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 • Nouvelle-Calédonie

Code sujet **22-NSIJ2NC1** 3 heures 30

Proposition de corrigé pédagogique • FormaV

Se repérer dans le corrigé

Exercice 1 — Programmation orientée objet en Python : personnages, points de vie et combats
16 réponses • sujet p. 3–5

Exercice 2 — Arbres binaires, dictionnaires Python et parcours
8 réponses • sujet p. 6–7

Exercice 3 — Base de données relationnelle et requêtes SQL sur les personnages
6 réponses • sujet p. 8–8

Exercice 4 — Architecture réseau, adressage IP et protocoles RIP et OSPF
13 réponses • sujet p. 9–11

Exercice 5 — Masques de réseau et tables de routage
11 réponses • sujet p. 12–16

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

Programmation orientée objet en Python : personnages, points de vie et combats

Partie 1

Question 1) a)

Sujet p. 3

> Recopier sur la copie et compléter le constructeur de la classe Personnage positionnant un personnage aux coordonnées choisies x, y et z.

Le constructeur reçoit les trois coordonnées et les enregistre dans les attributs de l'objet courant, accessibles avec `self`.

```
def __init__(self, coordx, coordy, coordz):
    self.x = coordx
    self.y = coordy
    self.z = coordz
```

RÉPONSE À RETENIR

Les coordonnées sont initialisées aux valeurs fournies : `x = coordx`, `y = coordy` et `z = coordz`.

Question 1) b)

Sujet p. 3

> Écrire une méthode avancex de la classe Personnage permettant d'augmenter d'une unité la coordonnée x d'un personnage.

La méthode ajoute 1 à l'attribut `x` de l'objet courant ; elle ne modifie pas `y` ni `z`.

```
def avancex(self):
    self.x = self.x + 1
```

RÉPONSE À RETENIR

Après l'appel à `personnage.avancex()`, la coordonnée `x` a augmenté d'une unité.

Question 1) c)

Sujet p. 3

> Écrire une méthode raz permettant de mettre toutes les coordonnées d'un personnage à zéro.

La méthode affecte la valeur zéro aux trois attributs de coordonnées.

```
def raz(self):
    self.x = 0
    self.y = 0
    self.z = 0
```

RÉPONSE À RETENIR

Après l'appel à `personnage.raz()`, les coordonnées sont `(0, 0, 0)`.

FormaV - document d'entraînement non officiel

Question 1) d)

Sujet p. 3

> Écrire une méthode coord renvoyant les coordonnées d'un personnage sous forme d'un tuple.

La méthode renvoie les trois attributs dans l'ordre `x`, `y`, `z`, sous forme de tuple.

```
def coord(self):
    return (self.x, self.y, self.z)
```

RÉPONSE À RETENIR

`personnage.coord()` renvoie le tuple des coordonnées du personnage, par exemple `(5, 5, 5)` si celui-ci est à cette position.

Question 2) a)

Sujet p. 4

> En utilisant les méthodes écrites dans la partie 1, écrire des lignes de code permettant de créer un personnage arthur démarrant à la position (5,5,5).

On transmet au constructeur les trois coordonnées initiales, dans l'ordre défini par sa signature.

```
arthur = Personnage(5, 5, 5)
```

RÉPONSE À RETENIR

`arthur` commence à la position `(5, 5, 5)`.

Question 2) b)

Sujet p. 4

> En utilisant les méthodes écrites dans la partie 1, écrire des lignes de code permettant d'augmenter d'une unité la coordonnée x du personnage arthur.

On appelle la méthode `avancex` sur l'objet `arthur`.

```
arthur.avancex()
```

RÉPONSE À RETENIR

La coordonnée `x` passe de 5 à 6, tandis que `y` et `z` restent égales à 5. La position devient `(6, 5, 5)`.

Question 2) c)

Sujet p. 4

> En utilisant les méthodes écrites dans la partie 1, écrire des lignes de code permettant d'afficher les coordonnées du personnage arthur.

La méthode `coord` renvoie un tuple ; celui-ci peut être affiché avec `print`.

```
print(arthur.coord())
```

RÉPONSE À RETENIR

Après la modification demandée en 2) b), l'affichage est `(6, 5, 5)`.

Partie 2

Question 1)

Sujet p. 4, 5

> Indiquer les valeurs possibles de la nouvelle variable valeurMerlin après exécution du programme : merlin = Personnage(4,5,8,15) ; merlin.potionmystere() ; valeurMerlin = merlin.get_etat().

La potion ajoute soit `-1`, soit `+1` aux 15 points de vie initiaux. Le choix dépend de la valeur tirée par `random.randint(1, 2)`.

- Si le tirage vaut 1, la vie devient `15 - 1 = 14`.
- Si le tirage vaut 2, la vie devient `15 + 1 = 16`.

> **RÉPONSE À RETENIR**
>
> Les valeurs possibles de `valeurMerlin` sont **14** et **16**.

