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 • Métropole • Jour 2 • Remplacement

Code sujet **22-NSIJ2ME3** 3 heures 30

Proposition de corrigé pédagogique • FormaV

Se repérer dans le corrigé

Exercice 1 — Files et programmation de base en Python
5 réponses • sujet p. 2–3

Exercice 2 — Arbres binaires de recherche
7 réponses • sujet p. 4–5

Exercice 3 — Bases de données relationnelles et langage SQL
6 réponses • sujet p. 6–13

Exercice 4 — Architecture matérielle, gestion de processus et réseaux
9 réponses • sujet p. 8–9

Exercice 5 — Programmation de base en Python et récursivité
5 réponses • sujet p. 10–12

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

Files et programmation de base en Python

Partie A

Question 1.

Sujet p. 2

> On suppose que la file f est dans l'état de la figure 1. Quel est l'état de f après l'exécution des instructions qui, trois fois, retirent l'élément en tête puis le réenfilent ?

La tête de la file est à gauche : la file initiale est `7, 8, 0, 5, 4`. À chaque tour, l'élément en tête est retiré puis ajouté en queue. Les trois premiers éléments passent donc successivement en fin de file.

1. Après le déplacement de `7` : `8, 0, 5, 4, 7`.
2. Après le déplacement de `8` : `0, 5, 4, 7, 8`.
3. Après le déplacement de `0` : `5, 4, 7, 8, 0`.

RÉPONSE À RETENIR

De la tête à la queue, la file `f` contient **5, 4, 7, 8, 0**.

Question 2.(a)

Sujet p. 2, 3

> On suppose que la file f est dans l'état de la figure 1. Après l'exécution des instructions fournies, quel est l'état de la file f ?

La première boucle retire de `f` chaque élément et l'ajoute à `g`. Elle vide donc `f` et place dans `g`, dans le même ordre, les éléments `7, 8, 0, 5, 4`. La deuxième boucle les retire de `g` et les réenfile dans `f` : leur ordre est conservé.

RÉPONSE À RETENIR

De la tête à la queue, la file `f` contient **7, 8, 0, 5, 4**, comme au départ.

Question 2.(b)

Sujet p. 2, 3

> Après l'exécution des instructions fournies, quelle est la valeur de k ? Que représente la variable k ?

La variable `k` est initialisée à `0`, puis augmentée de `1` chaque fois qu'un élément est retiré de `f` dans la première boucle. Cette boucle traite tous les éléments de la file initiale, qui en contient cinq.

RÉPONSE À RETENIR

`k` vaut 5. Elle représente le nombre d'éléments initialement présents dans `f`, donc le nombre d'éléments transférés dans `g`.

FormaV - document d'entraînement non officiel

Partie B

Question 3.

Sujet p. 3

> Recopier et compléter le code de la fonction supprimer, qui a pour paramètres une file f et un élément e présent une seule fois dans f, retire cet élément de la file et ne renvoie rien. Compléter les lignes indiquées.

Pour supprimer `e`, on transfère dans la file auxiliaire `g` tous les éléments extraits de `f`, sauf celui qui est égal à `e`. Comme l'énoncé précise que `e` est présent une seule fois, il n'est pas transféré. On remet ensuite dans `f` les éléments conservés dans `g` : ils gardent leur ordre initial.

Dans les lignes à compléter, on teste donc si `x` est différent de `e`, on ajoute alors `x` à `g`, puis, dans la seconde boucle, on retire un élément de `g` avant de le réenfiler dans `f`.

RÉPONSE À RETENIR

Fonction complétée :

```
def supprimer(f, e) :
    g = creer_file()
    while not est_vide(f) :
        x = defiler(f)
        if x != e :
            enfiler(g, x)
    while not est_vide(g) :
        x = defiler(g)
        enfiler(f, x)
```

La fonction ne renvoie rien ; après son exécution, `e` a été retiré de `f`.

Question 4.

Sujet p. 3

> En modifiant la fonction supprimer, écrire une fonction placer_en_priorite qui a pour paramètres une file f et un élément e et qui déplace cet élément en tête de la file d'attente d'impression. On suppose que e est déjà présent dans la file.

