Mayotte - mars 2023 - sujet 1

Exercice 1 (Réseaux et protocoles de routage - 4 points)
Quelques rappels :

- Une adresse IPv4 est composée de quatre octets `X1.X2.X3.X4` qui peuvent être écrits en notation binaire ou décimale.
- La notation CIDR `X1.X2.X3.X4/n` signifie que les `n` premiers bits de poids forts de l'adresse IP représentent la partie « réseau », les bits suivants la partie « hôte » (machine).

1. (a) Donner le nombre de bits formant un octet.
   (b) Déterminer l'écriture décimale de l'adresse IPv4 correspondant à l'écriture binaire :

   `11000000.10101000.00000100.11110001`

2. On considère la machine d'adresse IPv4 `172.20.1.242/24`.
   (a) Donner la notation décimale du masque de sous-réseau de cette machine.
   (b) Donner l'adresse décimale de ce réseau.
   (c) Donner le nombre maximal de machines que l'on peut connecter sur ce réseau.
3. On considère le réseau représenté ci-dessous :

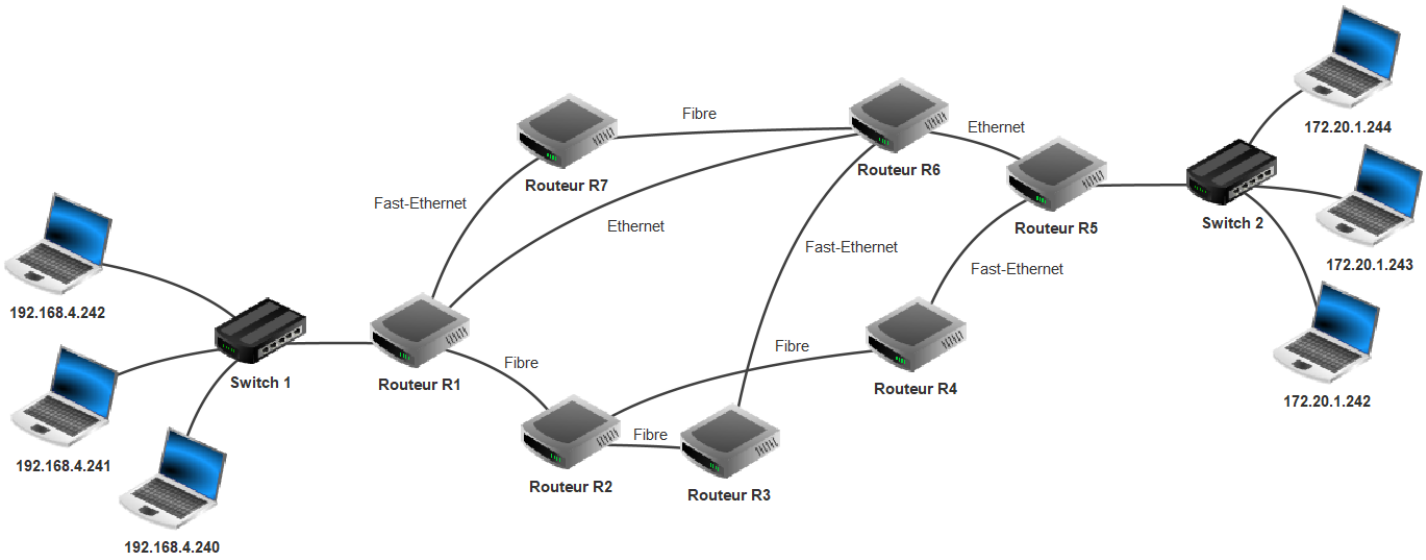

FIGURE 1 – Schéma du réseau

Le protocole RIP (Routing Information Protocol) est un protocole de routage qui cherche à minimiser le nombre de routeurs traversés (ce qui correspond à la distance ou au nombre de sauts).
Le réseau est composé de sept routeurs : R1, R2, R3, R4, R5, R6 et R7 et utilise le protocole RIP. Le routeur R1 doit transmettre des données au routeur R5.

(a) Déterminer le parcours pouvant être emprunté par ces données en vous aidant des tables de routage ci-dessous.

Table de routage de R1	
destination	passe par
R2	R2
R3	R6
R4	R2
R5	R6
R6	R6
R7	R7

Table de routage de R2	
destination	passe par
R1	R1
R3	R3
R4	R4
R5	R4
R6	R3
R7	R1

Table de routage de R3	
destination	passe par
R1	R6
R2	R2
R4	R2
R5	R6
R6	R6
R7	R6

Table de routage de R4	
destination	passe par
R1	R2
R2	R2
R3	R2
R5	R5
R6	R5
R7	R2

Table de routage de R5	
destination	passe par
R1	R6
R2	R4
R3	R6
R4	R4
R6	R6
R7	R6

Table de routage de R6	
destination	passe par
R1	R1
R2	R3
R3	R3
R4	R5
R5	R5
R7	R7

Table de routage de R7	
destination	passe par
R1	R1
R2	R1
R3	R6
R4	R7
R5	R6
R6	R6

Pour les deux questions suivantes, on suppose que la liaison entre R1 et R6 est coupée.