Question 2)

Sujet p. 4, 5

> Indiquer la valeur de la nouvelle variable valeurMerlin après exécution du programme : merlin = Personnage(4,5,8,20) ; merlin.piege() ; merlin.piege() ; valeurMerlin = merlin.get_etat().

Merlin commence avec 20 points de vie. La méthode `piege` retire 10 points à chaque appel et, dans le code donné, ne déclenche pas `newgame`. Ainsi, après les deux appels, il reste `20 - 10 - 10 = 0` point.

> **RÉPONSE À RETENIR**
>
> `valeurMerlin` vaut **0**.

Question 3)

Sujet p. 4, 5

> Écrire sur la copie la méthode newgame permettant, si le nombre de points de vie est inférieur ou égal à 0, de ramener les coordonnées du personnage à (0,0,0) et de lui attribuer 15 points de vie.

La condition doit couvrir les deux cas demandés : une vie négative ou une vie égale à zéro. La méthode `raz` remet les coordonnées à zéro ; on attribue ensuite 15 points de vie.

```
def newgame(self):
    if self.vie <= 0:
        self.raz()
        self.vie = 15
```

> **RÉPONSE À RETENIR**
>
> Si `self.vie <= 0`, la méthode réinitialise les coordonnées à `(0, 0, 0)` et la vie à `15`. Sinon, elle ne modifie pas le personnage.

Question 4) a)

Sujet p. 5

> Écrire un programme en langage Python permettant la création d'une instance lancelot de cette classe ayant pour attributs 5,5,5 en coordonnées, 15 en point_de_vie et 3 en point_degats.

Le constructeur présenté reçoit les coordonnées et les points de vie, mais pas les dégâts. Le sujet demande l'attribut `point_degats`, alors que la méthode `attaquer` fournie lit `self.degats`. Pour respecter le nom demandé et permettre à cette méthode de fonctionner, on attribue les deux noms à la même valeur.

```
lancelot = Personnage(5, 5, 5, 15)
lancelot.point_degats = 3
lancelot.degats = lancelot.point_degats
```

RÉPONSE À RETENIR

Lancelot est en position `(5, 5, 5)`, possède 15 points de vie et inflige 3 points de dégâts. Les deux attributs de dégâts sont utilisés pour concilier le nom demandé dans l'énoncé et celui lu par la méthode `attaquer`.

Question 4) b)

Sujet p. 5

> Écrire un programme en langage Python permettant la création d'une instance sorcier de cette classe ayant pour attributs 6,5,5 en coordonnées, 15 en point_de_vie et 2 en point_degats.

Comme pour Lancelot, on crée l'instance avec les quatre arguments acceptés par le constructeur. Le sujet demande `point_degats`, tandis que `attaquer` utilise `degats` ; on donne donc à ces deux attributs la même valeur.

```
sorcier = Personnage(6, 5, 5, 15)
sorcier.point_degats = 2
sorcier.degats = sorcier.point_degats
```

RÉPONSE À RETENIR

Le sorcier est en position `(6, 5, 5)`, possède 15 points de vie et inflige 2 points de dégâts. L'attribut `degats` permet à la méthode fournie `attaquer` de lire cette valeur.

Question 4) c)

Sujet p. 5

> Écrire un programme en langage Python permettant à lancelot d'attaquer le sorcier une première fois.

La méthode `attaquer` prend en argument le personnage visé. Lancelot passe donc le sorcier à cette méthode.

```
lancelot.attaquer(sorcier)
```

RÉPONSE À RETENIR

L'attaque inflige 3 points de dégâts au sorcier : ses points de vie passent de 15 à **12**.

Question 4) d)

Sujet p. 5

> Écrire un programme en langage Python permettant au sorcier d'attaquer à son tour lancelot.

Le sorcier attaque à son tour en passant Lancelot comme personnage visé.

```
sorcier.attaquer(lancelot)
```

> **RÉPONSE À RETENIR**
>
> Lancelot perd 2 points de vie et passe de 15 à **13**. À ce stade, le sorcier conserve ses 12 points de vie après l'attaque précédente.

Question 4) e)

Sujet p. 5

> Écrire un programme en langage Python permettant à lancelot de répliquer en attaquant quatre fois de suite le sorcier.

Une boucle avec quatre itérations réalise les quatre attaques successives de Lancelot.

```
for i in range(4):
    lancelot.attaquer(sorcier)
```

Trace de la vie du sorcier après chaque attaque : `12 - 3 = 9`, puis `9 - 3 = 6`, puis `6 - 3 = 3`, puis `3 - 3 = 0`. Après cette dernière attaque, `perdre_vie` appelle `newgame` ; la vie du sorcier est donc réinitialisée à 15 et ses coordonnées à zéro.

> **RÉPONSE À RETENIR**
>
> Après les quatre attaques, le sorcier a **15 points de vie** et ses coordonnées sont `(0, 0, 0)`.

