

# BACCALAURÉAT GÉNÉRAL

ÉPREUVE D'ENSEIGNEMENT DE SPÉCIALITÉ

**SESSION 2023**

## NUMÉRIQUE ET SCIENCES INFORMATIQUES

**JOUR 1**

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 8 pages numérotées de 1/8 à 8/8.

**Le candidat traite les 3 exercices proposés.**

## EXERCICE 1 (3 points)

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

On rappelle certains des mots-clefs du langage SQL : SELECT, FROM, WHERE, JOIN...ON, UPDATE...SET, INSERT INTO...VALUES.

Notre étude va porter sur les joueurs ayant participé à la coupe du monde de rugby depuis sa création. Pour cela, nous allons utiliser le langage SQL et mettre en place une base de données.

Dans la relation (ou table) `joueur`, l'attribut `id_cdm` correspond à l'identifiant de la première coupe du monde à laquelle le joueur a participé.

`id_joueur` de la relation `joueur` et `id_cdm` de la relation `cdm` sont des clés primaires.

L'attribut `id_cdm` de la relation `joueur` fait directement référence à la clé primaire de la relation `cdm`.

Voici un extrait de la relation `joueur` :

<code>id_joueur</code>	<code>nom</code>	<code>prenom</code>	<code>pays</code>	<code>id_cdm</code>
1	DUPONT	ANTOINE	FRANCE	9
2	BLANCO	SERGE	FRANCE	1
3	WILKINSON	JONNY	ANGLETERRE	5
4	LOMU	JONAH	NOUVELLE-ZELANDE	3
5	CARTER	DAN	NOUVELLE-ZELANDE	5
6	O'DRISCOLL	BRIAN	IRLANDE	3

Voici un extrait de la relation `cdm` :

<code>id_cdm</code>	<code>annee</code>	<code>lieu</code>	<code>vainqueur</code>
1	1987	AUSTRALIE	NOUVELLE-ZELANDE
2	1991	ANGLETERRE	AUSTRALIE
3	1995	AFRIQUE DU SUD	AFRIQUE DU SUD
4	1999	PAYS DE GALLES	AUSTRALIE
5	2003	AUSTRALIE	ANGLETERRE

1.

- Citer, en justifiant, la clé étrangère de la relation `joueur`.
- Écrire un schéma relationnel des relations `joueur` et `cdm` en indiquant les types de chaque attribut.

- c.** Expliquer pourquoi la requête suivante produit une erreur :

```
INSERT INTO cdm VALUES (1, 1987, 'NOUVELLE-ZELANDE',  
                          'NOUVELLE-ZELANDE') ;
```

**2.**

- a.** Écrire l’affichage résultant de la requête suivante pour les extraits des relations précédentes :

```
SELECT nom FROM joueur WHERE id_cdm = 5 ;
```

- b.** Écrire une requête permettant d’afficher les années où l’Australie a gagné la coupe du monde de rugby.

- c.** Écrire l’affichage résultant de la requête suivante pour les extraits des relations données :

```
SELECT nom, prenom FROM joueur ORDER BY id_cdm ;
```

**3.**

Brian O'DRISCOLL a participé à sa première coupe du monde de rugby en 1999 et non pas en 1995.

- a.** Écrire une requête permettant de modifier cette information.

En 2007, la coupe du monde de rugby a eu lieu en FRANCE et a été gagnée par l'AFRIQUE DU SUD.

- b.** Écrire une requête permettant d'ajouter cette information à la relation cdm. On donnera à cet enregistrement id\_cdm égal à 6.

- 4.** Écrire une requête permettant de lister les noms et prénoms des joueurs vainqueurs de la coupe du monde de rugby pour leur première participation en 2003.

## EXERCICE 2 (5 points)

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

On considère l'arbre ci-dessous :

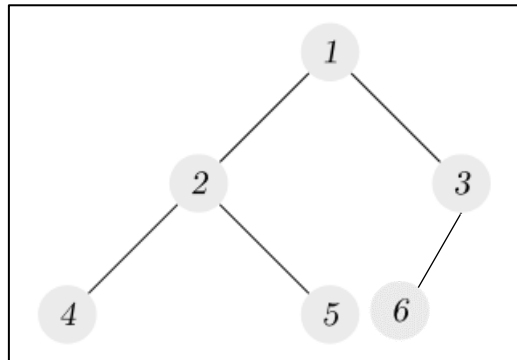


Figure 1 : Arbre binaire n°1

1. Justifier que cet arbre est un arbre binaire.

L'arbre de la figure 1 a une hauteur égale à 2 et une taille égale à 6.

