y remplacer. Il faut choisir six propositions parmi les suivantes : vider(sequence) ; defiler(sequence) ; enfiler(sequence, c_joueur) ; enfiler(stock, c_seq) ; enfiler(sequence, defiler(stock)) ; enfiler(stock, defiler(sequence)) ; affiche_seq(sequence) ; while not est_vide(sequence): ; while not est_vide(stock): ; if not est_vide(sequence): ; if not est_vide(stock):. Le jeu commence par ajouter une couleur à la séquence et en afficher l'intégralité. Le joueur la reproduit dans le même ordre en choisissant une couleur via saisie_joueur(). On vérifie si cette couleur est conforme à celle de la séquence : si c'est la bonne couleur, on poursuit ; sinon, on vide sequence. Si le joueur arrive au bout de la séquence, il valide le tour et le jeu se poursuit avec un nouveau tour ; sinon, il a perdu et le jeu s'arrête. La fonction tour_de_jeu s'arrête lorsque le joueur a trouvé toutes les bonnes couleurs de sequence dans l'ordre ou dès qu'il se trompe. Après son exécution, sequence ne devra pas être vidée en cas de victoire.

La zone A affiche la séquence après y avoir ajouté une couleur. Pour chaque réponse, la couleur attendue est retirée de `sequence`. Si la réponse est correcte, elle est conservée dans `stock`, qui servira à reconstituer la séquence en cas de victoire. Si la réponse est fausse, il faut vider à la fois `sequence` et `stock` : vider

seulement `sequence` laisserait le préfixe correctement deviné dans `stock`, qui serait ensuite réintroduit par le transfert final.

Les zones sont donc : A `affiche_seq(sequence)` ; B `defiler(sequence)` ; C `enfiler(stock, c_seq)` ; D `vider(sequence)` ; E `while not est_vide(stock):` ; F `enfiler(sequence, defiler(stock))`. L'instruction supplémentaire `vider(stock)` dans le cas d'une erreur est nécessaire pour que la séquence soit effectivement vide après l'erreur.

Le sujet appelle la fonction d'affichage `affiche_sequence` dans le code de la question 3, mais la nomme `affiche_seq` dans les choix de la question 4. Pour que le script soit exécutable avec la définition précédente, on ajoute l'alias suivant.

```
affiche_seq = affiche_sequence

def tour_de_jeu(sequence):
    affiche_seq(sequence)
    stock = creer_file_vide()
    while not est_vide(sequence):
        c_joueur = saisie_joueur()
        c_seq = defiler(sequence)
        if c_joueur == c_seq:
            enfiler(stock, c_seq)
        else:
            vider(sequence)
            vider(stock)
    while not est_vide(stock):
        enfiler(sequence, defiler(stock))
```

Trace illustrative : si la séquence est vide au début et que le tirage ajoute `'rouge'`, une réponse `'rouge'` place cette couleur dans `stock`, puis le transfert final la remet dans `sequence` : la victoire conserve la séquence. Si, dans le même cas, la réponse est `'bleu'`, les deux files sont vidées ; le transfert final ne déplace rien et la séquence reste vide.

> **RÉPONSE À RETENIR**
>
> Les zones A à F sont celles indiquées ci-dessus. En cas d'erreur, l'ajout de `vider(stock)` au bloc `else` garantit que la séquence reste vide ; en cas de victoire, le transfert final restitue tous ses éléments dans leur ordre initial.

Question (b)

Sujet p. 5, 6

> Proposer une modification pour que la fonction se répète si le joueur trouve toutes les couleurs de la séquence (dans ce cas, une nouvelle couleur est ajoutée) ou s'il se trompe (dans ce cas, la séquence est vidée et se voit ajouter une nouvelle couleur). On pourra ajouter des instructions qui ne sont pas proposées dans la question 4.(a).

Il suffit de rappeler `tour_de_jeu` dans une boucle. Après une victoire, la séquence a été restaurée ; le prochain appel ajoute une couleur à cette séquence. Après une erreur, elle a été vidée ; le prochain appel y ajoute une nouvelle couleur et une nouvelle partie recommence.

```
sequence = creer_file_vide()
while True:
    tour_de_jeu(sequence)
```

Par exemple, si une partie commence avec une séquence vide et que le premier tirage ajoute `'rouge'`, une victoire conserve `[rouge]` et le tour suivant y ajoute une couleur. Une erreur laisse la file vide et l'appel suivant démarre avec une seule nouvelle couleur.

> **RÉPONSE À RETENIR**
>
> La boucle relance un tour après chaque victoire ou erreur. Le comportement demandé repose sur la fonction de la question 4.(a) : conservation de la séquence en cas de victoire, vidage en cas d'erreur.

Sujet : https://aeif.fr/wp-content/uploads/2025/07/Centres_Etrangers_mars_2023_1.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.