Question 4) f)

Sujet p. 5

> Écrire un programme en langage Python permettant d'afficher le nombre de points de vie de lancelot et du sorcier.

La méthode `get_etat()` renvoie les points de vie. On l'appelle pour chaque personnage afin d'afficher leurs valeurs après toutes les attaques.

```
print(lancelot.get_etat())
print(sorcier.get_etat())
```

Lancelot a perdu 2 points lors de l'attaque du sorcier et n'a pas été réinitialisé : il lui reste 13 points. Le sorcier a été réinitialisé à 15 points après la quatrième attaque de Lancelot.

> **RÉPONSE À RETENIR**
>
> Les valeurs affichées, dans l'ordre, sont **13** pour Lancelot et **15** pour le sorcier.

Exercice 2

4 points

Arbres binaires, dictionnaires Python et parcours

Question 1)

Sujet p. 6

> Indiquer pourquoi il s'agit d'un arbre binaire, en vous appuyant sur le schéma 1.

Un arbre binaire est un arbre enraciné dans lequel chaque nœud possède au plus deux fils. Sur le schéma, A est la racine et chaque nœud a au maximum deux fils : par exemple, B a pour fils C et D, et F a pour fils G et H. Un nœud peut aussi n'avoir qu'un seul fils, comme C ou G.

RÉPONSE À RETENIR

Le schéma représente bien un arbre binaire : il est enraciné en A et chacun de ses nœuds a au plus deux fils.

Question 2) a)

Sujet p. 6

> Pour la variable V définie par V = {'A': 1, 'B': 2, 'C': 3, 'D': 5, 'E': 10, 'F': 15, 'G': 4, 'H': 5, 'I': 5, 'J': 7}, indiquer son type.

La variable est écrite entre accolades sous la forme de couples « clé : valeur ». En Python, cette structure est un dictionnaire : les lettres sont les clés et les nombres associés sont les valeurs.

RÉPONSE À RETENIR

`V` est un dictionnaire Python.

Question 2) b)

Sujet p. 6

> Écrire l'instruction permettant d'accéder à la valeur « 7 » stockée dans la variable V.

Pour consulter la valeur associée à une clé dans un dictionnaire, on place cette clé entre crochets après le nom du dictionnaire. La valeur 7 est associée à la clé `'J'`.

RÉPONSE À RETENIR

`V['J']` renvoie `7`.

Question 2) c)

Sujet p. 6

> Écrire une fonction somme en langage Python qui prend un paramètre W du même type que la variable V et qui renvoie la somme des valeurs de W.

La méthode `values()` fournit les valeurs du dictionnaire. La fonction intégrée `sum` en calcule la somme.

```
def somme(W):
    return sum(W.values())
```

Pour le dictionnaire V de l'énoncé, le calcul est : 1 + 2 + 3 + 5 + 10 + 15 + 4 + 5 + 5 + 7 = 57.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

La fonction renvoie la somme des valeurs de tout dictionnaire transmis en paramètre. Ainsi, `somme(V)` renvoie `57`.

Question 2) d)

Sujet p. 6

Écrire une fonction VMax en langage Python qui prend un paramètre W du même type que la variable V et qui renvoie la lettre ayant la valeur maximale de W.

On parcourt les clés du dictionnaire en comparant les valeurs associées à l'aide de l'argument `key=W.get`. La fonction renvoie ainsi la clé de valeur maximale.

```
def VMax(W):
    return max(W, key=W.get)
```

Dans V, la valeur maximale est 15, associée à la lettre F.

RÉPONSE À RETENIR

La fonction renvoie la lettre de valeur maximale. Pour le dictionnaire V donné, `VMax(V)` renvoie `'F'`.

Question 3)

Sujet p. 7

Indiquer, en justifiant la réponse, le rôle de l'algorithme CALCUL(T : arbre).

Si l'arbre est vide, l'algorithme renvoie 0. Sinon, il compte 1 pour sa racine, puis ajoute les résultats obtenus en appliquant récursivement le même calcul à ses sous-arbres gauche et droit. Ainsi, chaque nœud non vide est compté une fois et les arbres vides ne contribuent pas au total.

RÉPONSE À RETENIR

`CALCUL(T)` renvoie le nombre de nœuds de l'arbre T. Dans l'arbre du schéma 1, il y a 10 nœuds (A à J) ; le résultat est donc `10`.

Question 4) a)

Sujet p. 6, 7

On applique l'algorithme VISITE à l'arbre du schéma 1. Indiquer l'affichage obtenu.

L'algorithme affiche d'abord la racine, puis visite récursivement le sous-arbre gauche avant le sous-arbre droit. Sur le schéma, on commence donc par A. Dans le sous-arbre de B, l'ordre est C, E, puis D. Dans celui de F, l'ordre est G, I, puis H, J.

RÉPONSE À RETENIR

L'affichage obtenu est : `A B C E D F G I H J`.

Question 4) b)

Sujet p. 7

Indiquer le type de parcours réalisé par l'algorithme VISITE.