On vide d'abord `f` en transférant dans `g` tous les éléments sauf `e`. L'énoncé garantit que `e` est présent ; une fois `f` vidée, on peut donc enfiler `e` dans cette file vide : il en devient la tête. Enfin, les éléments de `g` sont transférés à la suite, en conservant leur ordre relatif.

RÉPONSE À RETENIR

Une fonction possible est :

```
def placer_en_priorite(f, e) :
    g = creer_file()
    while not est_vide(f) :
        x = defiler(f)
        if x != e :
            enfiler(g, x)
    enfiler(f, e)
    while not est_vide(g) :
        x = defiler(g)
        enfiler(f, x)
```

Après l'appel, `e` est en tête de `f`, suivi des autres éléments dans leur ordre initial.

Exercice 2

4 points

Arbres binaires de recherche

Question 1.(a)

Sujet p. 4

> On parcourt en profondeur l'arbre binaire de recherche de la figure 1. Quel est le parcours suffixe (ou postfixe) de cet arbre ?

Le parcours suffixe, ou postfixe, visite récursivement le sous-arbre gauche, puis le sous-arbre droit, et enfin la racine. Dans le sous-arbre de racine 5, on visite donc 4, puis 7, puis 5. Dans celui de racine 20, on visite d'abord son fils droit 30, puis 20. On termine par la racine 10.

> **RÉPONSE À RETENIR**
>
> Le parcours suffixe est : **4, 7, 5, 30, 20, 10**.

Question 1.(b)

Sujet p. 4

> Quel parcours permet d'obtenir les clés des nœuds de l'arbre dans l'ordre croissant, c'est-à-dire 4, 5, 7, 10, 20, 30 ?

Dans un arbre binaire de recherche, le parcours infixe visite le sous-arbre gauche, puis la racine, puis le sous-arbre droit. Comme les clés du sous-arbre gauche sont inférieures à la racine et celles du sous-arbre droit lui sont supérieures, ce parcours énumère les clés dans l'ordre croissant.

> **RÉPONSE À RETENIR**
>
> Le parcours recherché est le **parcours infixe** ; ici, il donne 4, 5, 7, 10, 20, 30.

Question 2.(a)

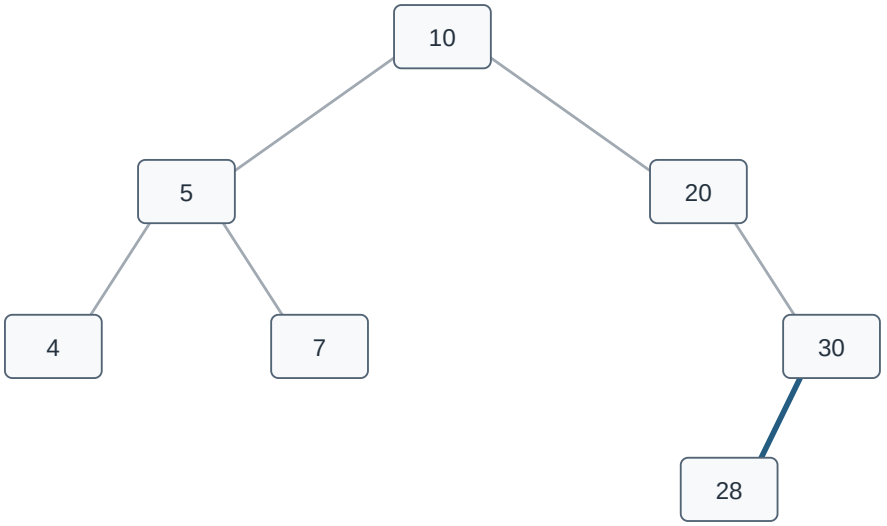
Sujet p. 4

> Recopier l'arbre binaire de recherche de la figure 1 après l'insertion de la clé 28.

Pour insérer 28, on suit les comparaisons dans l'ABR : 28 est supérieur à 10, donc on va à droite ; 28 est supérieur à 20, donc on va encore à droite ; 28 est inférieur à 30, donc il est inséré comme fils gauche de 30.