(b) Donner une nouvelle table de routage possible pour R1.

(c) En déduire le parcours que suivront les données pour aller du routeur R1 au routeur R5.

4. Pour la suite de l'exercice, on considère que la liaison entre R1 et R6 a été rétablie et on applique désormais le protocole de routage OSPF attribuant un coût à chaque liaison afin de trouver le chemin permettant une transmission plus rapide. Le coût d'une liaison est défini par la relation : coût $= \frac{10^8}{d}$, où d représente le débit en bit par seconde.

(a) Recopier et compléter le tableau suivant :

Liaison	**Débit**	**Coût**
Ethernet		10
Fast-Ethernet	10^8	
Fibre	10^9	0,1

(b) Reproduire la Figure 2, ci-dessous, en faisant apparaître le coût de chacune des liaisons.

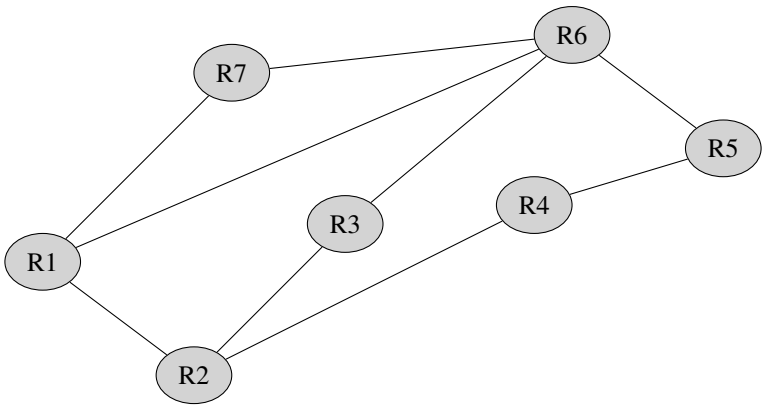

FIGURE 2 – Représentation du réseau

Le coût d'un chemin est la somme des coûts des liaisons empruntées.

(c) Donner les 6 chemins possibles ainsi que leur coût lors de l'envoi d'un paquet depuis le routeur R1 vers le routeur R5.

(d) Déduire, en respectant le protocole OSPF, le chemin le moins coûteux lors de l'envoi d'un paquet depuis le routeur R1 vers le routeur R5. Préciser le coût minimal.

Exercice 2 (POO et dictionnaires - 4 points)
Dans le tableau ci-dessous, on donne les caractéristiques nutritionnelles, pour une quantité de 100 grammes, de quelques aliments.

	Lait entier UHT	Farine de blé	Huile de tournesol
Énergie (kcal)	65,1	343	900
Protéines (grammes)	3,32	11,7	0
Glucides (grammes)	4,85	69,3	0
Lipides (grammes)	3,63	0,8	100

FIGURE 1 – Caractéristiques nutritionnelles

Pour chaque aliment, on souhaite stocker les informations dans un objet de la classe `Aliment` définie ci-dessous, où `e`, `p`, `g` et `l` sont de type `float` et désignent respectivement les quantités d'énergie, de protéines, de glucides et de lipides de l'aliment.

```
1 class Aliment:
2     def __init__(self, e, p, g, l):
3         self.energie = e
4         self.proteines = p
5         self.glucides = g
6         self.lipides = l
```

1. (a) Ecrire, à l'aide du tableau des caractéristiques nutritionnelles de la Figure 1, l'instruction en langage Python pour instancier l'objet `lait`.
   (b) Donner l'instruction qui permet d'obtenir la valeur `65.1` de l'objet `lait` instancié dans la question précédente.
   (c) Une erreur s'est introduite dans le tableau de la Figure 1 : la masse de protéines dans le lait est 3,4 au lieu de 3,32. Donner l'instruction qui modifie la masse de protéines de l'objet `lait` instancié dans la question 1.a.

