

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		
<i>Questions</i>	<i>Contenu et notions</i>	<i>Capacités exigibles / Niveau</i>	<i>Éléments de réponses et commentaires</i>
1	base de données relationnelle	N1	int ou char
2	SQL	N1	5
3	base de données relationnelle	N1	S'il est différent pour chaque agence
4	base de données relationnelle	N2	<u>#id_agence,#id_voiture</u> ou <u>id_asso,#id_agence,#id_voiture</u>
5	SQL	N2	<pre>INSERT INTO couple_voitures_agences (id_agence, id_voiture) VALUES (5, 2);</pre> ou <pre>INSERT INTO couple_voitures_agences (id,id_agence, id_voiture) VALUES (1,5, 2);</pre>
6	SQL	N2	<pre>UPDATE couple_voitures_agences SET id_agence = 2 WHERE id_voiture = 2</pre>
7	SQL	N3	<pre>SELECT `type`,`marque`,`Agence` FROM Voiture,Agence INNER JOIN couple_voitures_agences ON couple_voitures_agences.id_agence = Agence.id_agence WHERE couple_voitures_agences.id_voiture = Voiture.id_voiture</pre>
8	python	N2	<pre>1 def execute_requete_insert_voiture(liste,id_agence): 2 ... 3 ...</pre>
9	base de données relationnelle;python	N3	Il faut vérifier que id_agence existe et que tous les attributs sont bien passés en paramètre avec les bons types.

Exercice 1 6 points

Exercice 2		6 points	
Questions	Contenu et notions	Capacités exigibles / Niveau	Éléments de réponses et commentaires
1	adresses IP, réseau	N1	192.168.1.X avec $2 \leq X \leq 9$ ou $11 \leq X \leq 254$
2	adresse réseau, adresses réservées	0,75	$2^{16} - 2 = 65534$
3	modélisation graphe : dictionnaire	N1	code : $G = \{ 'R1': ['R2', 'R3', 'R4', 'R6'], 'R2': ['R1', 'R3', 'R5'], 'R3': ['R1', 'R2', 'R5', 'R6'], 'R4': ['R1', 'R6'], 'R5': ['R2', 'R3', 'R6', 'R7'], 'R6': ['R1', 'R3', 'R4', 'R5', 'R7', 'R8'], 'R7': ['R5', 'R6', 'R8', 'R9'], 'R8': ['R6', 'R7', 'R9'], 'R9': ['R7', 'R8'] \}$
4	table routage, protocole RIP	N1	Voir le tableau 1.
5	chemin parcouru par un paquet, protocole RIP	N1	<code>distance(G, 'R1')['R9']</code> vaut 3 donc en 3 sauts : R1, R6, R8, R9 (ou R1, R6, R7, R9)
6	modélisation graphe par matrice d'adjacence : liste de listes	N1	code : $M = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$
7	compléter une fonction : parcours matrice (boucles imbriquées)	N3	code : <pre>def degre(MATRICE): d=[] for ligne in MATRICE: cpt=0 for elt in ligne: cpt=cpt+elt d.append(cpt) return d</pre>
8	donner le degré d'un sommet	N1	[4, 3, 4, 2, 4, 6, 4, 3, 2]

Exercice 2	6 points		
9	parcours d'un graphe en utilisant le résultat d'une fonction	N2	2 sommets de degré impair donc le graphe admet une chaîne eulérienne
10	parcours d'un graphe, plus court chemin, protocole OFPS, conversion débit/coût	N3	$BP = 100 \text{ Mb/s} = 100 \times 10^6 \text{ b/s} \rightarrow \text{coût} = \frac{10^8}{100 \times 10^6} = 1$ $BP = 50 \text{ Mb/s} = 50 \times 10^6 \text{ b/s} \rightarrow \text{coût} = \frac{10^8}{50 \times 10^6} = 2$ $BP = 100 \text{ Mb/s} = 10 \times 10^6 \text{ b/s} \rightarrow \text{coût} = \frac{10^8}{10 \times 10^6} = 10$ Voir le tableau 2. R1, R2, R5, R6, R7, R9 coût = 6

Tableau 1

Destination	Suivant	Nombre de sauts
R2	R2	1
R3	R3	1
R4	R4	1
R5	R2 OU R3	2
R6	R6	1
R7	R6	2
R8	R6	2
R9	R6	3

Tableau 2

R1	R2	R3	R4	R5	R6	R7	R8	R9
0	inf	inf	inf	inf	inf	inf	inf	inf
	1 (R1)	2 (R1)	2 (R1)	inf	10 (R1)	inf	inf	inf
		2 (R1)	2 (R1)	2 (R2)	10 (R1)	inf	inf	inf
		2 (R1)		2 (R2)	4 (R4)	inf	inf	inf
				2 (R2)	4 (R4)	inf	inf	inf
					3 (R5)	12 (R5)	inf	inf
						4 (R6)	13 (R6)	inf
								6 (R7)

Exercice 3		8 points	
Questions	Contenu et notions	Capacités exigibles / Niveau	Éléments de réponses et commentaires
1	numération	N1	Voir le tableau 1.
2	numération	N1	$8 \times 20^2 + 11 \times 20^1 + 15 \times 20^0 = 3200 + 220 + 15 = 3435$
3	POO : créer une instance de classe	N1	instructions : M=Maya() M.ajouter([0,0,3]) M.ajouter([0,1,2]) M.ajouter([0,3,1])
4	POO: longueur pile	N1	code : def nbEtage(self): return len(self.nombre)
5	python : accès aux éléments d'une liste	N1	code : def valeurChiffre(L): return L[1]+5*L[2]
6	algorithme conversion de la base 20 vers la base 10, POO, PILE, appel de fonctions	N2	code : def MayaToDec(self): coeff=20** (self.nbEtages()-1) ch_Dec=0 while not self.estVide() : ch_Maya = self.retirer() ch_Dec=ch_Dec+(valeurChiffre(ch_Maya))*coeff coeff = coeff//20 return ch_Dec
7	listes, boucle WHILE, arrêt	N2	code : def decompChiffre(n): if n==0: return [1,0,0] L=[0,0,0] while n>=5: L[2]+=1 n=n-5 L[1]=n return L

Exercice 3		8 points	
8	POO, appel de fonctions, pile, boucle FOR	N2	code : <pre>def DecToMaya(n): M=Maya() for chiffre in DecToVige(n): M.ajouter(decompChiffre(chiffre)) return M</pre>
9	multiplier par la base, accès aux attributs	N2	code : <pre>def multiplie(self): M=Maya() M.nombre=[[1,0,0]]+self.nombre return M</pre>
10	Analyser une fonction	N2	<pre>mystere([0,1,1],[0,3,1],0) renvoie ([0,4,2],0) mystere([0,1,1],[0,4,2],0) renvoie ([1, 0, 0], 1)</pre>
11	Ecrire une fonction complexe	N3	code : <pre>def somme(self,maya2): if self.nbEtages == maya2.nbEtages: return None r=0 res=Maya() for i in range(self.nbEtages()): s,r=mystere(self.nombre[i],maya2.nombre[i],r) res.ajouter(s) if r==1: res.ajouter([0,1,0]) return res</pre>

Tableau 1

Etage	Ecriture Maya	Valeur du chiffre de l'étage	Valeur décimale dans la conversion
3		$1 \times 5 + 3 \times 1 = 8$	$8 \times 20^2 = 3200$
2		$2 \times 5 + 1 \times 1 = 11$	$11 \times 20 = 220$
1		$3 \times 5 = 15$	15