Ce document a été mis en ligne par l'organisme FormaV®.

Toute reproduction, représentation ou diffusion, même partielle, sans autorisation préalable, est strictement interdite.

Pour consulter tous nos sujets d'annales et corrigés du Bac Général Maths + NSI, veuillez visiter :
bac-maths-nsi.fr/annales-sujets-et-corriges

Centres étrangers - juin 2026 - sujet 2 (corrigé)

Exercice 1 (POO et sécurisation des communications)
Partie A

1. Les instructions suivantes conviennent :

```
from donnees import vehicules, energie
from chauffeur import Chauffeur
from taxi import Taxi
from client import Client
from course import Course
```

2. L'instruction suivante convient :

```
chauffeur0 = Chauffeur('Doe', 'John', '140159320012')
```

3. Les instructions suivantes conviennent :

```
taxi0 = Taxi('HG-818-AV', 'standard', 'électrique', False)
taxi0.choix_chauffeur(chauffeur0)
```

4. L'assertion suivante convient :

```
assert taxi.est_libre()
```

5. L'instruction suivante convient :

```
taxi.modifier_libre(False)
```

6. Les instructions suivantes conviennent :

```
client0 = Client('Doe', 'Jeanne', '6 rue des ordinateurs, Paris', 2)
course0 = Course(client0, taxi0, '6 rue des ordinateurs, Paris',
                 '211 avenue Jean Jaurès, Paris')
```

7. Le code suivant convient :

```
def temps(self):
    return (self.date_arrivee - self.date_depart).total_seconds()/60
```

8. Le code suivant convient :

```
def tarif(self):
    type_vehicule = self.taxi.type_vehicule
    donnees_tarifaire = vehicules[type_vehicule]
    prise_en_charge = donnees_tarifaire['prise_en_charge']
    tarif_h = donnees_tarifaire['tarif_horaire']
    prix_km = donnees_tarifaire['prix_km']
    tarif = prise_en_charge
    tarif = tarif + prix_km * self.distance()
    tarif = tarif + tarif_h * self.temps()/60
    return tarif
```

Partie B

9. Un protocole de chiffrement utilise un procédé de chiffrement associé à une clé de chiffrement et un procédé de déchiffrement associé à une clé de déchiffrement. Dans le cas d'un algorithme symétrique, ces deux clés sont identiques. En revanche, dans le cas d'un algorithme asymétrique, elles sont différentes.
10. Un agent qui intercepterait la transmission contenant la clé de sessions aurait accès à cette clé. Il pourrait alors, par exemple, déchiffrer les communications entre le client et le chauffeur après interception des messages, usurper l'identité du chauffeur en communiquant directement avec le client, ou usurper l'identité du client en communiquant directement avec le chauffeur.
11. On utilisera la stratégie suivante basée sur un chiffrement hybride. Le chauffeur possède une clé publique PUB et une clé privée PRI (par exemple avec le protocole RSA). La clé publique PUB est envoyé en clair à la centrale. La centrale encode la clé de session SES en utilisant la clé publique PUB et transmet le message chiffré obtenu au chauffeur. Le chauffeur déchiffre alors le message en utilisant sa clé privée PRI pour obtenir le message SES. Ainsi, le chiffrement asymétrique est utilisé uniquement pour la transmission de la clé de session et on garde le chiffrement symétrique (avec cette clé de session) pour le reste des données, ce qui ne ralentit pas les échanges entre la centrale et le chauffeur. La transmission de la clé de session est bien sécurisée car un agent extérieur ne peut pas déchiffrer le message contenant la clé de session puisque la clé privée n'a jamais transité dans la communication.

Exercice 2 (Langage SQL, réseaux et routage)

Partie A

1. La requête suivante convient :

```
SELECT modele FROM ordinateur WHERE etat = 'Réparation'
```

2. La requête suivante convient :

```
INSERT INTO ordinateur VALUES (22, 'Thomson' , 'MO5' , P'anne')
```

3. La requête suivante convient :

```
UPDATE ordinateur SET etat = 'Fonctionnel' WHERE id_ordi = 9
```

4. Cette requête renvoie le nombre d'ordinateurs de la marque Commodore en état fonctionnel (ici, 2)

Partie B

5. On a le schéma relationnel suivant :
   - `jeu(id_jeu: INT, titre: TEXT, genre: TEXT, etat: TEXT, #id_plat: INT)`
   - `plateforme(id_plat: INT, nom: TEXT, bit: INT)`
6. La requête suivante convient :

```
SELECT titre FROM jeu
JOIN plateforme ON jeu.id_plat = plateforme.id_plat
WHERE plateforme.nom = 'Oric'
```

7. La requête suivante convient :

```
SELECT titre FROM jeu
JOIN ordinateur ON jeu.id_plat = ordinateur.id_plat
WHERE ordinateur.modele = 'Armstrad 6128'
```

Partie C

8. Alan doit utiliser la commande `ping`
9. On a $240 = (1111\ 0000)_2$. Dans le masque de sous-réseau, seuls les quatre derniers bits sont à 0, donc il y a $2^4 = 16$ adresses disponibles dans le sous-réseau. En enlevant les deux adresses réservées (adresse réseau et adresse de diffusion), l'adresse du routeur, et les deux adresses de Alan et Grace. Il reste donc $16 - 5 = 11$ adresses disponibles. Il est donc encore possible d'ajouter 11 machines à ce réseau local.
10. On a la table de routage suivante :