FormaV - document d'entraînement non officiel

ABR après insertion de 28



La clé 28 est le fils gauche de 30.

> **RÉPONSE À RETENIR**
>
> L'arbre obtenu après insertion est représenté ci-dessous.

Question 2.(b)

Sujet p. 4

> Quelle est la hauteur de l'arbre obtenu après l'insertion de la clé 28 ? On rappelle qu'un arbre ne comportant qu'un seul nœud a une hauteur égale à 1.

La hauteur est le nombre de nœuds sur le plus long chemin entre la racine et une feuille. Après l'insertion, le chemin le plus long est 10 → 20 → 30 → 28 : il comporte quatre nœuds. Avec la convention donnée, la hauteur de l'arbre est donc 4.

> **RÉPONSE À RETENIR**
>
> **Hauteur : 4.**

Question 3.(a)

Sujet p. 4, 5

> Quelle valeur est renvoyée lors de l'appel de la fonction mystere avec l'arbre de la figure 1 comme argument ?

La fonction suit à chaque appel le sous-arbre droit tant qu'il n'est pas vide. Dans l'arbre de la figure, elle part de 10, passe à 20, puis à 30. Le sous-arbre droit de 30 est vide : la fonction renvoie alors la racine courante, soit 30.

> **RÉPONSE À RETENIR**
>
> `mystere(a)` renvoie **30**.

Question 3.(b)

Sujet p. 4, 5

> Que représente cette valeur ?

Dans un ABR, les clés situées à droite d'un nœud sont supérieures à sa clé. En suivant les fils droits jusqu'au dernier nœud, on atteint donc la plus grande clé de l'arbre.

RÉPONSE À RETENIR

La valeur renvoyée représente la **clé maximale de l'arbre**.

Question 4.

Sujet p. 5

> Recopier et compléter la fonction rechercher, qui a pour paramètres un ABR a et une clé e, et qui renvoie True si la clé e est dans l'arbre et False sinon.

Si l'arbre est vide, la clé ne s'y trouve pas. Sinon, on compare la clé cherchée à la racine : en cas d'égalité, elle est présente ; si elle est plus petite, il suffit de poursuivre dans le sous-arbre gauche ; sinon, on poursuit dans le sous-arbre droit. Cette démarche s'appuie sur la propriété d'ordre de l'ABR.

RÉPONSE À RETENIR

```
def rechercher(a, e) :
    if est_vide(a) :
        return False
    elif e == racine(a) :
        return True
    elif e < racine(a) :
        return rechercher(sous_arbre_gauche(a), e)
    else :
        return rechercher(sous_arbre_droit(a), e)
```


FormaV - document d'entraînement non officiel

Exercice 3

4 points

Bases de données relationnelles et langage SQL

Question 1.

Sujet p. 6, 13

> C'est la pleine saison des carottes. Le primeur décide de modifier leur prix : elles passent de 1,95 € à 1,80 €. Écrire la requête en langage SQL permettant de modifier le prix de ce produit dont l'attribut idProduit vaut 1.

On modifie la colonne `prix` dans la table `produit`. La condition sur la clé primaire `idProduit` cible le produit d'identifiant 1.

RÉPONSE À RETENIR

```
UPDATE produit
SET prix = 1.80
WHERE idProduit = 1;
```

Le prix du produit d'identifiant 1 passe ainsi de 1,95 € à 1,80 €.

Question 2.

Sujet p. 6, 7, 13

> Le primeur veut ajouter le citron caviar à la base de données et exécute la requête INSERT INTO produit VALUES (112, "citron caviar", "Z", 15, 3, 6);. Que se passe-t-il ? Justifier la réponse.

Les valeurs fournies correspondent, dans l'ordre des colonnes de `produit`, à `idProduit`, `nomProduit`, `idC`, `prix`, `stock` et `idP`. La valeur `Z` pour `idC` doit donc correspondre à une catégorie existante. Or l'annexe donne uniquement les identifiants de catégorie A, B et C.

