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 • Centres étrangers • Jour 1

Code sujet **23-NSIJ1G11** 3 heures 30

Proposition de corrigé pédagogique • FormaV

Se repérer dans le corrigé

Exercice 1 — Bases de données relationnelles et langage SQL
9 réponses • sujet p. 2–3

Exercice 2 — Réseaux et protocoles de routage
6 réponses • sujet p. 4–5

Exercice 3 — Structures de files — jeu Simon
5 réponses • sujet p. 6–9

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 relationnelles et langage SQL

Question 1.

Sujet p. 2, 3

> Écrire le schéma relationnel de la relation description, en précisant les attributs et leurs types probables, la clé primaire et la ou les clés étrangères éventuelles.

La relation `description` comporte les quatre attributs présentés dans le sujet. L'identifiant distingue chaque description ; le résumé est une chaîne de caractères et la durée, exprimée en minutes, un nombre entier. L'attribut `id_emission` désigne une émission. On peut donc proposer les types probables suivants :

```
description(
    id_description : INTEGER,
    resume : TEXT,
    duree : INTEGER,
    id_emission : INTEGER
)
Clé primaire : id_description
Clé étrangère : id_emission REFERENCES emission(id_emission)
```

Le sujet n'énonce pas explicitement les contraintes de la relation `description` ; on retient ici l'hypothèse naturelle que `id_description` est sa clé primaire et que `id_emission` référence la clé primaire de `emission`. Un type de chaîne comme `VARCHAR` pourrait remplacer `TEXT`.

> **RÉPONSE À RETENIR**
>
> `description(id_description INTEGER, resume TEXT, duree INTEGER, id_emission INTEGER)`, avec `id_description` comme clé primaire et `id_emission` comme clé étrangère vers `emission(id_emission)`.

a

Question 2.a.

Sujet p. 2, 3

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

La requête sélectionne les colonnes `theme` et `annee` pour les lignes de `podcast` dont `id_emission` vaut 10081. Dans l'extrait, ce sont les podcasts d'identifiants 1 et 2. L'identifiant du podcast n'est pas affiché, car il ne figure pas parmi les colonnes sélectionnées.

> **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

b

Question 2.b.

Sujet p. 2, 3

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

On filtre les lignes sur l'année 2019, puis on sélectionne uniquement leur thème. Comme la requête ne comporte pas `DISTINCT`, les thèmes répétés restent présents dans le résultat.

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

RÉPONSE À RETENIR

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

c

Question 2.c.

Sujet p. 2, 3

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

On sélectionne le thème et l'année, puis `ORDER BY annee` classe les lignes par année croissante. L'ordre des lignes ayant la même année n'est pas imposé par cette requête.

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

RÉPONSE À RETENIR

Un affichage possible sur les extraits fournis 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

a

Question 3.a.

Sujet p. 2, 3

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

`DISTINCT` élimine les doublons : la requête renvoie une ligne par thème différent présent dans `podcast`. Elle ne précise pas d'ordre de tri.

RÉPONSE À RETENIR

Sur les extraits fournis, les cinq thèmes distincts sont : « 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 » et « Dans le cloud en Islande, terre des data center ».

b

Question 3.b.

Sujet p. 2, 3

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

La commande `DELETE` supprime la ligne de `podcast` identifiée par `id_podcast = 40`.

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

RÉPONSE À RETENIR

La ligne dont `id_podcast` vaut 40 est supprimée de `podcast`.

a

Question 4.a.

Sujet p. 2, 3

> 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. ».

La commande `UPDATE` modifie l'attribut `animateur` de la ligne identifiée par le nom de l'émission « Le Temps du débat ».

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

RÉPONSE À RETENIR

L'animateur de l'émission « Le Temps du débat » est remplacé par « Emmanuel L. ».

b

Question 4.b.

Sujet p. 2, 3

> 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.

On insère une ligne dans `emission` en indiquant les colonnes et leurs valeurs dans le même ordre.

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

RÉPONSE À RETENIR

La relation `emission` reçoit une ligne pour l'émission « Hashtag », diffusée sur « France inter », animée par « Mathieu V. », et dont `id_emission` vaut 12850.

Question 5.

Sujet p. 2, 3

> 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 5 minutes.

La durée se trouve dans `description`, le thème dans `podcast` et le nom de l'émission dans `emission`. On relie ces tables par leur attribut commun `id_emission`, puis on filtre les descriptions dont la durée est strictement inférieure à 5 minutes.

