BACCALAURÉAT GÉNÉRAL

ÉPREUVE D'ENSEIGNEMENT DE SPÉCIALITÉ

SESSION 2023

NUMÉRIQUE ET SCIENCES INFORMATIQUES

JOUR 2

Durée de l'épreuve : **3 heures 30**

L'usage de la calculatrice n'est pas autorisé.

Dès que ce sujet vous est remis, assurez-vous qu'il est complet.
Ce sujet comporte 10 pages numérotées de 1/10 à 10/10.

Le candidat traite les 3 exercices proposés.

EXERCICE 1 (4 points)

Cet exercice porte sur les arbres binaires de recherche, la programmation orientée objet et la récursivité.

Le code Morse doit son nom à Samuel Morse, l'un des inventeurs du télégraphe.
Il a été conçu pour transférer rapidement des messages en utilisant une série de points et de tirets.
Pour cet exercice, les points seront représentés par le caractère "o" et les tirets par le caractère "-".

Chaque caractère du message que l'on veut transmettre est constitué d'une série de 1 à 5 points ou tirets. Le code a été conçu en tenant compte de la fréquence de chaque caractère dans la langue anglaise, de sorte que les caractères les plus fréquents, tels que E et T, ne comportent qu'un seul point ou tiret (E = "o", T = "-"), tandis que les caractères moins fréquents peuvent comporter 4 à 5 points ou tirets (par exemple, Q = "- - o -" et J = "o - - -").

Pour connaître le code morse de chaque caractère, on peut utiliser l'arbre binaire ci-dessous. En partant de la racine de l'arbre, la succession des branches reliant le nœud racine au caractère recherché nous donne le code morse de ce caractère en considérant que :

- une branche gauche correspond à un point ("o") ;
- une branche droite correspond à un tiret ("-").

Par exemple, le code morse de la lettre P est "**o - - o**" comme expliqué sur ce schéma :

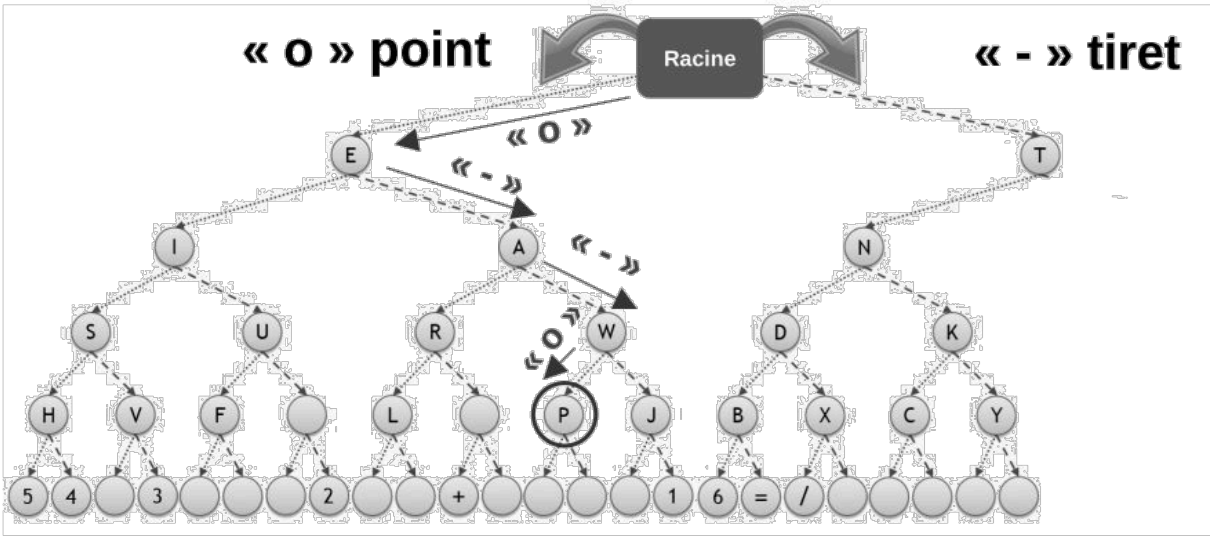


Figure 1 : Extrait de l'arbre binaire du code Morse

1. Déterminer le code morse du message "NSI", à l'aide de la figure de l'arbre binaire, en laissant un espace entre le code de chaque lettre.

