Polynésie – 2026 – sujet1 - Correction

Exercice 3 (8 points)

Partie 1 : Structures de données

1. Lignes 3 et 6

```
class Chatbot:
    def __init__(self):
        self.base_donnees = {}

    def ajouter_question_reponse(self, question, reponse):
        self.base_donnees[question] = reponse

    def repondre(self, question):
        return self.base_donnees[question]
```

2. Instancier Bot1 et ajouter l'interaction

```
Bot1 = Chatbot()
Bot1.ajouter_question_reponse("Bonjour, comment vas-tu ?", "Bonjour, je vais bien, merci.")
```

3. Obtenir la réponse

```
print(Bot1.repondre("Bonjour, comment vas-tu ?"))
```

👉*Commentaire : si la question n'est pas une clé du dictionnaire, repondre lève une KeyError. On pourrait utiliser self.base_donnees.get(question, "Je ne sais pas.").*

Partie 2 : Interaction avec la base de données

4. Mots clés et définitions

mot clé	définition
SELECT	extraire des données
INSERT	ajouter une donnée
UPDATE	mettre à jour une table

5. Clé primaire

Une **clé primaire** est un attribut (ou un ensemble d'attributs) dont la valeur **identifie de façon unique** chaque ligne de la table : deux lignes ne peuvent pas avoir la même valeur.

6. Clé primaire de questions_reponses

L'attribut **id** : c'est un identifiant numérique, unique pour chaque question. Une question pourrait être reformulée ou avoir des doublons, et plusieurs questions partagent le même mot_cle ou la même réponse.

7. Questions ayant le mot-clé python

```
SELECT question
FROM questions_reponses
WHERE mot_cle = 'python';
```

8. Insérer la question sur Marignan

```
INSERT INTO questions_reponses (id, question, reponse, mot_cle)
VALUES (15, 'Date de la bataille de Marignan ?', '1515', 'Histoire');
```

9. Mettre à jour la réponse de la question 12

```
UPDATE questions_reponses
SET reponse = 'Oui'
WHERE id = 12;
```

Partie 3 : Représentation de la recherche avec les arbres

10. IA, bot, programmation, algo, chatbot, informatique, python

On visite les nœuds niveau par niveau : c'est un **parcours en largeur**.

11. Parcours infixe

algo, bot, chatbot, IA, informatique, programmation, python.

12. Parcours postfixe (suffixe)

algo, chatbot, bot, informatique, python, programmation, IA.

13. Ordre alphabétique

Le **parcours infixe** donne les mots-clés dans l'ordre alphabétique, puisque l'arbre est un ABR.

👉*Commentaire : l'ordre alphabétique est ici celui du dictionnaire, sans tenir compte des majuscules. En Python, 'IA' < 'algo' car les majuscules passent avant les minuscules : avec des chaînes Python, cet arbre ne serait pas un ABR. Il faudrait comparer cle.lower().*

14. Méthode de recherche

```
def recherche(arbre, cle):
    if arbre is None or arbre.cle == cle:
        return arbre
```

```
    if cle < arbre.cle:
        return recherche(arbre.gauche, cle)
    else:
        return recherche(arbre.droit, cle)
```

👉*Commentaire : dans l'énoncé, recherche est écrite dans la classe Noeud mais sans paramètre self et en s'appelant elle-même comme une fonction : telle quelle, l'appel recherche(...) déclenche une NameError. Il faut l'écrire en dehors de la classe (comme ci-dessus), ou comme une vraie méthode avec self et self.gauche.recherche(cle).*