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 / Niveau*	*Éléments de réponses et commentaires*
1	Savoir dénombrer les adresses IP disponibles sur un réseau en fonction du masque de sous-réseau.	Simuler ou mettre en oeuvre un réseau /1	Avec un tel masque, on ne peut pas obtenir plus de 256 adresses IP sur un sous-réseau, dont 2 sont réservées en plus ce qui donne un total de 254 adresses disponibles.
2	Conversion décimal à binaire	Passer de la représentation d'une base à une autre /1	11011001
3	Conversion binaire à décimal	Passer de la représentation d'une base à une autre /1	50
4	Déterminer si une machine fait partie d'un réseau en utilisant son adresse IP et celle du réseau.	Simuler ou mettre en oeuvre un réseau /2	Non car avec le masque utilisé, les adresses doivent toutes commencer par 110.217.52.
5	Remplir une table de routage simple en indiquant les adresses IP et les interfaces.	Simuler ou mettre en oeuvre un réseau /2	Voir le tableau 1.

Exercice 1	6 points		
6	Remplir une table de routage simple et tenir compte du protocole RIP	Identifier, suivant le protocole de routage utilisé, la route empruntée par un paquet /1	Sur la sdeuxième ligne de la table de routage, la passerelle est modifée en 110.217.50.253 et l'interface en 110.217.54.253.
7	Vérifier si le protocole RIP est bien respecté.	Identifier, suivant le protocole de routage utilisé, la route empruntée par un paquet /1	Aucun changement car depuis R2, les routes menant aux réseaux P1 et P2 ont le même nombre de routeurs.
8	Comprendre la notion de condition terminale d'un algorithme récursif.	Chercher un chemin dans un graphe /2 + analyser le fonction-nement d'un programme récursif /1	Problème : la fonction n'arrêtera pas de s'appeler car jamais la condition terminale R1==R2 ne sera satisfaite.
9	Connaître un algorithme de parcours de graphe	Chercher un chemin dans un graphe /3	La solution classique est de marquer les sommets par lesquels l'algorithme passe, en les stockant par exemple dans une liste et en vérifiant que le sommet S n'est pas marqué avant de lancer l'appel recherche(S,R2).

Tableau 1 :

Destination	Passerelle	Interface
110.217.50.0	on-link	110.217.50.254
110.217.52.0	110.217.54.253	110.217.54.254
110.217.54.0	on-link	110.217.54.254
110.217.56.0	110.217.54.253	110.217.54.254

Exercice 2 — 6 points

Questions	*Contenu et notions*	*Capacités exigibles / Niveau*	*Éléments de réponses et commentaires*
1	langage	N1	`def possible_avec_penalites_seules(score):` `  return score % 3 == 0`
2	algorithmique	N1	0 –> [0] ; 1 1 –> [] ; 0 2 –> [] ; 0 3 –> [0, 3] ; 1 4 –> [] ; 0 5 –> [0, 5] ; 1 6 –> [0, 3, 6] ; 1 7 –> [0, 7] ; 1 8 –> [0, 5, 8], [0, 3, 8] ; 2 9 –> [0, 3, 6, 9] ; 1 10 –> [0, 3, 10], [0, 5, 10], [0, 7, 10]; 3
3	langage	N1	f(10) = f(7) + f(5) + f(3) = (f(4) + f(2) + f(0)) + f(5) + f(3) = 3.
4	algorithmique	N1	Les cases de base sont n = 1, 2 et 4 pour lesquels la fontion doit renvoyer 0 et n = 0, 3, 5 et 6 pour lesquels la fonctions doit renvoyer 1.

5	algorithmique	N2	`def nb_solutions(score):`
			`  if score in [1, 2, 4]:`
			`  resultat = 0`
			`  elif score in [0, 3, 5, 6]:`
			`  resultat = 1`
			`  else:`
			`  resultat = nb_solutions(score - 3) + nb_solutions(score - 5) + nb_solu`
			`  return resultat`

6	algorithmique ; récursivité	N3	On peut utiliser la programmation dynamique, en utilisant une structure de données pour stocker les valeurs déjà calculées.
7	récursivité, conception	N3	On utilise les lignes correspondant aux scores 8, 6 et 4, en complétant les listes présentes.

Exercice 2	6 points		
8	récursivité, conception	N3	

```
def solutions_possibles(score):
    if score < 0:
        resultat = []
    elif score == 0:
        resultat = [[0]]
    else:
        resultat = []
        for coup in [3, 5, 7]:
            liste = solutions_possibles(score - coup)
            for solution in liste:
                solution.append(score)
            resultat.append(solution)
  return resultat
```

Exercice 3 — 8 points

Questions	*Contenu et notions*	*Capacités exigibles / Niveau*	*Éléments de réponses et commentaires*
1	Cours dictionnaire	N1	`python participants['PHILIPSEN Jasper'] classement_general[Participants['PHILIPSEN Jasper']] temps_etapes[Participants['PHILIPSEN Jasper']][3]`
2.	programmation Python	N2	`python def temps_totale(d): temps_total = 0 for temps in temps_etapes[d]: temps_total += temps return T`
3.	programmation Python	N3	`8 while pos>= 0 and element[1] < classement[pos][1]:` `9 classement[pos + 1] = classement[pos]` `14 classement_general[classement[i][0]] = i + 1`
4	programmation Python - Langage SQL	N2	```` ```python 5 tableau_final = [] 6 difference_temps = 0 7 premier = True 8 for ligne in tableau_temps: 9 coureur = [ligne[0]] 10 coureur.append(ligne[1]) 11 if premier: 12 temps_premier = ligne[2] 13 coureur.append(temps_premier) 14 premier = False 15 else: 16 difference_temps = ligne[2] - temps_premier 17 coureur.append(difference_temps) 18 tableau_final.append(coureur) ``` ````
5	BDD - Modèle relationnel	N1	Les mêmes numéros de dossard apparaîssent à chaque étape. Un même numéro d'étape apparaît pour chaque coureur. Par contre, un coureur ne participe qu'une seule fois à chaque étape. Chaque valeur du couple `(#numDossard, #NumEtape)` est unique. Celui-ci peut donc servir de clef primaire.
6	BDD - Langage SQL	N1	La requête donne l'ensemble des noms des coureurs de l'équipe Cofidis.

Exercice 3	8 points		
7	BDD - Langage SQL	N1	`sql SELECT Date FROM Etapes WHERE Type = 'contre-la-montre';`
8	BDD - Langage SQL	N1	`sql SELECT DirecteurSportif FROM Equipes JOIN Coureurs ON Equipes.nomEquipe = Coureurs.Equipes WHERE Coureurs.nomCoureur = 'BARDET Romain';`
9	BDD - Modèle relationnel	N1	la série de code SQL provoque une erreur car il faut que la valeur de l'attribut `numEtape` existe dans la table `Etapes` avant de pouvoir être utiliser dans la table `Temps`.
10	BDD - Langage SQL	N2	Pour corriger cette erreur, il suffit d'inverser les deux instructions SQL.
11	BDD - Langage SQL	N2	`sql SELECT sum(tempsRéalisé) From Temps JOIN Coureurs ON Coureurs.numDossard = Temps.numDossard WHERE Coureurs.nomCoureur = 'BARDET Romain';`