À chaque nœud, VISITE affiche la clé avant d'appeler la procédure sur les sous-arbres gauche puis droit. Il s'agit donc d'un parcours en profondeur préfixe, aussi appelé parcours en préordre : racine, gauche, droite.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

VISITE réalise un parcours en profondeur préfixe (préordre).

FormaV - document d'entraînement non officiel

Exercice 3

4 points

Base de données relationnelle et requêtes SQL sur les personnages

Question 1)

Sujet p. 8

> Écrire une requête permettant d'afficher le nom et les points de tous les personnages.

On sélectionne les colonnes demandées, `nom` et `points`, dans la table `Personnage`. Comme aucun filtre n'est demandé, la requête ne comporte pas de clause `WHERE`.

```
SELECT nom, points
FROM Personnage;
```

Voici le résultat sur les lignes de la table fournies dans le sujet.

RÉPONSE À RETENIR

nom	points
Merlin	40
Galehaut	40
Lancelot	50
Gareth	20
Galahad	25
Gaheris	30
Erek	25
Lamorak	30
Perceval	35
Hector	50
Keud	80
Bedivere	20
Lucan	25

Question 2)

Sujet p. 8

> Une erreur a été identifiée. Le frère d'Arthur ne s'appelle pas Keud mais Antor. Écrire une requête permettant la modification de ce nom.

Dans la table, la ligne à corriger est celle où `frere_de` vaut `Arthur` et où le nom est `Keud`. On modifie uniquement la colonne `nom` de cette ligne.

FormaV - document d'entraînement non officiel

```
UPDATE Personnage
SET nom = 'Antor'
WHERE frere_de = 'Arthur' AND nom = 'Keud';
```

RÉPONSE À RETENIR

Le nom **Keud** devient **Antor**. La valeur de `frere_de` reste `Arthur`.

Question 3)

Sujet p. 8

> Écrire une requête permettant d'ajouter le nom_qualite « roi » à la table Qualite ayant 8 pour Idqualite.

Il faut ajouter dans `Qualite` une ligne associant l'identifiant 8 au nom de qualité `roi`.

```
INSERT INTO Qualite (Idqualite, nom_qualite)
VALUES (8, 'roi');
```

RÉPONSE À RETENIR

La ligne ajoutée est `(8, 'roi')`.

Question 4)

Sujet p. 8

> Écrire une requête permettant de créer un quatorzième personnage : le roi Arthur avec 100 points.

Les identifiants de 1 à 13 étant déjà utilisés, on attribue 14 au quatorzième personnage. La question ne précise ni la valeur de `frere_de` ni l'identifiant de qualité. En supposant que la qualité `roi` ajoutée à la question 3 est attribuée à Arthur et que son frère n'est pas communiqué, on peut insérer la ligne suivante, en reprenant la notation `NC` du tableau.

```
INSERT INTO Personnage (Idperso, nom, frere_de, points, Idqualite)
VALUES (14, 'Arthur', 'NC', 100, 8);
```

RÉPONSE À RETENIR

Cette insertion crée Arthur avec l'identifiant 14, 100 points et la qualité d'identifiant 8, soit `roi` selon l'hypothèse indiquée. Les valeurs non précisées par l'énoncé sont donc explicitées comme hypothèses, et non comme données fournies.

Question 5)

Sujet p. 8

> Écrire une requête permettant d'afficher le nom et le nom de la qualité des personnages dont les points sont égaux à 40.

On relie les tables par leur colonne commune `Idqualite`, puis on sélectionne les personnages ayant exactement 40 points. Les données indiquent que Merlin a la qualité d'identifiant 2 (`magicien`) et Galehaut celle d'identifiant 5 (`seigneur`).

```
SELECT Personnage.nom, Qualite.nom_qualite
FROM Personnage
JOIN Qualite ON Personnage.Idqualite = Qualite.Idqualite
WHERE Personnage.points = 40;
```

Résultat sur les données fournies dans le sujet :

RÉPONSE À RETENIR

nom	nom_qualite
Merlin	magicien
Galehaut	seigneur

Question 6)

Sujet p. 8

> On applique la requête UPDATE Personnage SET points=points+10 WHERE points < 40. Indiquer alors le nombre de points des personnages suivants : Arthur, Perceval et Merlin.

La condition `points < 40` est stricte : seuls les personnages ayant moins de 40 points reçoivent 10 points supplémentaires. En supposant que la création d'Arthur demandée à la question 4 a bien été effectuée, Arthur a 100 points et reste inchangé. Perceval passe de 35 à `35 + 10 = 45`. Merlin a exactement 40 points, donc il ne satisfait pas la condition et reste à 40.

RÉPONSE À RETENIR

Arthur : **100 points** ; Perceval : **45 points** ; Merlin : **40 points**.