RÉPONSE À RETENIR

L'insertion est refusée car la valeur `Z` ne correspond à aucune valeur de `idCategorie` dans la table `categorie` : la contrainte de clé étrangère n'est pas respectée. L'identifiant producteur 6 existe dans l'annexe ; c'est l'identifiant de catégorie qui bloque l'insertion.

Question 3.(a)

Sujet p. 6, 7, 13

> Écrire une requête qui affiche tous les produits dont le stock est inférieur à 5 kg.

On sélectionne les lignes de `produit` pour lesquelles le stock, exprimé en kilogrammes, est strictement inférieur à 5.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

```
SELECT *
FROM produit
WHERE stock < 5;
```

Parmi les lignes visibles dans l'annexe, le résultat est :

idProduit	nomProduit	idC	prix	stock	idP
4	pomme	A	1,55	4	3
6	tomme de brebis	C	31,50	3	1

L'annexe omet des enregistrements : ce tableau ne présente donc que les résultats parmi les lignes affichées et ne permet pas de connaître toutes les lignes éventuellement sélectionnées dans la base.

Question 3.(b)

Sujet p. 6, 7, 13

> Que doit-on ajouter à la requête pour afficher les résultats précédents dont le prix est supérieur à 2 € ?

Il faut ajouter le critère sur le prix tout en conservant celui sur le stock. Les deux conditions étant simultanées, on les relie par `AND`.

RÉPONSE À RETENIR

```
SELECT *
FROM produit
WHERE stock < 5 AND prix > 2;
```

Parmi les lignes visibles dans l'annexe, le résultat est :

idProduit	nomProduit	idC	prix	stock	idP
6	tomme de brebis	C	31,50	3	1

La ligne d'identifiant 4 est exclue car son prix est de 1,55 €. Des lignes non affichées dans l'annexe peuvent aussi satisfaire les deux critères.

Question 4.

Sujet p. 6, 7, 13

> Le primeur souhaite connaître le poids de tout son stock. Quelle requête doit-il écrire ?

La colonne `stock` contient les stocks en kilogrammes. La fonction `SUM` additionne cette colonne sur les lignes de la table.

RÉPONSE À RETENIR

```
SELECT SUM(stock)
FROM produit;
```

Les enregistrements visibles dans l'annexe ont des stocks de 10, 8, 12, 4, 6, 3 et 15 kg, soit une somme de 58 kg pour ces seules lignes. Ce n'est pas le total de toute la base : les enregistrements omis dans l'annexe empêchent de déterminer la valeur complète à partir de l'extrait.

Enregistrements pris en compte	Somme des stocks
Lignes visibles dans l'annexe	58 kg

Question 5.

Sujet p. 6, 7, 13

> Lors d'un contrôle, le primeur doit fournir la liste de tous ses produits bio avec le nom du produit, l'identifiant du producteur et le nom du producteur. Quelle requête permet d'établir cette liste ?

Il faut relier `produit` à `producteur` par la clé étrangère `produit.idP` et la clé primaire `producteur.idProducteur`. Un produit est issu d'un producteur bio lorsque `producteur.bio` vaut 1.

RÉPONSE À RETENIR

```
SELECT produit.nomProduit, produit.idP, producteur.nomProducteur
FROM produit JOIN producteur
ON produit.idP = producteur.idProducteur
WHERE producteur.bio = 1;
```

Pour les lignes de produits visibles dans l'annexe, le résultat est :

nomProduit	idP	nomProducteur
poireau	4	Le verger de Julie
pomme	2	La ferme de Pierre
tomme de brebis	1	Laiterie du Haut Cantal
pomme de terre	4	Le verger de Julie

L'annexe omet des lignes de la table `produit` ; le tableau ne porte donc que sur les produits affichés.

Exercice 4

4 points

Architecture matérielle, gestion de processus et réseaux

PARTIE A : Architecture

Question 1.