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

La relation `description` ne comporte pas d'attribut `id_podcast`. La jointure sur `id_emission` associe donc chaque description aux thèmes de tous les podcasts de la même émission ; elle ne permet pas d'identifier un thème unique comme étant celui du résumé. Le résultat ci-dessous correspond aux lignes produites par cette jointure sur les extraits fournis. Leur ordre n'est pas garanti, car la requête ne comporte pas de `ORDER BY`.

RÉPONSE À RETENIR

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 ?
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 ?
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 ?	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

a

Question 1.a.

Sujet p. 4

> 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 séparément chaque octet de l'adresse décimale en binaire sur 8 bits. Le troisième octet vaut 2, soit 00000010. Le quatrième vaut 13 = 8 + 4 + 1, soit 00001101.

> **RÉPONSE À RETENIR**
>
> 164.178.2.13 s'écrit en binaire : 10100100.10110010.00000010.00001101.

b

Question 1.b.

Sujet p. 4

> 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 signifie que les 24 premiers bits, soit les trois premiers octets, constituent la partie réseau. Les 8 bits restants constituent la partie hôte. Pour obtenir l'adresse du réseau, on met ces bits de la partie hôte à 0.

> **RÉPONSE À RETENIR**
>
> L'adresse du réseau est 164.178.2.0. En effet, la partie réseau de 164.178.2.13/24 est 164.178.2 et la partie hôte est 13 ; l'adresse du réseau s'obtient en remplaçant cette dernière par 0.

Question 2.

Sujet p. 4

> Donner tous les chemins de parcours optimaux pour un paquet émis par A et à destination de G en suivant le protocole RIP, qui cherche à minimiser le nombre de routeurs traversés.

Avec RIP, le coût d'un chemin est son nombre de sauts, c'est-à-dire le nombre de liaisons parcourues. En lisant la topologie de la figure 1, on trouve trois chemins de trois sauts entre A et G. Le trajet A–B–C–H–G, par exemple, comporte quatre sauts et n'est donc pas optimal.

> **RÉPONSE À RETENIR**
>
> Les chemins optimaux sont :
>
> - A → B → E → G ;
> - A → D → E → G ;
> - A → D → F → G.
>
> Chacun comporte trois sauts.

FormaV - document d'entraînement non officiel

a

Question 3.a.

Sujet p. 5

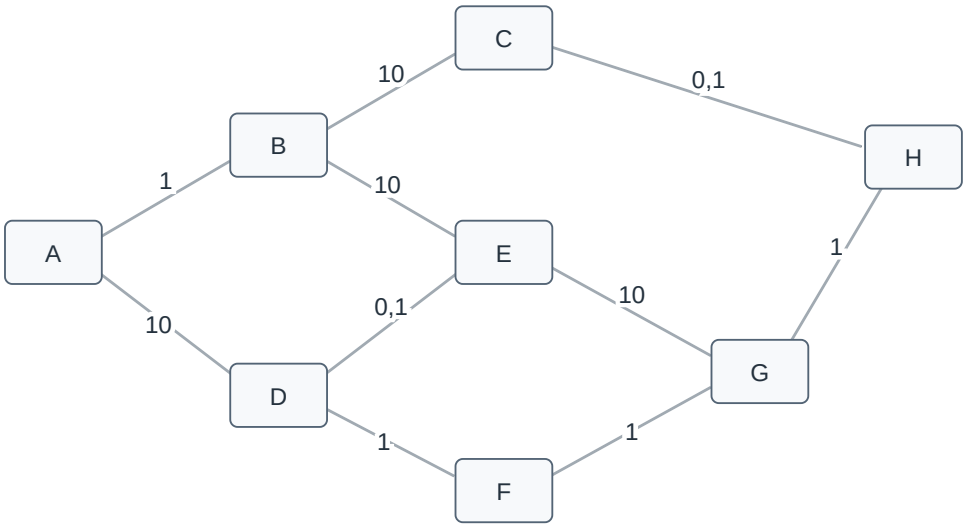
> Dessiner sur votre copie le schéma du réseau en remplaçant le type de connexion par le coût. Se limiter aux noms des routeurs et aux coûts. Le coût d'une liaison est donné par la formule coût = 10^9 / BP, avec BP la bande passante en bit/s ; les types de connexion et leurs bandes passantes estimées sont indiqués dans le sujet.