2. Représenter le sous-arbre binaire pour les lettres M, G, O, Z et Q à l'aide de l'extrait de la table du code morse international :

G : --o **M :** -- **O :** --- **Q :** --o- **Z :** --oo

On donne, la déclaration de la classe et un extrait de la définition de l'arbre binaire :

```
1.  class Noeud:
2.    def __init__(self, valeur, gauche=None, droite=None):
3.      self.valeur = valeur
4.      self.gauche = gauche
5.      self.droite = droite
6.
7.  arbre = Noeud("Racine")
8.  arbre.gauche = Noeud("E")
9.  arbre.droite = Noeud("T")
10. arbre.gauche.gauche = Noeud("I")
11. arbre.gauche.droite = Noeud("A")
12. arbre.droite.gauche = Noeud("N")
13. arbre.droite.droite = Noeud("M")
```

3. Écrire les instructions à placer en ligne 14 et 15 permettant de créer les nœuds pour les lettres K et S.

4. La fonction `est_present(n,car)` permet de tester si le caractère `car` est présent ou non dans l'arbre `n` de type `Noeud`.

```
1.   def est_present(n, car) :
2.       if n == ..... :
3.           return False
4.       elif n.valeur == ..... :
5.           return True
6.       else :
7.           return est_present(n.droite, car) or .....
```

   a. Recopier le code et compléter les lignes 2, 4 et 7 de la fonction `est_present`.

   b. La fonction `est_present` est-elle récursive ? Justifier votre réponse.

   c. Déterminer quel type de parcours utilise la fonction `est_present`.

5. La fonction `code_morse(n, car)` permet de traduire un caractère `car` **présent** dans l'arbre `n` et renvoie son code morse sous forme d'une chaîne de caractères.

```
8.   def code_morse(n, car):
9.       if n.valeur == car :
10.          return .....
```

```
11.      elif est_present(.....) :
12.          return "-" + code_morse(n.droite, car)
13.      else :
14.          return .....
```

a. Recopier et compléter les ..... des lignes 10, 11 et 14 de la fonction `code_morse`.

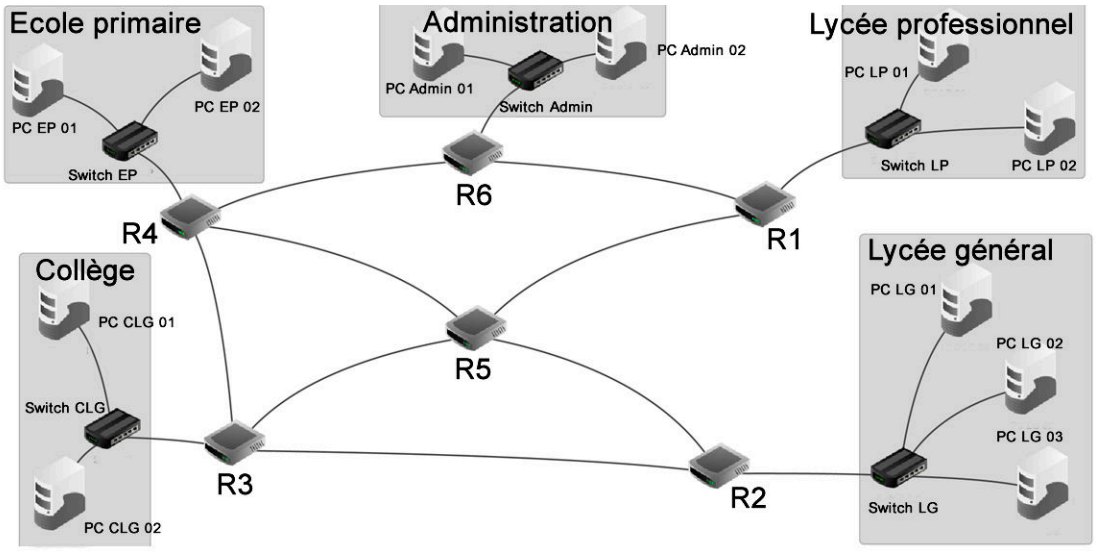
b. Écrire une fonction `morse_message` qui reçoit un arbre de code morse et un message sous forme d'une chaîne de caractères et renvoie le message codé où chaque lettre est séparée par un trait vertical. Par exemple :

```
>>> morse_message(arbre, 'PYTHON')
>>>  o--o|-o--|-|oooo|---|-o|
```

EXERCICE 2 (4 points)

Cet exercice porte sur les réseaux et les protocoles de routage.