Sujet p. 8

> On désire créer un réseau informatique équipé de 3 machines et connecté en filaire sur le routeur B. L'adresse IPv4 du nouveau réseau est 192.168.10.0 et le masque de sous-réseau est 255.255.255.0. Toutes les machines devront disposer d'une carte électronique dédiée pour se connecter à ce réseau. Donner son nom.

Pour se connecter à un réseau filaire, chaque machine doit disposer d'une interface réseau adaptée, généralement une carte réseau Ethernet.

RÉPONSE À RETENIR

Il faut une **carte réseau Ethernet** (ou interface réseau Ethernet).

Question 2.

Sujet p. 8

> Quel appareil assure la connexion des machines sur ce réseau ?

Dans un réseau local filaire, un commutateur relie les machines entre elles et au routeur B.

RÉPONSE À RETENIR

Un **commutateur (switch)** assure la connexion des machines sur ce réseau.

Question 3.

Sujet p. 8

> Combien de machines pourront avoir une adresse IPv4 sur ce réseau ?

Le masque 255.255.255.0 est un masque /24 : le dernier octet permet 256 valeurs. L'adresse du réseau, 192.168.10.0, et l'adresse de diffusion, 192.168.10.255, ne sont pas attribuables à des interfaces. Il reste donc 254 adresses utilisables. L'une d'elles doit être attribuée à l'interface du routeur B ; il reste alors 253 adresses pour les machines raccordées en plus du routeur.

RÉPONSE À RETENIR

Le sous-réseau permet d'attribuer **254 adresses IPv4 utilisables au total**, dont une à l'interface du routeur B. Il reste donc **253 adresses pour les autres machines**.

Question 4.

Sujet p. 8

> Donner l'adresse IPv4 du routeur B sur le nouveau réseau.

Le sujet ne fixe pas l'adresse de l'interface de B sur le nouveau sous-réseau. On peut choisir n'importe quelle adresse utilisable de 192.168.10.0/24, à condition qu'elle ne soit pas déjà attribuée. Pour les questions suivantes, on choisit 192.168.10.1.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

Par exemple, l'adresse IPv4 de l'interface de B sur ce réseau peut être **192.168.10.1**, avec le masque **255.255.255.0**.

Question 5.

Sujet p. 8

Donner une adresse IPv4 pour la machine 1.

La machine doit recevoir une adresse utilisable de 192.168.10.0/24, distincte de l'adresse de B choisie à la question précédente. Si B a l'adresse 192.168.10.1, l'adresse 192.168.10.2 convient.

RÉPONSE À RETENIR

Par exemple, on peut attribuer à la machine 1 l'adresse **192.168.10.2**, avec le masque **255.255.255.0**.

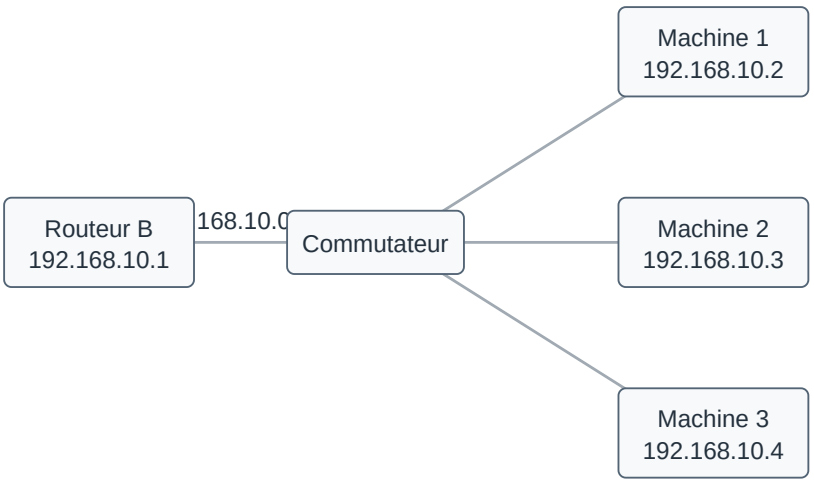
Question 6.

Sujet p. 8

Dessiner sur votre copie une représentation de ce réseau.

