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

25NSIJ1PO1

Exercice 1 (6 points)

Cet exercice porte sur les protocoles réseaux, l'algorithmique et la POO.

Partie A

1. On complète la table de routage de R_1 avec RIP.

Table de routage de R_1		
destination	prochain saut	distance
R_2	R_2	0
R_3	R_2	1
R_4	R_4	0
R_5	R_5	0
R_6	R_5	1

2. La route suivie par un paquet du LAN_1 à Internet est LAN_1-R_1-R_5-R_6-Internet avec un routage RIP.
3. On complète la table de routage de R_1 avec OSPF.

Table de routage de R_1		
destination	prochain saut	distance
R_2	R_2	10
R_3	R_2	11
R_4	R_2	12
R_5	R_2	13
R_6	R_2	14

4. La route suivie par un paquet du LAN_1 à Internet est LAN_1-R_1-R_2-R_3-R_4-R_5-R_6-Internet avec un routage OSPF.
5. Si R_2 tombe en panne, la route suivie par un paquet du LAN_1 à Internet est LAN_1-R_1-R_5-R_6-Internet avec un routage OSPF, de distance 101.

Partie B

6. On complète le tableau pour déterminer l'adresse réseau.

Calcul d'une adresse de réseau				
machine (binaire)	11000000	10101000	00000001	01100100
masque (binaire)	11111111	11000000	00000000	00000000
réseau (binaire)	11000000	10000000	00000000	00000000
réseau (déc.pointée)	192	128	0	0

7. On complète le tableau pour déterminer l'adresse de broadcast.

Calcul d'une adresse de broadcast				
réseau (binaire)	11000000	10000000	00000000	00000000
masque (binaire)	11111111	11000000	00000000	00000000
complément du masque (binaire)	00000000	00111111	11111111	11111111
broadcast (binaire)	11000000	10111111	11111111	11111111
réseau (déc.pointée)	192	191	255	255

8. Avec l'adresse 172.16.1.100/16 et le masque 255.255.0.0, on a l'adresse 172.16.0.0 du LAN_1, l'adresse de broadcast 172.16.255.255, et il y a 65534 adresses disponibles, de 172.16.0.1 à 172.16.255.254.
9. On complète la méthode `masquer`.

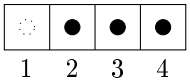
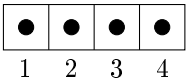
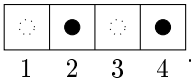
```
13          for i in range(4):
14              # Opération booléenne :
15              tmp.append(ip[i] & crible[i])
16          return ".".join([str(k) for k in tmp])
```

10. On complète la méthode `adresse_suivante`.

```
19                valeur, retenue = somme%256, somme//256
20                liste_suivante = [str(valeur)] + liste.suivante
```

Exercice 2 (6 points)

Cet exercice porte sur la programmation Python et la récursivité.

1. Avec la situation  on peut jouer 1 et obtenir  ou jouer 3 et obtenir .

2. On écrit le code de la fonction `initialiser`.

```
def initialiser(n):
    return [False] * n
```

3. On complète le code de la fonction `victoire`.

```
def victoire(tab):
    for etat_case in tab:
        if etat_case == False:
            return False
    return True
```

4. On écrit le code de la fonction `indice_premiere_case_occupee`.

```
def indice_premiere_case_occupee(tab):
    for i in range(len(tab)):
        if etat_case:
            return i
    return None
```

5. On écrit le code de la fonction `coup_valide`.

```
def coup_valide(tab, case):
    return case==0 or case in range(len(tab)) and case == indice_premiere_case_occupee(tab)+1
```

6. On écrit le code de la fonction `changer_case`.

```
def changer_case(tab, case):
    if coup_valide(tab, case):
        tab[case] = not tab[case]
    return tab
```

7. On complète le code de la fonction `vider`.

```
def vider(n):
    if n == 1:
        print('Vider case 1')
    elif n > 1:
        vider(n-2)
        print('Vider case '+str(n))
        remplir(n-2)
        vider(n-1)
```

8. On donne l'affichage produit par l'appel `vider(3)`.

```
>>> vider(3)
Vider case 1
Vider case 3
Remplir case 1
Vider case 2
Vider case 1
```

9. On écrit le code de la fonction `remplir`.

```
def remplir(n):
    if n == 1:
        print('Remplir case 1')
    elif n > 1:
        remplir(n-1)
        vider(n-2)
        print('Remplir case '+str(n))
        remplir(n-2)
```

10. Chacune des deux fonctions récursives `vider` et `remplir` s'appelle elle-même deux fois et l'autre une fois donc on a une complexité exponentielle, il est donc déraisonnable de vouloir traiter des baguenaudiers de grande taille.