Un extrait de l'architecture réseau d'un ensemble scolaire « Etablissement » est présenté ci-dessous. Plusieurs sites de cet ensemble scolaire y sont représentés : l'Administration, l'école primaire, le collège, le lycée général et le lycée professionnel. Les différents postes de ce réseau sont reliés entre eux par des commutateurs (switchs) eux-mêmes reliés à des routeurs.
Au sein de ce réseau, R1, R2, R3, R4, R5 et R6 correspondent aux routeurs de l'Etablissement.



Rappels :
Une adresse IPv4 est composée de 4 octets, soit 32 bits. Elle est notée A.B.C.D où A, B, C et D représentent les 4 octets de l'adresse.
La notation A.B.C.D/n signifie que les n premiers bits de l'adresse IP représentent la partie "réseau", les bits qui suivent représentent la partie "hôte". Les n premiers bits du masque sont donc égaux à 1 et les autres à 0.

L'adresse IPv4 dont tous les bits de la partie "hôte" sont à 0 est appelée "adresse du réseau".
L'adresse IPv4 dont tous les bits de la partie "hôte" sont à 1 est appelée "adresse de diffusion".

Le poste PC LG 03 de la salle informatique du Lycée général a pour adresse IPV4 : 192.168.162.4. On donne ci-dessous le début de l'écriture binaire de cette adresse.

1. Recopier et compléter cette adresse IP : 11000000.10101000.xxxxxxxx.xxxxxxxx

2. Le poste PC Admin 02 du secteur "Administration" a pour adresse IPv4 : 192.168.16.12/24.
   a. Donner l'adresse du réseau local dédié au secteur "Administration" et son masque de sous-réseau.
   b. Donner l'adresse de diffusion (appelée aussi broadcast) de ce réseau.
   c. Donner le nombre maximal de machines que l'on peut connecter sur ce réseau.

3. L'administrateur réseau s'intéresse désormais à la performance des protocoles de routage au sein de l'établissement. En particulier, il porte son attention sur la transmission de paquets de données numériques du secteur "Administration" vers le secteur "Collège".
Son analyse portera donc sur les chemins entre les routeurs R6 et R3.

Le protocole RIP (Routing Information Protocol) permet de construire les tables de routage des différents routeurs, en indiquant pour chaque routeur la distance, en nombre de sauts, qui le sépare d'un autre routeur.

a. On donne ci-dessous les tables de routage des routeurs R1, R2, R3 et R4 obtenues par ce protocole.
Donner les tables de routage des routeurs R5 et R6.

Table de routage de R1		
Destination	passe par	Distance
R2	R5	2
R3	R5	2
R4	R6	2
R5	R5	1
R6	R6	1

Table de routage de R2		
Destination	passe par	Distance
R1	R5	2
R3	R3	1
R4	R3	2
R5	R5	1
R6	R5	3

Table de routage de R3		
Destination	passe par	Distance
R1	R5	2
R2	R2	1
R4	R4	1
R5	R5	1
R6	R4	2

Table de routage de R4		
Destination	passe par	Distance
R1	R6	2
R2	R3	2
R3	R3	1
R5	R5	1
R6	R6	1

On souhaite transmettre un paquet de données depuis le poste PC Admin 01 vers le poste PC CLG 01.

b. Déterminer le parcours emprunté par ce paquet, en utilisant les tables de routage données précédemment,

Suite à une panne, le routeur R4 est déconnecté.

c. Déterminer alors une nouvelle route pouvant être empruntée par les données depuis le secteur "Administration" vers le Collège en effectuant le moins de sauts possibles.