Les trois machines sont reliées au commutateur, lui-même relié à l'interface du routeur B sur le nouveau sous-réseau. Le schéma présente un plan d'adressage possible. Les adresses des machines 2 et 3 sont des exemples, comme celles choisies pour B et la machine 1.

Nouveau réseau local 192.168.10.0/24



Les adresses proposées pour les machines 2 et 3 sont des exemples d'adresses utilisables distinctes.

RÉPONSE À RETENIR

Une représentation possible est la suivante :

PARTIE B : Table de routage

Question 7.

Sujet p. 8, 9

> Dans cette question, on utilise le protocole de routage RIP, qui minimise le nombre de routeurs traversés. Recopier et compléter les lignes incomplètes de la table de routage du routeur B pour les réseaux de destination 16.0.0.0, 19.0.0.0 et 152.0.0.0.

Les réseaux 5.0.0.0, 7.0.0.0 et 11.0.0.0 sont directement connectés à B : les lignes correspondantes indiquent une métrique nulle et l'adresse de l'interface de B sur le réseau concerné. Pour une destination distante, la passerelle est l'adresse de l'interface du routeur voisin sur le réseau directement relié à B. D'après le schéma, B rejoint A par 11.0.0.0 et D par 5.0.0.0 : les adresses de ces interfaces voisines sont respectivement 11.0.0.1 et 5.0.0.4.

Le réseau 19.0.0.0 est atteint via A, en traversant un routeur depuis B. Les réseaux 152.0.0.0 et 16.0.0.0 sont atteints via D ; le premier nécessite un routeur traversé, et le second deux, en passant ensuite par F.

RÉPONSE À RETENIR

Les lignes complétées sont :

Réseau de destination	Passerelle	Métrique
16.0.0.0	5.0.0.4	2
19.0.0.0	11.0.0.1	1
152.0.0.0	5.0.0.4	1

Question 8.

Sujet p. 8, 9

> On utilise le protocole de routage OSPF, qui minimise le coût des liaisons. Le coût d'un contact établi entre une machine et l'interface d'un routeur d'un même réseau est zéro ; le coût de la traversée d'un réseau est donné dans le tableau. Déterminer la route suivie si les informations partent du routeur B et se dirigent vers le routeur E. Justifier la réponse après avoir calculé le coût total de chaque parcours possible.

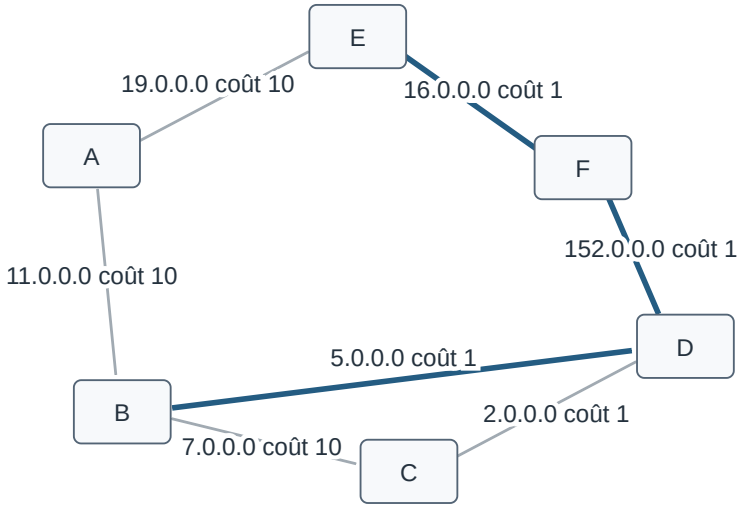
Le schéma donne trois parcours simples possibles entre B et E. Pour chaque parcours, on additionne les coûts des réseaux traversés, tels qu'ils figurent dans le tableau de l'énoncé :

Parcours	Réseaux traversés et coûts	Coût total
B – A – E	11.0.0.0 : 10 ; 19.0.0.0 : 10	10 + 10 = 20
B – D – F – E	5.0.0.0 : 1 ; 152.0.0.0 : 1 ; 16.0.0.0 : 1	1 + 1 + 1 = 3
B – C – D – F – E	7.0.0.0 : 10 ; 2.0.0.0 : 1 ; 152.0.0.0 : 1 ; 16.0.0.0 : 1	10 + 1 + 1 + 1 = 13

Le plus petit coût est 3 ; il correspond au parcours B – D – F – E.