FormaV - document d'entraînement non officiel

Exercice 4

4 points

Architecture réseau, adressage IP et protocoles RIP et OSPF

Question 1) a)

Sujet p. 9, 10, 11

> Indiquer l'adresse du réseau local dont fait partie l'ordinateur Portable 3.

Portable 3 appartient au réseau local T1. Portable 1 et l'interface de R1 reliée à S1 ont respectivement les adresses 195.168.1.40/24 et 195.168.1.1/24. Avec un masque /24, les trois premiers octets désignent le réseau ; son adresse est donc 195.168.1.0.

> **RÉPONSE À RETENIR**
>
> L'adresse du réseau local de Portable 3 est **195.168.1.0/24**.

Question 1) b)

Sujet p. 9, 10, 11

> Indiquer une adresse possible pour l'ordinateur Portable 3.

Dans le réseau 195.168.1.0/24, les adresses utilisables pour les machines vont de 195.168.1.1 à 195.168.1.254. L'adresse 195.168.1.1 est celle de l'interface de R1, et 195.168.1.40 est celle de Portable 1. On peut donc proposer 195.168.1.2, à condition qu'elle ne soit pas déjà attribuée à un autre ordinateur : le sujet ne donne pas les adresses des autres portables de T1.

> **RÉPONSE À RETENIR**
>
> Par exemple, **195.168.1.2/24**, si cette adresse n'est pas déjà utilisée sur T1.

Question 1) c)

Sujet p. 9, 10, 11

> Indiquer, en justifiant, le nombre d'adresses encore disponibles pour l'ordinateur Portable 4 du réseau local T2.

Le réseau T2 est 197.162.1.0/24. Il contient 256 adresses au total ; après exclusion de l'adresse du réseau (.0) et de l'adresse de diffusion (.255), il reste 254 adresses utilisables pour les équipements. Deux sont déjà attribuées d'après le tableau : 197.162.1.1 à l'interface de R2 et 197.162.1.50 à Portable 5. Il reste donc 254 − 2 = 252 adresses utilisables.

> **RÉPONSE À RETENIR**
>
> Il reste **252 adresses IP utilisables** dans T2 pour Portable 4, en tenant compte des adresses déjà attribuées dans le tableau.

Question 2)

Sujet p. 10, 11

> Indiquer les adresses IP des interfaces du routeur 3. Chaque adresse IP est la première disponible sur la plage d'adressage de chaque réseau connecté.

Pour l'interface 1, côté S3, Portable 6 à Portable 9 appartiennent au réseau 199.160.1.0/24 : la première adresse disponible est 199.160.1.1. Pour l'interface 2, le lien avec R2 est sur le réseau 198.164.3.0/24, puisque R2 utilise 198.164.3.1 ; R3 prend la première autre adresse disponible, 198.164.3.2. Pour l'interface 3, le lien avec R4 est sur le réseau 200.158.4.0/24, où R4 utilise 200.158.4.2 ; la première adresse disponible est 200.158.4.1.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

Les adresses des interfaces de R3 sont :

- Interface 1, côté S3 : **199.160.1.1/24** ;
- Interface 2, côté R2 : **198.164.3.2/24** ;
- Interface 3, côté R4 : **200.158.4.1/24**.

Question 3) a)

Sujet p. 9, 10, 11

> Lors du jeu, l'ordinateur Portable 1 veut communiquer avec l'ordinateur Portable 5. Indiquer trois parcours possibles en listant tous les éléments utilisés du réseau.

Portable 1 est relié à S1, lui-même relié à R1. Portable 5 est relié à S2, lui-même relié à R2. Entre les routeurs, la figure montre notamment les liaisons R1–R2, R1–R4, R4–R2, R4–R3 et R3–R2. On peut donc emprunter trois chemins différents entre les deux portables.

RÉPONSE À RETENIR

Trois parcours possibles sont :

- Portable 1 → S1 → R1 → R2 → S2 → Portable 5 ;
- Portable 1 → S1 → R1 → R4 → R2 → S2 → Portable 5 ;
- Portable 1 → S1 → R1 → R4 → R3 → R2 → S2 → Portable 5.

Question 3) b)

Sujet p. 9, 10, 11

> Indiquer le plus court chemin au sens du protocole RIP, minimisant le nombre de routeurs traversés (sauts), en précisant son nombre de sauts, pour la communication entre Portable 1 et Portable 5.

Selon le critère donné pour RIP, on minimise le nombre de liaisons entre routeurs traversées. La liaison directe R1–R2 correspond à un saut ; les chemins passant par R4 ou par R3 traversent davantage de liaisons entre routeurs. Le chemin direct est donc le plus court selon ce critère.

RÉPONSE À RETENIR

Le chemin est **Portable 1 → S1 → R1 → R2 → S2 → Portable 5** ; il comporte **1 saut** entre routeurs.

Question 3) c)

Sujet p. 9, 10, 11

> La liaison R1-R2 vient de se rompre. Indiquer le plus court chemin au sens du protocole RIP, minimisant le nombre de routeurs traversés, pour la communication entre Portable 1 et Portable 5.