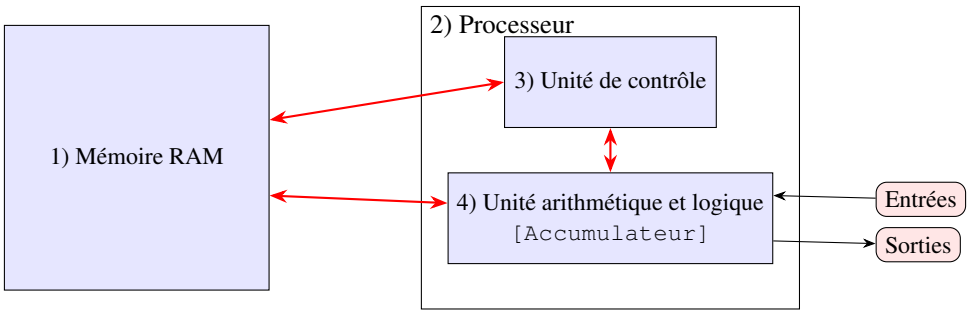
Routeur R8		
Destination	Routeur suivant	Nombre de sauts
R1	R1	3
R2	R4	2
R3	R4	3
R4	R4	1
R5	R5	1
R6	R7	2
R7	R7	1

11. Le routeur associé à l'ordinateur d'Alan est R8. Celui associé à l'ordinateur de Ada est R6. Ainsi, le chemin suivant convient : Alan - R8 - R7 - R6 - Ada
12. On a le chemin suivant pour un coût de 31 : R8 - R5 - R6 - R1 - R3

Exercice 3 (Architecture matérielle, files et POO)

Partie A

1. Il s'agit de von Neumann
2. On a le schéma suivant :



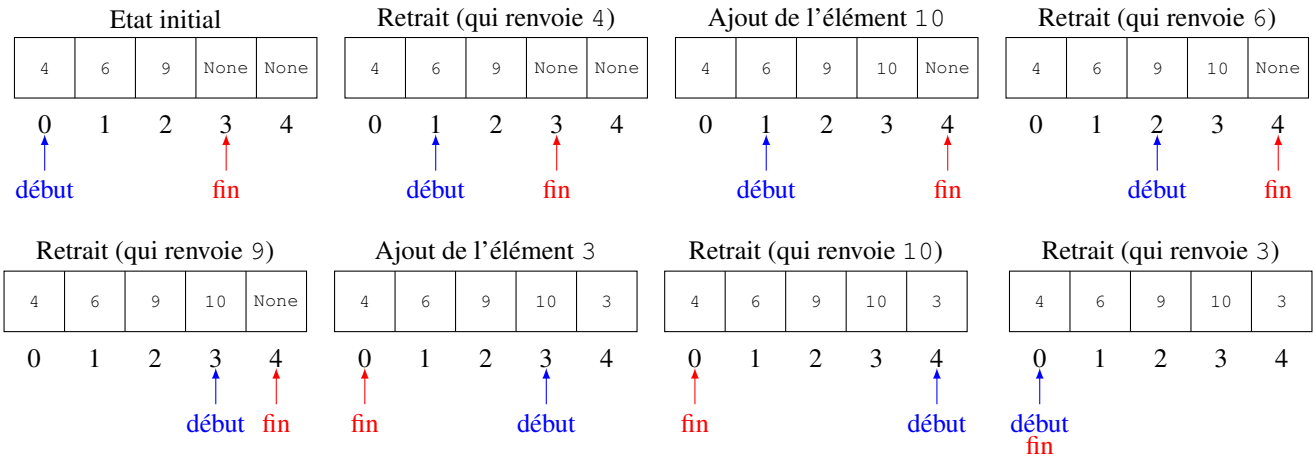
3. A chaque extinction de l'ordinateur, la mémoire RAM est perdue, c'est pour cela qu'on dit qu'elle est volatile.
4. De la plus rapide à la plus lente : Registre, Mémoire vive, Mémoire cache, Disque dur
5. L'instruction suivante convient : `sub r2, r5, r4`
6. Cette instruction permet de placer dans le registre `r1` la somme de la valeur du registre `r2` et de la valeur `0`. Ainsi, elle place dans le registre `r1` la valeur du registre `r2`
7. Les instructions suivantes conviennent :

```
addi r3 r1 0
addi r1 r2 0
addi r2 r3 0
```

8. On a A = 0, B = 1, C = 1, D = 1
9. On déduit du tableau précédent que le circuit est un additionneur binaire, il fait la somme de E1 + E2 + Re et donne le résultat (RsS), où Rs est le bit de retenue et S est le bit de même niveau que les entrées E1 et E2.

Partie B

10. Dans une file, le premier arrivé est le premier servi, en anglais on dit FIFO pour First in First out.
11. La valeur de `contenu` est `[None, None, None, None, None]`
12. On obtient les états successifs suivants :



13. L'attribut `contenu` a pour valeur `['o', 's', 'i', 'n', 'f']`
14. Le code suivant convient :

```
def est_vide(self):
    return self.debut = self.fin
```

15. La méthode suivante convient :

```
def retirer_element(self):
    elt = self.contenu[self.debut]
    self.debut = self.debut + 1
    if self.debut == self.capacite:
        self.debut = 0
    return elt
```

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.