On souhaite ajouter une méthode `energie_reelle` à la classe `Aliment` qui calcule l'énergie en kcal d'un aliment en fonction d'une masse donnée.
Par exemple :

- pour 245 grammes de lait, l'énergie réelle sera $245 \times 65{,}1 \div 100 = 159{,}495$ kcal ;
- l'instruction `lait.energie_reelle(245)` renvoie alors `159.495`

2. Recopier et compléter les lignes 1 et 2 dans la méthode ci-dessous.

```
1 def energie_reelle(.........., masse):
2     return .........
```

3. On regroupe les caractéristiques nutritionnelles du tableau de la Figure 1 dans le dictionnaire suivant, les clés étant des chaînes de caractères donnant le nom de l'aliment et les valeurs associées des objets de la classe `Aliment` :

```
nutrition = {'lait' : Aliment(65.1,3.4,4.85,3.63),
             'farine' : Aliment(343,11.7,69.3,0.8),
             'huile' : Aliment(900,0,0,100)}
```

   (a) Donner l'instruction qui permet d'obtenir la valeur énergétique en kcal du lait à partir des données de ce dictionnaire.
   (b) Donner l'instruction qui permet d'obtenir la valeur énergétique réelle de 220 grammes de lait à partir des données de ce dictionnaire.

4. Une recette de gâteau (sans oeuf) utilise les ingrédients suivants :
   - 230 g de farine ;
   - 220 g de lait ;
   - 100 g d'huile.

   Les quantités d'ingrédients, en grammes, sont regroupées dans le dictionnaire suivant :

```
recette_gateau = {'lait' : 220, 'farine' :230, 'huile':100}
```

   Ecrire, en utilisant la classe `Aliment` et la méthode `energie_reelle`, les instructions nécessaires pour calculer l'énergie réelle totale du gâteau.

Exercice 3 (Bases de données et SQL - 4 points)

L'énoncé de cet exercice utilise les mots-clefs du langage SQL suivants : `SELECT`, `FROM`, `WHERE`, `JOIN ... ON`, `UPDATE ... SET`, `INSERT INTO ... VALUES`.

Un orchestre souhaite créer une base de données relationnelle contenant la table de ses membres et celle des instruments joués. Pour cela, il est choisi d'utiliser le langage SQL. Une relation `musiciens` est créée. Elle contient également les informations d'inscription ainsi que leur ancienneté au sein de l'orchestre.
`id_mus` est la clé primaire de la relation `musiciens`.
`Code_instrument` est la clé primaire de la relation `instruments`.

id_mus	Nom	Prenom	Mail	Anciennete	Code_instrument
1	Dupont	Claudine	cDupont@mail.fr	5	1
2	Leclerc	Jean	leclerc.jean@mail.fr	2	2
3	M'bake	Doume	doubake@mail.fr	7	5
4	Descarte	Hugo	Hugo.arnaque@mail.fr	4	1
5	Tourelle	Charlène	char.tour@mail.fr	1	7
6	Dupuis	Alice	dupuis.Alice@mail.fr	9	4
7	Tourelle	Phillipe	filoutou@mail.fr	1	4
8	Fernandez	Kader	kader454@mail.fr	4	1
9	Dujardin	Laure	dujardin.laure@mail.fr	1	6
10	Lefevre	Marie	lefevre.marie@mail.fr	2	2

Extrait de la relation `musiciens`

Afin de compléter les informations personnelles, la relation `instruments` est créée :

Code_instrument	type_ins
1	Violon
2	Violon
3	Alto
4	Alto
5	Violoncelle
6	Contrebasse
7	Harpe

Extrait de la relation `instruments`

1. (a) Donner la clé étrangère de la relation `musiciens` en justifiant la réponse.
   (b) Ecrire le schéma relationnel des relations `musiciens` et `instruments`.
2. (a) Ecrire le résultat de la requête suivante :

```
SELECT Nom, Prenom FROM musiciens WHERE Code_instrument=4;
```

   (b) Ecrire une requête permettant d'afficher les noms et prénoms de tous les musiciens membres depuis 5 ans ou plus.
3. (a) Donner la requête permettant de modifier l'adresse mail de Tourelle Charlène par `tour.char@mail.fr`.
   (b) Ecrire la requête permettant d'afficher le nom et prénom de tous les musiciens violonistes à l'aide d'une jointure.
4. Afin de gérer la partie administrative, une nouvelle relation `Admin` est créée. Elle contient un intitulé du poste en clé primaire, l'identifiant du musicien associé et son ancienneté dans le poste.

Poste	id_mus	Anciennete_poste
Archiviste	8	1
Chef d'Orchestre	4	4
Président	6	7

Extrait de la relation `Admin`

   (a) Insérer le poste `Trésorier` qui vient d'être créé par l'orchestre et dont le rôle vient d'être attribué à Leclerc Jean dans la table `Admin`.
   (b) Ecrire la requête permettant d'afficher l'instrument joué par le président, en utilisant des jointures.