FormaV - document d'entraînement non officiel

Réseau pondéré par les coûts OSPF — chemin optimal de B à E



Le chemin optimal mis en évidence est B–D–F–E.

RÉPONSE À RETENIR

Les informations suivent le chemin **B – D – F – E**, en traversant les réseaux **5.0.0.0**, **152.0.0.0** et **16.0.0.0**. Le coût total est **1 + 1 + 1 = 3**, inférieur aux coûts des deux autres parcours possibles, 20 et 13.

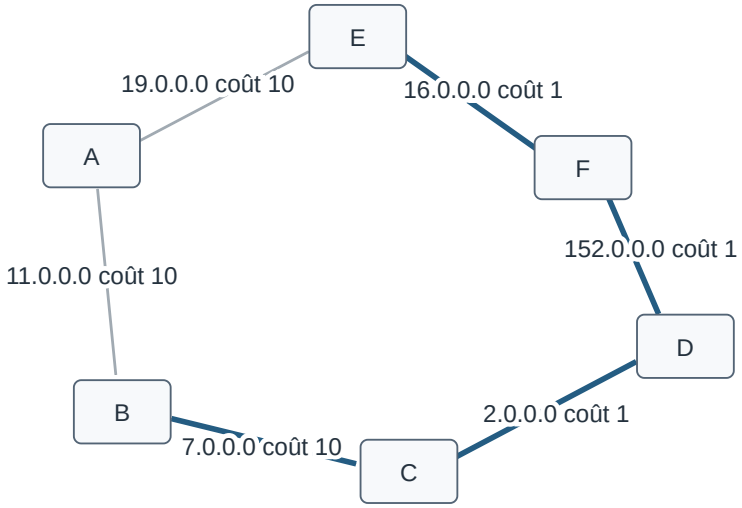
Question 9.

Sujet p. 8, 9

> Si la liaison du réseau 5.0.0.0 est coupée, quel est le nouveau chemin proposé par la table de routage OSPF pour le même parcours des informations ?

Après la coupure du réseau 5.0.0.0, le parcours B – D – F – E n'est plus possible. Les deux autres parcours du schéma restent disponibles : B – C – D – F – E coûte 10 + 1 + 1 + 1 = 13, tandis que B – A – E coûte 10 + 10 = 20. OSPF retient donc le parcours de coût 13.

Réseau après la coupure de 5.0.0.0 — nouveau chemin OSPF



Le réseau 5.0.0.0, coupé, n'est pas représenté. Le chemin restant de coût minimal est B–C–D–F–E.

> **RÉPONSE À RETENIR**
>
> Le nouveau chemin est **B – C – D – F – E**, en traversant les réseaux **7.0.0.0**, **2.0.0.0**, **152.0.0.0** et **16.0.0.0**. Son coût total est **13**.

Exercice 5

4 points

Programmation de base en Python et récursivité

Question 1.

Sujet p. 10

> La fonction calculer_score_sans_bonif a pour paramètre une chaîne de caractères mot et renvoie le score du mot sans bonification. Une erreur s'est glissée dans son corps entre les lignes 5 et 9 incluses. Indiquer la ligne à modifier et corriger l'erreur.

La ligne 8 doit être dans la boucle `for` : elle doit donc être indentée comme le corps de la boucle. De plus, le score doit être augmenté de la valeur de chaque lettre, et non remplacé par cette valeur. On modifie donc la ligne 8 ainsi :

```
    for k in range(len(mot)):
        score = score + scrabble[mot[k]]
```

La boucle ajoute alors successivement la valeur de chaque lettre au score initialement égal à 0.

RÉPONSE À RETENIR

Modifier la ligne 8 en l'indentant dans la boucle et en écrivant `score = score + scrabble[mot[k]]` (ou `score += scrabble[mot[k]]`).