2. Donner la taille maximale d'un arbre binaire de hauteur 10 dont tous les nœuds ont une valeur.
3. On souhaite implémenter les arbres binaires en Python sous forme de tableau de la manière suivante :

```
AB = [arbregauche, etiquette, arbredroit]
```

Par exemple, l'arbre binaire de la figure 2 est implémenté de la manière suivante :

```
AB2 = [[None, 5, None], 1, None]
```

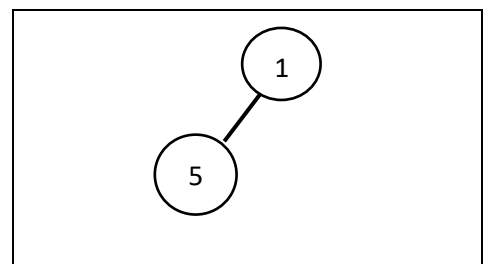
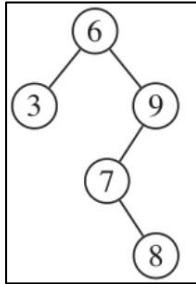


Figure 2 : Arbre binaire n°2

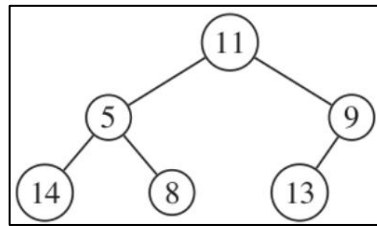
- a. Écrire en une ligne l'instruction permettant de construire et stocker dans la variable `AB1` l'arbre binaire n°1 représenté en Figure 1.
- b. L'instruction `AB[0]` permet d'extraire le sous-arbre gauche de l'arbre binaire `AB`.  
Donner l'instruction permettant d'extraire le sous-arbre droit de l'arbre binaire `AB`.

4. Un arbre binaire de recherche est ordonné de la manière suivante :
- Pour chaque nœud,
- les valeurs de tous les nœuds du sous-arbre gauche sont strictement inférieures à la valeur du nœud ;
  - les valeurs de tous les nœuds du sous-arbre droit sont supérieures ou égales à la valeur du nœud.

Parmi ces deux propositions, déterminer celle qui ne correspond pas à un arbre binaire de recherche et expliquer pourquoi.



**Proposition A**



**Proposition B**

5. On souhaite désormais insérer une valeur `element` dans un arbre binaire de recherche ABR.

Recopier et compléter uniquement les lignes 10, 12, 14, 15 et 17 du code ci-dessous pour que la fonction `insere` positionne la valeur `element` dans l'arbre ABR en respectant la définition d'un arbre binaire de recherche.

```

1  def insere(ABR, element):
2      '''Fonction qui permet d'insérer un élément dans
3          un arbre binaire de recherche et qui renvoie le
4          nouvel arbre construit
5          condition : element ne doit pas être présent
6          dans l'arbre ABR initial'''
7
8      if ABR[1] > element:
9          if ABR[0] is not None:
10             insere(....., .....)
11         else:
12             ABR[0] = [....., ....., .....]
13     if ABR[1] < element:
14         if ABR[.....] is not None:
15             insere(....., element)
16         else:
17             ABR[2] = [....., ....., .....]
18     return ABR
  
```

6. On rappelle que le parcours infixe d'un arbre consiste à parcourir le sous-arbre gauche, la racine puis le sous-arbre droit de cet arbre.

Par exemple, le parcours infixe de l'Arbre binaire n°1 (Figure 1) nous donne :

4 – 2 – 5 – 1 – 6 – 3

- Déterminer le parcours infixe de l'arbre de la **Proposition B** de la question 4.
- Donner la particularité du parcours infixe si l'arbre choisi est un arbre binaire de recherche.
- Écrire une fonction récursive `parcours_infixe` prenant en paramètre un arbre binaire `AB` comme défini dans la question 3. et retournant un tableau contenant les nœuds parcourus dans l'ordre lors d'un parcours infixe.

### EXERCICE 3 (4 points)

*Cet exercice porte sur les réseaux et les protocoles de routages.*

#### 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 sont les valeurs des 4 octets.
- 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 « machine ».
- L'adresse IPv4 dont tous les bits de la partie « machine » sont à 0 est appelée « adresse du réseau ».
- L'adresse IPv4 dont tous les bits de la partie « machine » sont à 1 est appelée « adresse de diffusion ».