4. Le routeur R4 est réparé et reconnecté au réseau.
On applique désormais le protocole de routage OSPF (Open Shortest Path First) attribuant un coût à chaque liaison afin de privilégier le choix de certaines routes plus rapides.

Le coût d'une liaison est défini par la relation :

$$\text{coût} = \frac{10^8}{d}$$

où d représente le débit en bit/s.

a. Recopier et compléter le tableau suivant :

Liaison	Débit	Coût
Ethernet		10
Fast-Ethernet	10^8	
Fibre	10^9	

b. Recopier le schéma ci-dessous et indiquer le coût de chacune des liaisons connues.

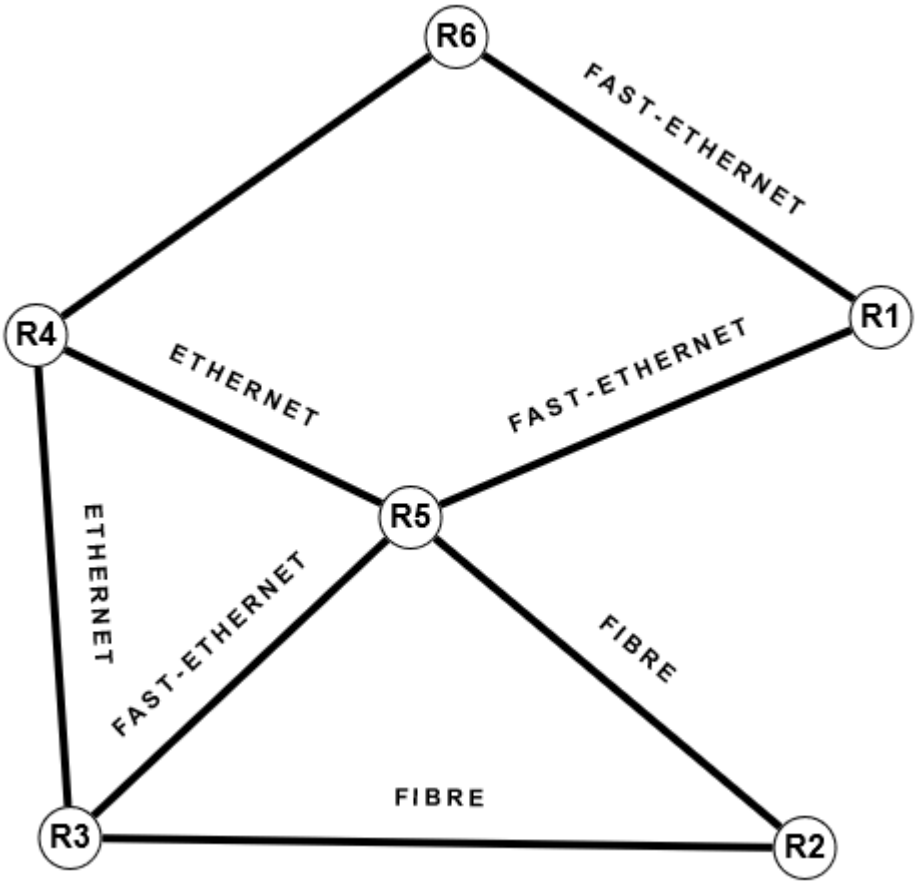


Figure 2 : Représentation du réseau

On sait que le parcours R6 – R4 – R5 – R2 a un coût de 11,1.

c. Déterminer le type de liaison entre R6 et R4.

On souhaite acheminer un paquet de données depuis le Secteur "Administration" vers le réseau du Lycée Général en utilisant le protocole OSPF.

d. **Déterminer** alors la route empruntée par un paquet de données pour aller du poste PC Admin 01 vers le poste ayant pour adresse PC LG 01.

EXERCICE 3 (4 points)

Cet exercice porte sur les bases de données relationnelles et le langage SQL.

Dans cet exercice, on pourra utiliser les mots clés suivants du langage SQL : `SELECT`, `FROM`, `WHERE`, `JOIN…ON`, `INSERT INTO…VALUES`, `UPDATE…SET`, `COUNT`, `ORDER BY`.
On pourra également utiliser la requête `SELECT COUNT(*) FROM table` qui renvoie le nombre d'enregistrements de la relation `table`.