Si la liaison R1–R2 est rompue, le trajet direct n'est plus possible. Parmi les chemins restants, R1–R4–R2 traverse deux liaisons entre routeurs, tandis que R1–R4–R3–R2 en traverse trois. RIP retient donc le trajet par R4.

RÉPONSE À RETENIR

Le plus court chemin est **Portable 1 → S1 → R1 → R4 → R2 → S2 → Portable 5**, soit **2 sauts** entre routeurs.

FormaV - document d'entraînement non officiel

Question 4) a)

Sujet p. 11

> La liaison R1-R2 entre les deux réseaux locaux est rétablie en changeant le câble de raccordement. Parmi les quatre propositions, examiner la proposition a) : Internet.

« Internet » désigne un réseau de réseaux, et non le type du câble physique permettant de relier directement deux routeurs. Cette proposition ne convient pas.

> **RÉPONSE À RETENIR**
>
> **a) Internet : réponse incorrecte.**

Question 4) b)

Sujet p. 11

> La liaison R1-R2 entre les deux réseaux locaux est rétablie en changeant le câble de raccordement. Parmi les quatre propositions, examiner la proposition b) : VGA.

Un câble VGA sert à transmettre un signal vidéo, notamment vers un écran ; il n'est pas destiné à la liaison réseau entre les routeurs.

> **RÉPONSE À RETENIR**
>
> **b) VGA : réponse incorrecte.**

Question 4) c)

Sujet p. 11

> La liaison R1-R2 entre les deux réseaux locaux est rétablie en changeant le câble de raccordement. Parmi les quatre propositions, examiner la proposition c) : Ethernet.

Ethernet est une technologie de réseau utilisée pour relier des équipements réseau, notamment au moyen de câbles adaptés. C'est la proposition qui convient pour rétablir la liaison entre R1 et R2.

> **RÉPONSE À RETENIR**
>
> **c) Ethernet** est la bonne réponse.

Question 4) d)

Sujet p. 11

> La liaison R1-R2 entre les deux réseaux locaux est rétablie en changeant le câble de raccordement. Parmi les quatre propositions, examiner la proposition d) : HDMI.

Un câble HDMI transporte principalement des signaux audiovisuels ; il ne convient pas à la liaison réseau entre les deux routeurs.

> **RÉPONSE À RETENIR**
>
> **d) HDMI : réponse incorrecte.**

Question 5) a)

Sujet p. 11

> Avec une bande passante de référence de 10^9 bps, le coût d'une liaison est donné par coût = 10^9/d, où d est le débit en bps. On admet que si le quotient est inférieur ou égal à 1, le coût est égal à 1. La liaison entre les routeurs R1 et R2 a un coût de 10. Calculer le débit de cette liaison en Mbps.

On utilise la formule coût = 10^9/d. Comme le coût vaut 10, on obtient 10 = 10^9/d, donc d = 10^9/10 = 10^8 bits par seconde. Or 1 Mbps = 10^6 bits par seconde : le débit vaut donc 10^8/10^6 = 100 Mbps.

> **RÉPONSE À RETENIR**
>
> Le débit de la liaison R1–R2 est **100 Mbps**.

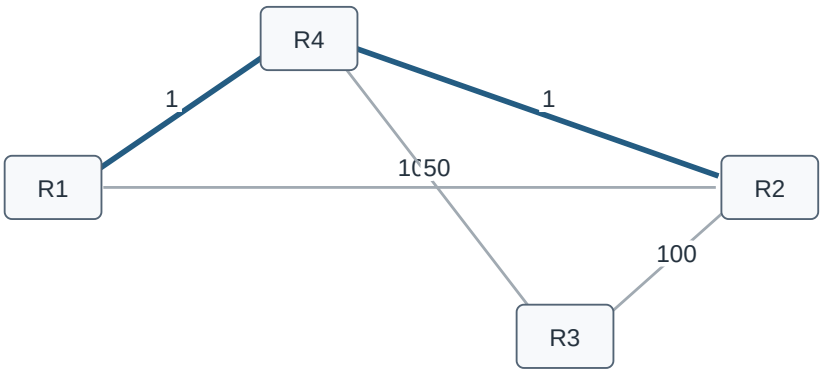
Question 5) b)

Sujet p. 11

> L'ordinateur Portable 1 communique avec l'ordinateur Portable 5. Le routeur R1 doit transmettre les paquets au routeur R2. À partir des débits du tableau (R1-R2 : inconnu, R1-R4 : 1000 Mbps, R2-R3 : 10 Mbps, R2-R4 : 1000 Mbps et R3-R4 : 20 Mbps), déterminer le chemin de meilleur coût entre R1 et R2 et indiquer ce coût. Justifier les réponses.