Exercice 3 (8 points)

Cet exercice porte sur la programmation en Python, les bases de données relationnelles, le langage SQL, les systèmes d'exploitation et la sécurisation des communications.

Partie A

1. L'appel `gen_mdp(8, True, True, False)` convient.
2. On complète les lignes 8 à 10 de la fonction `gen_mdp`.

```
    minuscules = [chr(i) for i in range(97,123)]
    majuscules = [chr(i) for i in range(65,91)]
    caracteres_speciaux = [chr(i) for i in range(33,48)] + [chr(i) for i in range(58,65)]
```

3. On complète les lignes 13 et suivantes de la fonction `gen_mdp`.

```
    if cont_min:
        jeu_caracteres = jeu_caracteres + minuscules
    if cont_maj:
        jeu_caracteres = jeu_caracteres + majuscules
    if cont_spe:
        jeu_caracteres = jeu_caracteres + caracteres_speciaux
```

4. On complète la ligne 21 de la fonction `gen_mdp`.

```
    for i in range(longueur):
        mot_de_passe = mot_de_passe + jeu_caracteres[randint(0, len(jeu_caracteres)]
```

5. La fonction `gen_mdp` construit un mot de passe de longueur donnée en choisissant les caractères dans les catégories sélectionnées, mais ne garantit pas que des caractères de chaque catégorie seront utilisés.

Partie B

6. L'attribut `mot_de_passe` de la relation `compte` sert de clé primaire et doit donc être unique, ce qui empêche d'utiliser plusieurs fois le même mot de passe.
7. La requête `SELECT url FROM site;` convient.
8. La requête `UPDATE compte SET mot_de_passe = 'yhTS?d@UTJe' WHERE mot_de_passe = '@rDfohpj!&';` effectue la mise à jour.
9. La requête `SELECT id_site FROM compte WHERE renouvellement < '2024-03-20';` convient.
10. Le format AAAA-MM-JJ permet de comparer des dates en utilisant l'ordre lexicographique usuel.
11. La requête `SELECT utilisateur, mot_de_passe FROM compte JOIN site ON compte.id_site = site.id WHERE nom_site = 'Votremailp' ORDER BY renouvellement;` convient.
12. Le fait de séparer les données en plusieurs tables facilite les mises à jour et permet de limiter les incohérences ; p.ex on peut avoir plusieurs comptes sur un même site, et on pourra modifier l'url une seule fois dans la table `site` plutôt qu'une fois pour chacun des comptes.

Partie C

13. L'appel `chiffrement('/home/Documents/gestionnaire.db', '/home/Perso/secret.db', '/home/Perso/cle'` convient, avec des chemins absolus ; avec des chemins relatifs au répertoire courant, on peut utiliser. l'appel `chiffrement('gestionnaire.db', '../Perso/secret.db', '../Perso.cle')`.

14. On calcule `0xA3 ^ 0x59` en passant par le binaire.

hexa	binaire
A3	1010.0011
59	0101.1001
FA	1111.1010

15. Pour prouver l'identité `(a XOR b) XOR b = a`, on présente les quatre cas dans un tableau.

a	b	a XOR b	(a XOR b) XOR b
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

On constate que la première et la dernière colonne sont identiques, ce qui prouve la propriété.

16. Puisque la même clé permet de chiffrer et de déchiffrer le fichier, Alice utilise un chiffrement symétrique.

17. Les permissions sont `rw` (lecture et écriture) pour Alice, `r` (lecture seule) pour son groupe (le groupe eleves) et aussi `r` pour les autres. Cela signifie que n'importe qui peut lire le fichier `secret.db` ; Alice pourrait changer les permissions de ce fichier avec `chmod 600 secret.db` ou bien `chmod go-r secret.db`.

Cependant sans accès au fichier `cle`, l'attaque cryptographique est difficile puisque n'importe quel fichier de même taille que `secret.db` peut être reconstruit en choisissant bien le fichier `cle`.

Partie D

18. La préconisation P1 est suivie, comme on l'a vu à la question 1.

Pour P2, on a vu que la fonction `gen_mdp` de la partie A ne garantissait pas que chaque catégorie de caractère serait utilisée, on peut conseiller à Alice d'améliorer ce point.

La préconisation P3 sort du cadre de cet exercice ; on peut espérer qu'Alice, sensibilisée à la cybersécurité, évitera une telle erreur de débutant. Malgré tout, Alice stocke sa base de données en clair dans `/home/Documents/gestionnaire.db`, et on a vu question 17 qu'elle pouvait être négligente sur les permissions des fichiers. Clairement on a un problème ici.

La préconisation P4 est manifestement suivie, mais utiliser une solution libre déjà existante serait sans doute préférable, avec moins de risque de failles de sécurité que la solution « maison » d'Alice.

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.