On applique la formule coût = 10^9 / BP à chaque type de liaison indiqué sur la figure 2 :

- Ethernet : $10^9 / 10^8 = 10$;
- Fast-Ethernet : $10^9 / 10^9 = 1$;
- Fibre : $10^9 / 10^{10} = 0{,}1$.

On conserve les liaisons du réseau et on remplace chacune par le coût correspondant à son type.

Réseau avec les coûts des liaisons



Chaque liaison conserve son type dans la figure 2 : Ethernet coûte 10, Fast-Ethernet coûte 1 et Fibre coûte 0,1.

RÉPONSE À RETENIR

Les coûts sont : A–B : 1 ; A–D : 10 ; B–C : 10 ; B–E : 10 ; C–H : 0,1 ; D–E : 0,1 ; D–F : 1 ; E–G : 10 ; F–G : 1 ; G–H : 1.

b

Question 3.b.

Sujet p. 5

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

OSPF choisit un chemin dont la somme des coûts est minimale. À partir des coûts déterminés en 3.a, on compare les principaux itinéraires vers G :

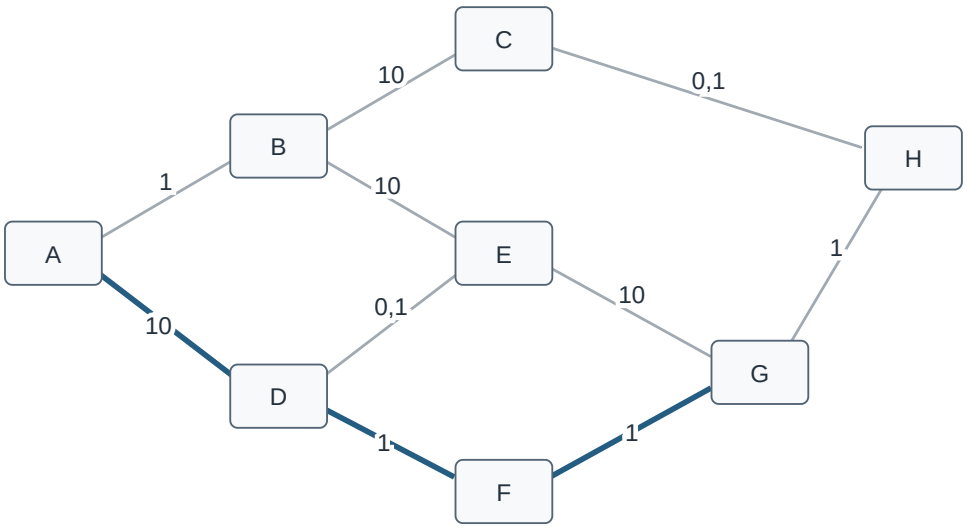
- A → D → F → G : 10 + 1 + 1 = 12 ;

FormaV - document d'entraînement non officiel

- A → B → C → H → G : 1 + 10 + 0,1 + 1 = 12,1 ;
- A → D → E → G : 10 + 0,1 + 10 = 20,1 ;
- A → B → E → G : 1 + 10 + 10 = 21.

Les autres détours ajoutent des liaisons de coût positif et ne permettent pas de diminuer ces coûts. Le plus petit coût est donc 12.

Chemin OSPF optimal de A à G



Les liaisons mises en évidence forment le chemin minimal A–D–F–G, de coût 12.

> **RÉPONSE À RETENIR**
>
> Le chemin OSPF de A à G est A → D → F → G, de coût total 12.

c

Question 3.c.

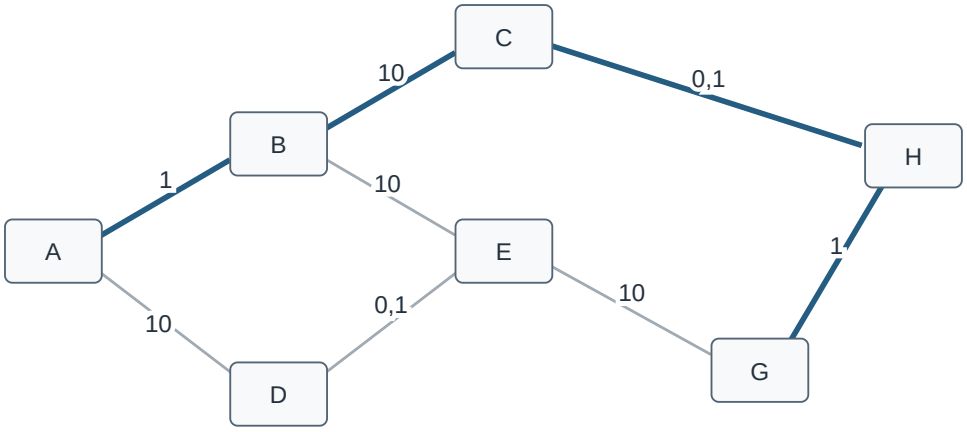
Sujet p. 5

> 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.