On convertit les débits en bits par seconde avant d'appliquer la formule. R1–R2 a un coût de 10, obtenu à la question précédente. Pour R1–R4 et R2–R4, le débit est 1000 Mbps = 10^9 bps : le coût est 1 pour chacune. Pour R2–R3, 10 Mbps = 10^7 bps : le coût est 100. Pour R3–R4, 20 Mbps = 2 × 10^7 bps : le coût est 50. Les chemins possibles les moins coûteux à comparer sont donc R1–R2, de coût 10, et R1–R4–R2, de coût 1 + 1 = 2. Le chemin R1–R4–R3–R2 coûte 1 + 50 + 100 = 151. Le minimum est donc le chemin passant par R4.

Coûts des liaisons entre routeurs et chemin de coût minimal



Les coûts des cinq liaisons du tableau sont indiqués ; le chemin R1–R4–R2, de coût total 2, est mis en évidence.

> **RÉPONSE À RETENIR**
>
> Le chemin de meilleur coût est **R1 → R4 → R2**, de coût total **2**.

FormaV - document d'entraînement non officiel

Exercice 5

4 points

Masques de réseau et tables de routage

Question 1) a) — 255.255.225.0

Sujet p. 12

> Pour l'écriture de masque de réseau 255.255.225.0, indiquer et justifier si elle correspond à un masque de réseau valide.

Un masque valide doit être formé d'une suite de bits 1, puis uniquement de bits 0. Les deux premiers octets, 255.255, fournissent déjà 16 bits à 1. Or 225 s'écrit en binaire 11100001 : après des bits 0, le dernier bit vaut de nouveau 1. La suite n'est donc pas de la forme exigée.

> **RÉPONSE À RETENIR**
>
> **255.255.225.0 n'est pas un masque de réseau valide.** Son troisième octet contient un 1 après des 0.

Question 1) a) — 255.255.224.0

Sujet p. 12

> Pour l'écriture de masque de réseau 255.255.224.0, indiquer et justifier si elle correspond à un masque de réseau valide.

Les deux premiers octets, 255.255, donnent 16 bits à 1. Le troisième octet, 224, s'écrit 11100000 : il prolonge la suite de 1, puis les bits restants sont tous des 0. La forme binaire complète est donc une suite de 19 bits à 1 suivie de 13 bits à 0.

> **RÉPONSE À RETENIR**
>
> **255.255.224.0 est un masque de réseau valide**, de notation CIDR /19.

Question 1) b)

Sujet p. 12, 13

> On considère que les masques en notation CIDR /0 et /32 sont valides. Écrire en langage Python la fonction cidr qui prend en paramètre une liste de 32 bits et qui renvoie l'entier n de la notation CIDR si la liste correspond à l'écriture binaire d'un masque de réseau valide, et renvoie -1 sinon. La fonction convBin, qui prend en paramètre une liste de quatre nombres entiers compris entre 0 et 255 et renvoie la liste des 32 bits de l'écriture binaire correspondante, est fournie.

On parcourt les 32 bits dans l'ordre. Tant qu'aucun 0 n'a été rencontré, chaque 1 augmente le nombre de bits de réseau. Après le premier 0, tout 1 ultérieur rend le masque invalide. On vérifie aussi que la liste comporte bien 32 valeurs, chacune égale à 0 ou à 1. Le compteur peut prendre toute valeur de 0 à 32, ce qui inclut les cas /0 et /32 demandés.

La boucle examine chaque bit une seule fois : le temps d'exécution est proportionnel à la taille de la liste.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

```
def cidr(bits):
    if len(bits) != 32:
        return -1

    n = 0
    zero_rencontre = False

    for bit in bits:
        if bit != 0 and bit != 1:
            return -1
        if bit == 1:
            if zero_rencontre:
                return -1
            n += 1
        else:
            zero_rencontre = True

    return n
```

Par exemple, cette fonction renvoie 18 pour le masque correspondant à 255.255.192.0. Pour 255.252.128.0, elle rencontre un 1 après un 0 et renvoie -1.

Question 2) Réponse A

Sujet p. 13

> Parmi les réponses proposées pour afficher la table de routage, examiner la réponse A : dir (sous Windows) ou ls (sous Linux).

`dir` sous Windows et `ls` sous Linux servent à afficher le contenu d'un répertoire. Ce ne sont pas des commandes d'affichage de la table de routage.

RÉPONSE À RETENIR

Réponse A incorrecte.

Question 2) Réponse B

Sujet p. 13

> Parmi les réponses proposées pour afficher la table de routage, examiner la réponse B : cacls (sous Windows) ou chmod (sous Linux).

`cacls` sous Windows et `chmod` sous Linux servent à gérer des permissions d'accès aux fichiers. Elles n'affichent pas la table de routage.

RÉPONSE À RETENIR

Réponse B incorrecte.

Question 2) Réponse C

Sujet p. 13

> Parmi les réponses proposées pour afficher la table de routage, examiner la réponse C : route print (sous Windows) ou ip route (sous Linux) ou route -n (sous Linux).

`route print` sous Windows, `ip route` sous Linux et `route -n` sous Linux permettent d'afficher les informations de routage de la machine.

