

BACCALAURÉAT GÉNÉRAL

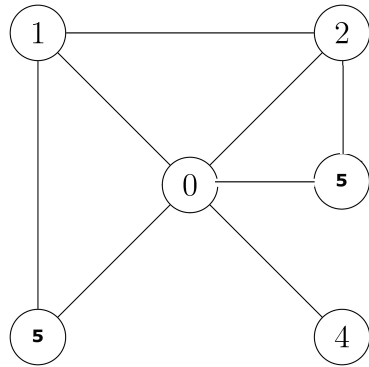
CORRECTION DE L'ÉPREUVE D'ENSEIGNEMENT DE SPÉCIALITÉ

SESSION 2024

NUMÉRIQUE ET SCIENCES INFORMATIQUES

Durée de l'épreuve : 3 heures 30

Le sujet est composé de trois exercices indépendants.

Questions	Contenu et notions	Capacités exigibles / Niveau	Éléments de réponses et commentaires
1	Modéliser des situations sous forme de graphes. Écrire les implémentations correspondantes d'un graphe : liste de successeurs.	N1	<pre>voisins = [[1,2,3,4], [0,2,3], [0,1], [0,1], [0]]</pre>  Voisins.
2	Modéliser des situations sous forme de graphes.	N1	
3	Écrire les implémentations correspondantes d'un graphe : liste de successeurs.	N1	<pre>voisins = [[1,2,3,4,5], [0,2,3], [0,1,5], [0,1], [0], [0,2]]</pre>
4	Modéliser des situations sous forme de graphes. Écrire les implémentations correspondantes d'un graphe : liste de successeurs.	N2	<pre>def voisin_alea(voisins, s): vs = voisins[s] return vs[random.randrange(len(vs))]</pre>

Exercice 1	6 points		
5	Analyser le fonctionnement d'un programme récursif.	N2	La fonction fait appelle à elle-même.
6	Analyser le fonctionnement d'un programme récursif. Modéliser des situations sous forme de graphes.	N2	Cette fonction simule les déplacements d'un virus commençant au sommet i durant n étapes et renvoie le sommet auquel il se trouve à la fin.
7	Modéliser des situations sous forme de graphes. ; Écrire un algorithme de calcul d'une moyenne. (1ère)	N3	<pre>def simule(voisins, i, n_tests, n_pas): results = [0] * len(voisins) for k in range(n_tests): j = marche_alea(voisins, i, n_pas) results[j] = results[j] + 1 return [x/n_tests for x in results]</pre>
8	Modéliser des situations sous forme de graphes.	N1	la priorité doit être mise sur l'ordinateur 0.
9	Modéliser des situations sous forme de graphes. ; Algorithmes sur les graphes.	N3	On fait un parcours en largeur à partir de s. On stocke dans un tableau l'état des sommets : au départ, seul le sommet s est contaminé. On fait une boucle while avec un compteur qui s'arrête lorsque tous les sommets sont contaminés. À chaque itération, on contamine les voisins de tous les sommets contaminés et on incrémente le compteur. On peut être un peu plus efficace en conservant uniquement les sommets découverts à l'itération précédente et/ou en utilisant une file. <pre>def virus(voisins, s): contamine = [False] * len(voisins) contamine[s] = True cpt = 0 while False in contamine: cpt = cpt + 1 contamine_copy = contamine.copy() for i in range(len(voisins)): if contamine[i]: for v in voisins[i]: contamine_copy[v] = True contamine = contamine_copy return cpt</pre>

Exercice 2	6 points																						
Questions	Contenu et notions	Capacités exigibles	Éléments de réponses et commentaires																				
1	Architecture d'un réseau	N1	255.255.0.0																				
2	Architecture d'un réseau	N1	172.16.0.0																				
3	Architecture d'un réseau	N1	172.16.255.255																				
4	Architecture d'un réseau	N1	$2^{16} - 2 = 65534$. On acceptera la réponse $2^{16} - 3$ en considérant l'adresse de la passerelle comme déjà utilisée.																				
5	Protocoles de routage.	N2	A-H-D																				
6	Protocoles de routage.	N2	Deux chemins possibles : A-B-C-D ou A-H-C-D																				
7	Protocoles de routage.	N3	Si le chemin A-B-C-D est choisi alors la règle de routage est obligatoirement modifiée pour les routeurs A et H : <table border="1"> <tr> <th>Routeur</th><th>Réseau destinataire</th><th>Passerelle</th><th>Interface</th></tr> <tr> <td>A</td><td>L2</td><td>193.55.24.1</td><td>193.55.24.2</td></tr> <tr> <td>H</td><td>L2</td><td>53.10.10.10</td><td>53.10.10.9</td></tr> </table> Figure 1. ou Si le chemin A-H-C-D est choisi alors la règle de routage est obligatoirement modifiée uniquement pour le routeur H : <table border="1"> <tr> <th>Routeur</th><th>Réseau destinataire</th><th>Passerelle</th><th>Interface</th></tr> <tr> <td>H</td><td>L2</td><td>53.10.10.10</td><td>53.10.10.9</td></tr> </table> Figure 2.	Routeur	Réseau destinataire	Passerelle	Interface	A	L2	193.55.24.1	193.55.24.2	H	L2	53.10.10.10	53.10.10.9	Routeur	Réseau destinataire	Passerelle	Interface	H	L2	53.10.10.10	53.10.10.9
Routeur	Réseau destinataire	Passerelle	Interface																				
A	L2	193.55.24.1	193.55.24.2																				
H	L2	53.10.10.10	53.10.10.9																				
Routeur	Réseau destinataire	Passerelle	Interface																				
H	L2	53.10.10.10	53.10.10.9																				
8	Protocoles de routage.	N2	Pour 100 Mbit/s coût C = 10 Pour 1 Gbit/s coût C = 1 Pour 10 Gbit/s coût = 0,1																				
9	Protocoles de routage.	N2	Chemin A-G-F-E-D pour un coût de 1,3.																				
10	Protocoles de routage.	N3	Le nouveau chemin est A-H-F-E-D pour un coût de 2,2.																				

