

BACCALAURÉAT GÉNÉRAL

CORRECTION DE L'ÉPREUVE D'ENSEIGNEMENT DE SPÉCIALITÉ

SESSION 2024

NUMÉRIQUE ET SCIENCES INFORMATIQUES

Durée de l'épreuve : 3 heures 30

Le sujet est composé de trois exercices indépendants.

Exercice 1		6 points	
Questions	Contenu et notions	Capacités exigibles	Éléments de réponses et commentaires
1	Programmation élémentaire.	N1	<pre>def echange(tab, i, j): temp = tab[i] tab[i] = tab[j] tab[j] = temp</pre>
2	Programmation.	N2	<pre>triStooge(tab, i, j-k) triStooge(tab, i+k, j) triStooge(tab, i, j-k)</pre>
3	Récurtivité.	N1	Cet algorithme est récursif car la fonction <code>triStooge</code> s'appelle elle-même.
4	Calcul simple	N1	Lors du premier appel, $i = 0$ et $j = 5$ ainsi $k = (5-0+1)//3 = 2$. Il s'agit d'une division entière.
5	Analyser le fonctionnement d'un programme récursif.	N1	On compte 39 appels récursifs sans l'appel initial.
6	Analyser le fonctionnement d'un programme récursif.	N2	<pre>1 : triStooge(A,1,3) 2 : triStooge(A,2,3) 3 : triStooge(A,0,3) 0.5 par réponse juste</pre>
7	Analyser le fonctionnement d'un programme récursif.	N3	Voir le tableau 1. Enlever 0.25 par réponse fausse.
8	Connaissances des exemples du cours	N1	Tri rapide ou tri par insertion.

Tableau 1.

24-NSIJ2AN1C

Appel	Valeur de A avant l'appel	Valeur de A après l'appel
triStooge(A, 0, 3)	[5, 6, 4, 2]	[2, 4, 5, 6]
triStooge(A, 0, 2)	[2, 6, 4, 5]	[2, 4, 6, 5]
triStooge(A, 1, 3)	[2, 4, 6, 5]	[2, 4, 5, 6]
triStooge(A, 0, 2)	[2, 4, 5, 6]	[2, 4, 5, 6]

Exercice 2		6 points	
Questions	Contenu et notions	Capacités exigibles / Niveau	Éléments de réponses et commentaires
1	requête SQL affichant les noms et prénoms de tous les clients par ordre alphabétique des noms	N1	Dufour Marc Martin Sophie
2	requête SQL affichant les noms de tous les médicaments dont le prix est inférieur à 3 euros	N1	<pre>SELECT nom_medic, FROM medicament WHERE prix <3;</pre>
3	requête SQL pour l'ajout les informations de la carte vitale	N1	<pre>INSERT INTO client (3, 'Durand', 'Nathalie', '269054958815780');</pre>
4	les clés étrangères de la relation ordonnance et leur utilité.	N2	id_client et id_medic contrainte d'intégrité de référence : permet de faire le lien avec les autres tables
5	lignes 7 et 8 de la table le nombre de boites prescrites	N1	ligne 7 : 1 boite, ligne 8 : 3 boites
6	requête SQL mettant à jour la quantité du médicament Acide ascorbique en stock	N1	<pre>UPDATE 'medicament' SET quantite = quantite-3 WHERE id_medic=4</pre>
7	coût total des médicaments fournis à Madame Martin	N1	$1 \times 3,50 + 3 \times 5,50 = 20,00$ euros

Exercice 2	6 points		
8	requête SQL permettant d'afficher le nom du médicament pour l'ordonnance ayant l'id_ordo numero 6	N3	<pre>SELECT nom_medic FROM medicament JOIN ordonnance ON medicament.id_medic = ordonnance.id_medic WHERE ordonnance.id_ordo = 6</pre>