Lorsque F est en panne, on retire ce routeur et ses liaisons D–F et F–G. Le chemin A → B → C → H → G a pour coût 1 + 10 + 0,1 + 1 = 12,1. Les autres chemins pertinents ont un coût supérieur : A → D → E → G coûte 10 + 0,1 + 10 = 20,1, et A → B → E → G coûte 1 + 10 + 10 = 21. Le chemin passant par C et H est donc le moins coûteux.

FormaV - document d'entraînement non officiel

Chemin OSPF optimal de A à G lorsque F est en panne



F et les deux liaisons qui lui sont incidentes sont retirés ; les liaisons mises en évidence forment A–B–C–H–G.

> **RÉPONSE À RETENIR**
>
> Si F est en panne, le chemin OSPF de A à G est A → B → C → H → G, de coût total 12,1.

FormaV - document d'entraînement non officiel

Exercice 3

6 points

Structures de files — jeu Simon

Question 1.

Sujet p. 6

> Recopier et compléter les « ... » 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. La fonction prend en paramètre la séquence f et renvoie la séquence f modifiée, qui intègre la couleur ajoutée au format chaîne de caractères.

Le tuple `couleurs` contient quatre éléments, indexés de 0 à 3. On tire donc un indice avec `randint(0, 3)`, puis on ajoute à la file la couleur située à cet indice. Enfin, la fonction renvoie la file modifiée.

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

Par exemple, si le tirage donne 2, la couleur ajoutée est `"jaune"`, puisque `couleurs[2]` vaut `"jaune"`.

RÉPONSE À RETENIR

Ligne 3 : `randint(0, 3)`. Ligne 4 : `enfiler(f, couleurs[indice])`.

Question 2.

Sujet p. 7

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

Pour vider la file sans renvoyer de valeur, on retire ses éléments en tête tant qu'elle n'est pas vide. À chaque itération, le nombre d'éléments diminue de un : la boucle termine donc après un nombre fini d'itérations.

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

Par exemple, si `f` contient `["rouge", "bleu"]`, les deux appels successifs à `defiler(f)` retirent « rouge », puis « bleu » ; la file est alors vide.

RÉPONSE À RETENIR

La fonction vide `f` en place et ne renvoie rien.

Question 3.

Sujet p. 7

> Recopier et compléter les « ... » des lignes 4 à 10 de la fonction affich_seq(sequence). La fonction ajoute d'abord une nouvelle couleur à sequence, puis affiche les couleurs de la séquence une par une, avec une pause de 0,5 s entre chaque affichage. Utiliser affichage(couleur) pour afficher la couleur et time.sleep(0.5) pour la temporisation. Après l'exécution de la fonction, la file sequence ne doit pas être vidée de ses éléments.

La file `sequence` est parcourue en retirant chaque couleur de sa tête. Chaque couleur est affichée, puis placée dans une file auxiliaire `stock`. Le transfert final de `stock` vers `sequence` rétablit l'ordre initial et conserve tous les éléments, y compris la couleur ajoutée au début.