Le gestionnaire d'une agence de locations de voitures a mis en place une base de données relationnelle afin de pouvoir gérer au mieux son parc automobile et ses locations.
Pour cela, il crée trois relations (`vehicule`, `utilisateur` et `location`) dont on donne un extrait ci-dessous :

Relation `vehicule` :

id_vehicule	immatriculation	marque	modele	type	carburant
1	AB – 135 – YZ	Peugeot	208	citadine	diesel
2	EC – 246 – TP	Renault	Zoe	citadine	electrique
3	LC – 231 – MG	Tesla	Model X	SUV	electrique
4	ML – 128 – VM	Citroën	C3	citadine	essence
5	CL – 142 – CE	Citroën	C5	berline	diesel
6	JL – 526 – LM	Peugeot	508	berline	diesel
...	...	...	...	...	...

Relation `utilisateur` :

id_utilisateur	nom	prenom	permis	adresse	ville
131	MARTIN	Jeanne	106566874	2 place de l'Etoile	Paris
132	PETIT	Pierre	75625569	6 rue d'Alsace	Lille
133	DUBOIS	Louise	1448526	52 rue de la Liberté	Rennes
134	DUPONT	Paul	1288638	4 boulevard Jaurès	Paris
135	MOREAU	Violette	14081221926	14 rue du 8 mai	Rouen
136	DURAND	Kevin	19262316811	20 avenue Curie	Caen
...	...	...	...	...	...

Relation `location` :

id_location	id_utilisateur	id_vehicule	date_debut	date_fin
1	135	4	2022-04-14	2022-05-14
2	131	2	2022-06-21	2022-06-28
3	136	5	2022-06-21	2022-06-22
...	...	...	...	...

L'attribut `id_vehicule` de la relation `vehicule`, l'attribut `id_utilisateur` de la relation `utilisateur` et l'attribut `id_location` de la relation `location` sont des clés primaires.

Les attributs `date_debut` et `date_fin` sont de type DATE sous forme AAAA-MM-JJ où AAAA correspond à l'année, MM correspond au mois et JJ au jour du mois.

1.

a. Écrire le résultat de la requête suivante :

```
SELECT id_vehicule
FROM vehicule WHERE type = 'citadine';
```

b. Écrire une requête permettant d'afficher les immatriculations des véhicules diesel.

c. Écrire le résultat de la requête suivante :

```
SELECT immatriculation, modele
FROM vehicule ORDER BY marque ;
```

d. Écrire une requête permettant de calculer le nombre de véhicules dans la table `vehicule`.

e. Écrire une requête permettant d'afficher les noms et prénoms des utilisateurs par ordre alphabétique du nom.

2.

a. Citer, en justifiant, les clés étrangères de la table `location`.

b. Écrire un schéma relationnel des relations `vehicule`, `utilisateur` et `location`.

c. Expliquer pourquoi la requête suivante produit une erreur :

```
INSERT INTO location
VALUES (1, 132, 4, '2022-05-10', '2022-05-12');
```

3.

Une erreur s'est glissée dans la relation `utilisateur` :
Louise DUBOIS habite au numéro 50 de la rue de la Liberté, et non pas au 52.

a. Écrire une requête permettant de modifier cette information.

Un nouveau client souhaite louer une voiture. Ses informations sont les suivantes :

NOM	Prénom	n° de permis	adresse	ville
LEFEBVRE	Gabriel	124689	30 ruelle des champs	Amiens

b. **Écrire** une requête permettant d'ajouter ce nouvel `utilisateur`.
On lui affectera un `id_utilisateur` de 137.

Une nouvelle location de véhicule a été effectuée : Pierre PETIT souhaite louer le véhicule de marque Tesla entre le 21 juin 2022 et le 23 juin 2022.

c. Écrire une requête permettant d'ajouter cette information à la table `location`. On lui donnera un `id_location` égal à 4.

4. Écrire une requête permettant de lister les modèles et les immatriculations des véhicules dont les locations ont débuté le 21 juin 2022 ainsi que les noms et prénoms des utilisateurs correspondants.