Exercice 3		8 points	
Questions	Contenu et notions	Capacités exigibles	Éléments de réponses et commentaires
1	Architecture d'un réseau / Transmission de données sur un réseau (première)	N1	l'adresse doit être unique et non réservée. Elle doit donc être choisie dans l'intervalle [192.168.1.3, 192.168.1.254]. En effet l'adresse 192.168.1.0 (adresse réseau) et 192.168.1.255 (adresse de diffusion) sont réservées et les adresses 192.168.1.1 et 192.168.1.2 sont déjà utilisées
2	Structure de données/spécifier une structure de donnée par son interface	N1	[Transaction("Alice", "Charlie", 10), Transaction("Bob", "Alice", 5)]
3	Vocabulaire de la programmation objet : attribut	N1	Deux explications possibles (une seule explication suffit à obtenir tous les points) : Premièrement le bloc0 étant le premier bloc de la Blockchain, il n'existe pas de bloc précédent. C'est pourquoi l'attribut <code>bloc_precedent</code> prend la valeur <code>None</code> qui veut dire "aucun". Deuxièmement, on remarque que le code de la méthode <code>creer_bloc_0</code> de la classe <code>Blockchain</code> renvoie un bloc avec un attribut <code>bloc_precedent</code> égal à <code>None</code>
4	liste : structure linéaire	N1	bloc0
5	Vocabulaire de la programmation objet : accéder aux attributs et méthodes d'une classe	N2	<pre>ma_blockchain = Blockchain() bloc1 = Bloc([Transaction("Alice", "Charlie", 50), Transaction("Charlie", "Bob", 30)]) bloc2 = Bloc([Transaction("Bob", "Charlie", 20), Transaction("Bob", "Charlie", 20)]) ma_blockchain.tete = bloc2</pre>
6	liste : structure linéaire	N1	Bob possède $100 + 30 - 20 - 20 = 90$ nsicoin.
7	Structure de données : implémentation	N2	<pre>def ajouter_bloc(self, liste_transactions): nouveau_bloc = Bloc(liste_transactions, self.tete) self.tete = nouveau_bloc</pre>
8	Transmission de données dans un réseau (programme de première)	N1	Comme il doit l'envoyer à tous les autres utilisateurs, il doit utiliser l'adresse de diffusion (ou broadcast) : 192.168.1.255

Exercice 3		8 points	
9	Récurtivité / Structure de données : implémentation	N2	<pre>def calculer_solde(self, utilisateur): if self.bloc_precedent is None : solde = 100 else : solde = self.bloc_precedent.calculer_solde(utilisateur) for transaction in self.liste_transactions : if transaction.expediteur == utilisateur : solde = solde - **transaction.montant elif transaction.destinataire == utilisateur : solde = solde + transaction.montant return solde</pre>
10	Vocabulaire de la programmation objet : accéder aux attributs et méthodes d'une classe	N2	<pre>ma_blockchain.tete.calculer_solde("Alice")</pre>
11	Construction élémentaire : boucles non bornées	N2	Comme il est impossible de déduire le bloc (donc le nonce) à partir du hash, il faut utiliser une boucle pour essayer de nombreuses valeurs possibles pour le nonce jusqu'à en trouver une satisfaisant les propriétés souhaitées.
12	Vocabulaire de la programmation objet / Construction élémentaire : affectation conditionnelle et appel de fonction	N2	Comme le bloc0 a son attribut <code>bloc_precedent</code> égal à <code>None</code> , l'analyse du code de la méthode <code>donner_hash_precedent</code> qui calcule la valeur de l'attribut <code>hash_bloc_precedent</code> montre que celui-ci a pour valeur 0.
13	Ecriture d'un entier positif dans une base b >= 2 (programme de première)	N2	2^{256}
14	Vocabulaire de la programmation objet / Construction élémentaire : boucles non bornées	N3	<pre>def minage_bloc(self): """modifie le nonce d'un bloc pour que son hash commence par '00'""" self.nonc = 0 while self.hash[0] != "0" or self.hash[1] != "0" : self.nonc = self.nonc + 1 self.hash = self.calculer_hash()</pre>

Exercice 3 8 points