> **RÉPONSE À RETENIR**
>
> **Réponse C correcte.**

Question 2) Réponse D

Sujet p. 13

> Parmi les réponses proposées pour afficher la table de routage, examiner la réponse D : ping.

`ping` sert à tester la connectivité vers une machine, notamment au moyen de messages ICMP. Il n'affiche pas la table de routage.

> **RÉPONSE À RETENIR**
>
> **Réponse D incorrecte.**

Question 2) Réponse E

Sujet p. 13

> Parmi les réponses proposées pour afficher la table de routage, examiner la réponse E : tracert (sous Windows) ou traceroute (sous Linux).

`tracert` sous Windows et `traceroute` sous Linux servent à observer les routeurs traversés vers une destination. Ils ne servent pas à afficher la table de routage locale.

> **RÉPONSE À RETENIR**
>
> **Réponse E incorrecte.**

Question 3) a)

Sujet p. 14, 16

> À l'étape n°1, les routeurs connaissent uniquement les adresses réseaux de leurs voisins. Compléter, sur le document réponse Exercice 5 page 16, les tables de routage des routeurs R2 et R3 à l'étape n°1.

À l'étape n°1, chaque routeur ne connaît que les réseaux auxquels il est directement connecté. Les interfaces sont celles indiquées sur la figure : R2 utilise eth0 pour le réseau 192.168.2.0/24 et eth1 pour le réseau 192.168.3.0/24 ; R3 utilise eth0 pour le réseau 192.168.3.0/24 et eth1 pour le réseau 192.168.4.0/24. Une destination directement connectée a pour routeur suivant « direct » et pour nombre de sauts 0.

FormaV - document d'entraînement non officiel

RÉPONSE À RETENIR

Routeur R2, étape n°1 :

Destination	Interface	Routeur suivant	Saut
192.168.2.0/24	192.168.2.2	direct	0
192.168.3.0/24	192.168.3.2	direct	0

Routeur R3, étape n°1 :

Destination	Interface	Routeur suivant	Saut
192.168.3.0/24	192.168.3.3	direct	0
192.168.4.0/24	192.168.4.3	direct	0

Question 3) b)

Sujet p. 14, 15, 16

> À l'étape n°2, R2 envoie le contenu de sa table à ses voisins puis les routeurs R1 et R3 mettent à jour leur table ; ensuite R1 et R3 envoient le contenu de leur table à R2, puis le routeur R2 met à jour sa table de routage. Compléter, sur le document réponse Exercice 5 page 16, la table de routage du routeur R1 à l'étape n°2. La table de routage de R2 se stabilisant à partir de l'étape n°2 est donnée dans le sujet.

À l'étape n°2, R1 reçoit de R2 la table de l'étape n°1. Il apprend donc l'existence du réseau 192.168.3.0/24, accessible en passant par R2. Pour sortir, R1 emprunte son interface 192.168.2.1 ; le prochain routeur est R2, dont l'interface d'entrée sur ce lien est 192.168.2.2. Le réseau annoncé est à 0 saut de R2, donc à 1 saut de R1. Les deux réseaux directement connectés à R1 restent dans sa table avec 0 saut.

RÉPONSE À RETENIR

Table du routeur R1 à l'étape n°2 :

Destination	Interface	Routeur suivant	Saut
192.168.1.0/24	192.168.1.1	direct	0
192.168.2.0/24	192.168.2.1	direct	0
192.168.3.0/24	192.168.2.1	192.168.2.2	1

Question 3) c)

Sujet p. 14, 15, 16

> À l'étape n°3, R2 envoie le contenu de sa table à ses voisins puis les routeurs R1 et R3 mettent à jour leur table ; ensuite R1 et R3 envoient le contenu de leur table à R2, puis le routeur R2 met à jour sa table de routage. Compléter, sur le document réponse Exercice 5 page 16, la table de routage de R1 à l'étape n°3.

À l'étape n°3, R1 reçoit la table stabilisée de R2. R2 annonce le réseau 192.168.4.0/24 à 1 saut ; R1 l'atteint donc en 2 sauts via R2. Pour ce trajet, R1 sort par son interface 192.168.2.1 et le routeur suivant est R2 à l'adresse 192.168.2.2. Les réseaux 192.168.1.0/24 et 192.168.2.0/24 restent directement connectés, et le réseau 192.168.3.0/24 reste accessible via R2 en 1 saut.

RÉPONSE À RETENIR

Table du routeur R1 à l'étape n°3 :

Destination	Interface	Routeur suivant	Saut
192.168.1.0/24	192.168.1.1	direct	0
192.168.2.0/24	192.168.2.1	direct	0
192.168.3.0/24	192.168.2.1	192.168.2.2	1
192.168.4.0/24	192.168.2.1	192.168.2.2	2

Sujet : https://aeif.fr/wp-content/uploads/2022/12/22_NSIJ2NC1.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.