Exercice 3		8 points										
Questions	Contenu et notions	Capacités exigibles / Niveau	Éléments de réponses et commentaires									
1	Modèle relationnel : identifier les concepts définissant le modèle relationnel Analyser et modéliser un problème	N1	Forte cohérence des données, Réduire les redondances, Faciliter la maintenabilité, langage de requête basé sur théorie des ensembles, gérer des volumes important de données, accès multi-utilisateur ...									
2	Modèle relationnel clef primaire. Analyser et modéliser un problème	N1	Cet attribut doit posséder la propriété d'unicité, il permet identifier de manière unique, sans aucune ambiguïté, chaque t-uplet de la relation.									
3	Modèle relationnel clef étrangère Analyser et modéliser un problème	N1	Les clés étrangères <code>id_client</code> , <code>id_emplacement</code> font référence respectivement aux clés primaires <code>id_client</code> de la relation <code>Client</code> et <code>id_emplacement</code> de la relation <code>Emplacement</code> . Elles permettent de relier la relation <code>Reservation</code> avec les relations <code>Client</code> et <code>Emplacement</code> .									
4	Modèle relationnel clef primaire, schéma relationnel Analyser et modéliser un problème	N1	<code>Emplacement (id_emplacement, nom, localisation, tarif_journalier)</code> la clé primaire est <code>id_emplacement</code> de type entier, les attributs <code>localisation</code> et <code>nom</code> sont de type chaîne de caractères et <code>tarif_journalier</code> est de type float.									
5	Langage SQL : requêtes d'interrogation, identifier les composants d'une requête. Mobiliser les concepts et les technologies utiles	N1	<table><tr><td>1</td><td>myrtille</td><td>A4</td></tr><tr><td>4</td><td>mandarine</td><td>B1</td></tr><tr><td>6</td><td>melon</td><td>A2</td></tr></table>	1	myrtille	A4	4	mandarine	B1	6	melon	A2
1	myrtille	A4										
4	mandarine	B1										
6	melon	A2										
6	Langage SQL : requêtes d'interrogation. Mobiliser les concepts et les technologies utiles	N1	<pre>SELECT nom, prenom FROM Client WHERE ville = 'Strasbourg'; ou SELECT nom, prenom FROM Client WHERE ville LIKE 'Strasbourg';</pre>									

Exercice 3		8 points	
7	Langage SQL : Construire des requêtes d'insertion. Mobiliser les concepts et les technologies utiles	N1	<pre>INSERT INTO `Client` (`id_client`,`nom`, `prenom`, `adresse`, `ville`, `pays`, `telephone`) VALUES (42, 'Codd','Edgar','28 rue des Capucines', 'Lyon', 'France', '0555555555');</pre>
8	Langage SQL : Construire des requêtes d'interrogation. Mobiliser les concepts et les technologies utiles	N2	<pre>SELECT Client.nom, Client.prenom, Client.adresse, Client.ville, Client.pays, Client.telephone, Reservation.id_reservation,Reservation.nombre_personne, Reservation.date_arrivee, Reservation.date_depart, Emplacement.nom, Emplacement.tarif_journalier FROM Reservation JOIN Client ON Reservation.id_client = Client.id_client JOIN Emplacement ON Reservation.id_emplacement = Emplacement.id_emplacement WHERE Reservation.id_reservation = 18;</pre>
9	Vocabulaire de la programmation objet : classes, attributs, méthodes, objets. Mobiliser les concepts et les technologies utiles	N1	Le terme self permet de faire référence à l'objet lui-même.
10	Vocabulaire de la programmation objet : classes, attributs, méthodes, objets. Mobiliser les concepts et les technologies utiles	N1	<pre>client01 = Client('Codd', '28 rue des capucines', 'Lyon', 'France', '0555555555')</pre>
11	Vocabulaire de la programmation objet : classes, attributs, méthodes, objets. Traduire un algorithme dans un langage de programmation	N1	<pre>return (emplacement.tarif_journalier + reservation.nombre_personne * 2.2) * reservation.nb_jours() ou return emplacement.tarif_journalier * reservation.nb_jours() + reservation.nombre_personne * 2.2 * reservation.nb_jours()</pre>

Exercice 3		8 points	
12	Mise au point de programmes / Constructions élémentaires	N2	Erreur ligne 25 <code>not(2018 <= annee <= 2024)</code> erreur de typage on ne peut pas comparer un str avec un int.
13	Mise au point de programmes / Constructions élémentaires	N2	Correction utilisation de la fonction <code>int</code> pour utiliser un entier dans la comparaison : <code>not(2018 <= int(annee) <= 2024)</code>
14	Mise au point de programmes / Constructions élémentaires	N2	<pre> if not(mois in calendrier): return False if not(que_des_chiffre(numero)) or not(len(numero) == 3): return False </pre> une réponse avec trois if successifs est valide : <pre> if not(mois in calendrier): return False if not(que_des_chiffre(numero)): return False if not(len(numero) == 3): return False </pre>