FormaV - document d'entraînement non officiel

```
def affich_seq(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 possible avec l'exemple du sujet : si `sequence` contient initialement rouge, bleu, rouge, jaune, bleu et que la couleur tirée est vert, les couleurs affichées sont rouge, bleu, rouge, jaune, bleu, vert. `stock` reçoit ces six couleurs dans cet ordre ; le transfert final reconstitue la file avec les six couleurs dans le même ordre.

> **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))`.

a

Question 4.a.

Sujet p. 8, 9

> Afin d'obtenir la fonction tour_de_jeu(sequence) correspondant au comportement décrit, recopier le script et compléter le « ... ». Choisir parmi les propositions de syntaxes celles qui correspondent aux ZONES A, B, C, D, E et F figurant dans le script et les y remplacer — il faut en choisir six parmi les onze : vider(sequence) ; defiler(sequence) ; enfiler(sequence,c_joueur) ; enfiler(stock,c_seq) ; enfiler(sequence, defiler(stock)) ; enfiler(stock, defiler(sequence)) ; affich_seq(sequence) ; while not est_vide(sequence): ; while not est_vide(stock): ; if not est_vide(sequence): ; if not est_vide(stock):. La fonction doit gérer un tour de jeu : ajouter une couleur à la séquence et afficher son intégralité, permettre au joueur de la reproduire dans le même ordre avec saisie_joueur(), vérifier chaque couleur et poursuivre si elle est correcte, sinon vider sequence. Elle s'arrête si le joueur trouve toutes les bonnes couleurs dans l'ordre ou dès qu'il se trompe ; après son exécution, sequence ne doit pas être vidée en cas de victoire.

Pour parcourir les réponses, on retire chaque couleur attendue de `sequence`. Si la réponse est correcte, on conserve cette couleur dans `stock`. En cas de victoire, le transfert final de `stock` vers `sequence` restaure la séquence dans son ordre initial.

Les six zones proposées correspondent à : A = `affich_seq(sequence)` ; B = `defiler(sequence)` ; C = `enfiler(stock, c_seq)` ; D = `vider(sequence)` ; E = `while not est_vide(stock):` ; F = `enfiler(sequence, defiler(stock))`. La comparaison à compléter est ==.

Il existe toutefois une incohérence dans les choix proposés : avec ces six zones seulement, si le joueur se trompe après une ou plusieurs réponses correctes, `stock` contient le préfixe déjà trouvé. La zone E le retransfère alors dans `sequence`, qui n'est donc pas vide après l'erreur. Pour respecter aussi la consigne qui demande de vider `sequence` en cas d'erreur, il faut ajouter `vider(stock)` dans la branche `else`. Le script fonctionnel est donc :

```
def tour_de_jeu(sequence):
    affich_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 possible : pour la séquence affichée bleu, rouge, vert, si le joueur répond bleu, rouge, vert, ces couleurs sont transférées dans `stock`, puis remises dans `sequence` dans le même ordre. S'il répond bleu puis jaune, le bleu est dans `stock` au moment de l'erreur ; les deux files sont vidées et le transfert final ne réintroduit aucun élément.

> **RÉPONSE À RETENIR**
>
> La complétion attendue des zones est A = `affich_seq(sequence)`, B = `defiler(sequence)`, C = `enfiler(stock, c_seq)`, D = `vider(sequence)`, E = `while not est_vide(stock):`, F = `enfiler(sequence, defiler(stock))`, avec la comparaison `==`. Mais ces six choix seuls ne vident pas la séquence après une erreur précédée d'une bonne réponse : il faut également vider `stock` dans la branche `else`, comme dans le code corrigé ci-dessus.

b

Question 4.b.

Sujet p. 9

> 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 une nouvelle couleur est ajoutée. On pourra ajouter des instructions qui ne sont pas proposées dans la question a.

Pour répéter le jeu, on place le déroulement d'un tour dans une boucle. Au début de chaque tour, `affich_seq(sequence)` ajoute une couleur et affiche la séquence. Après une victoire, la séquence est restaurée et le tour suivant ajoute une nouvelle couleur. Après une erreur, on vide les deux files : le tour suivant repart donc d'une séquence vide, à laquelle `affich_seq` ajoute une nouvelle couleur.

```
def tour_de_jeu(sequence):
    while True:
        affich_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 possible : si la première couleur tirée est bleu et que le joueur répond correctement, bleu est remis dans `sequence` à la fin du tour ; au tour suivant, une nouvelle couleur est ajoutée. Si, lors d'un tour, le joueur se trompe après une réponse correcte, `sequence` et `stock` sont vidées ; l'itération suivante ajoute une nouvelle couleur à la file vide.

> **RÉPONSE À RETENIR**
>
> On répète le déroulement avec `while True` et on appelle `affich_seq(sequence)` à chaque tour. La victoire conserve la séquence avant l'ajout suivant ; l'erreur vide la séquence et le stockage auxiliaire, puis un nouveau tour ajoute une couleur à la file vide.

Sujet : https://www.sujetdebac.fr/annales-pdf/2023/spe-numerique-informatique-2023-centres-etranger-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.