On considère le réseau représenté sur la Figure 3 ci-dessous :

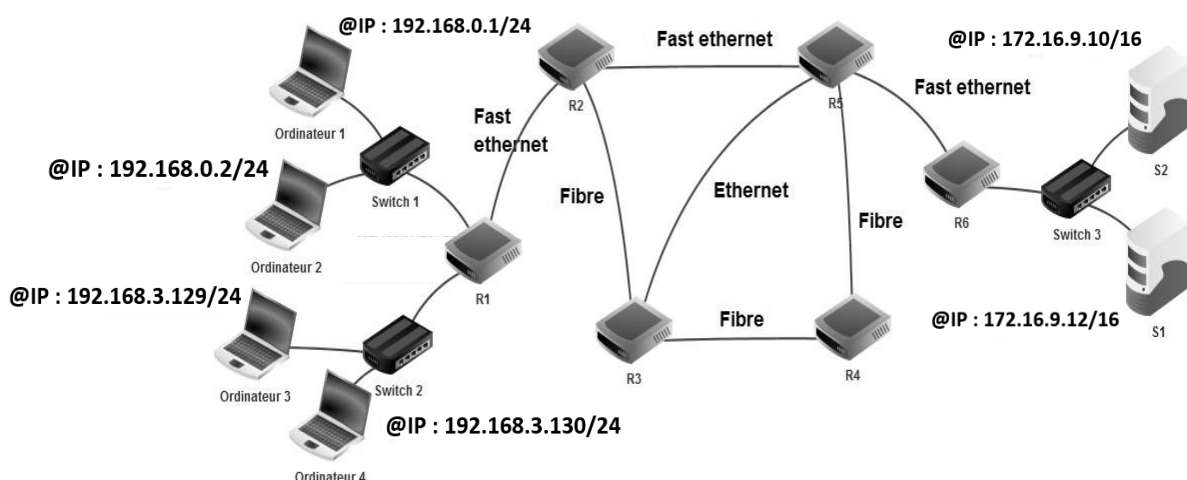


Figure 3 : schéma du réseau

- Compléter les deux derniers octets de l'adresse IP de l'ordinateur n°4 sous forme binaire. IP : 11000000.10101000.xxxxxxxx.xxxxxxxx

- b. Donner le masque de sous-réseau de l'ordinateur n°4.
  - c. Donner l'adresse du réseau de l'ordinateur n°4 sous forme binaire et sous forme décimale.
  - d. Donner l'adresse de diffusion (ou broadcast) du réseau de l'ordinateur n°4 sous forme décimale.
  - e. Donner, en le justifiant, le nombre maximum de machines que l'on peut connecter à ce réseau.
2. Dans cette question, on suppose que le protocole de routage mis en place dans le réseau est RIP. Ce protocole consiste à minimiser le nombre de routeurs traversés. Le schéma du réseau est celui de la figure 1. Les tables de routage utilisées sont données ci-dessous :

Routeur R1	
Destination	Passe par
R2	R2
R3	R2
R4	R2
R5	R2
R6	R2

Routeur R2	
Destination	Passe par
R1	R1
R3	R3
R4	R3
R5	R5
R6	R5

Routeur R3	
Destination	Passe par
R1	...
R2	R2
R4	...
R5	R5
R6	...

Routeur R4	
Destination	Passe par
R1	R3
R2	R3
R3	R3
R5	R5
R6	R5

Routeur R5	
Destination	Passe par
R1	R2
R2	R2
R3	R3
R4	R4
R6	R6

Routeur R6	
Destination	Passe par
R1	R5
R2	R5
R3	R5
R4	R5
R5	R5

- a. Recopier et compléter sur la copie la table de routage du routeur R3.
- Un paquet IP doit aller du routeur R1 au routeur R6.
- b. Donner le parcours emprunté par celui-ci en respectant le protocole RIP et en utilisant les tables de routage.
- Les connexions entre les routeurs R2-R5 et R3-R5 sont coupées.
- c. Réécrire les tables de routage des routeurs R2, R3 et R5 sur la copie.
  - d. Déterminer le nouveau parcours emprunté par le paquet IP pour aller du routeur R1 au routeur R6.
3. Dans cette question, on suppose que le protocole de routage mis en place dans le réseau est OSPF. Ce protocole consiste à minimiser la somme des coûts des liaisons empruntées. Le schéma du réseau est celui de la figure 3 sans problème de connexion.

- Coût des liaisons :

Type de liaison	Coût
Ethernet	10
Fast Ethernet	1
Fibre	0,1

Un paquet IPv4 doit être acheminé d'une machine rajoutée dans le réseau ayant pour adresse IPv4 192.168.0.130/24 à la machine ayant pour adresse IPv4 172.16.9.12/16.

- Écrire les 3 routes possibles, c'est-à-dire la liste des routeurs traversés, et le coût de chacune de ces routes, chaque routeur ne pouvant être traversé qu'une seule fois.
- Donner, en la justifiant, la route qui sera empruntée par un paquet IPv4 pour aller d'une machine ayant pour adresse IPv4 192.168.0.130 à une machine ayant pour adresse IPv4 172.16.9.12.