Question 2.

Sujet p. 11

> La fonction calculer_score_avec_bonif a pour paramètres une chaîne de caractères mot et une liste bonif de même longueur que mot, précisant pour chaque lettre la bonification correspondante. Elle renvoie le score du mot en tenant compte des bonifications. Recopier et compléter la ligne incomplète.

À chaque indice `k`, on ajoute au score la valeur de la lettre `mot[k]`, multipliée par le coefficient de bonification correspondant `bonif[k]`. L'exemple confirme la formule : pour `JEU`, le score vaut `8 × 1 + 1 × 3 + 1 × 1 = 12`.

Attention : sur la page du sujet, la clé de la lettre C apparaît comme `"c"` (minuscule), alors que les lettres du mot sont annoncées majuscules. Pour que la fonction traite aussi la lettre majuscule C, il faut corriger cette clé en `"C"` dans le dictionnaire ; cela ne change pas la ligne à compléter.

RÉPONSE À RETENIR

La ligne 8 est :

```
score = score + scrabble[mot[k]] * bonif[k]
```


FormaV - document d'entraînement non officiel

Question 3.(a)

Sujet p. 12

> Écrire une fonction mot_lettre_position qui a pour paramètres une liste de mots liste, un caractère c et une position i (la position 2 correspond à la 3e lettre), et qui renvoie la liste de tous les mots de liste ayant le caractère c en position i.

On parcourt la liste et on ajoute à une nouvelle liste chaque mot dont la lettre à l'indice `i` est `c`. Le test `i < len(mot)` évite d'accéder à une position qui n'existe pas ; le test `0 <= i` garantit que l'indice est non négatif.

RÉPONSE À RETENIR

```
def mot_lettre_position(liste, c, i):
    resultat = []
    for mot in liste:
        if 0 <= i < len(mot) and mot[i] == c:
            resultat.append(mot)
    return resultat
```

Question 3.(b)

Sujet p. 12

> Avec le tirage B A O L E L N, Lucien a composé les mots BAL, NOBLE, ANE, BALLE et LOBE. Il souhaite placer la lettre B sur une case « lettre compte triple », mais pour cela B doit être la 3e lettre du mot. Quel appel doit-il effectuer pour trouver les solutions à son problème ?

Les indices commencent à 0 : la troisième lettre est donc à l'indice `2`. Il faut rechercher les mots ayant `"B"` à cet indice. Dans la liste du sujet, c'est le cas de `NOBLE` et de `LOBE`.

RÉPONSE À RETENIR

```
liste = ["BAL", "NOBLE", "ANE", "BALLE", "LOBE"]
mot_lettre_position(liste, "B", 2)
```

La fonction renvoie `["NOBLE", "LOBE"]`.

Question 4.

Sujet p. 12

> Recopier et compléter la fonction récursive anagramme, qui a pour paramètres deux chaînes de caractères mot1 et mot2 et renvoie True si mot2 est une anagramme de mot1 et False sinon. On dispose de la fonction enlever(mot, c), qui renvoie mot privé de la première occurrence du caractère c.

Si les longueurs des deux chaînes diffèrent, elles ne peuvent pas être des anagrammes : on renvoie `False`. Deux chaînes vides sont des anagrammes, d'où le résultat `True`. Sinon, si la première lettre de `mot1` n'apparaît pas dans `mot2`, on renvoie `False`. Dans le cas contraire, on retire cette lettre de chacune des chaînes et on poursuit récursivement. À chaque appel, les chaînes raccourcissent d'un caractère ; la récursion atteint donc le cas de base.

RÉPONSE À RETENIR

```
def anagramme(mot1, mot2):
    if len(mot1) != len(mot2):
        return False
    elif len(mot1) == 0:
        return True
    elif mot1[0] not in mot2:
        return False
    else:
        return anagramme(mot1[1:], enlever(mot2, mot1[0]))
```

Sujet : https://www.sujetdebac.fr/annales-pdf/2022/spe-numerique-informatique-2022-metropole-2-remplacement-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.