

## Éléments de correction sujet 13 (2024)

### Exercice 1

1. L'attribut CP peut être un INT ou un TEXT (STR)
  2. la requête renvoie 5
  3. Il faut qu'il n'y ait pas 2 agences avec le même numéro de téléphone.
  4. couple\_voitures\_agences (#id\_agence : INT, #id\_voiture : INT)
  5.

```
INSERT INTO couple_voitures_agences
VALUES
(5, 2)
```
  6.

```
UPDATE couple_voitures_agences
SET id_agence = 2
WHERE id_voiture = 2
```
  7.

```
SELECT type, marque, Agence
FROM couple_voitures_agences
JOIN Agences ON couple_voitures_agences.id_agence =
Agences.id_agence
JOIN Voitures ON couple_voitures_agences.id_voiture =
Voitures.id_voiture
```
  8.

```
def insert_voiture(list_valeurs, id_agence):
    req1 = execute_requete_insert("""INSERT INTO Voitures
        ('marque', 'modele', 'kilometrage', 'nombre_place',
        'type', 'carburant')
        VALUES
        (list_valeurs[0], list_valeurs[1], list_valeurs[2],
        list_valeurs[3], list_valeurs[4], list_valeurs[5])
        """)
    req2 = execute_requete_insert("""INSERT INTO couple_voitures_agences
        VALUES
        (?,id_agence)
        """)
    return req1 AND req2
```
- Remarque : Il y a un problème avec cette question : on ne connaît pas l'id de la voiture et on n'a aucun moyen de le récupérer**
9. Il faut bien vérifier la présence de tous les attributs dans le premier paramètre de la fonction. Il faut aussi vérifier le type de ces attributs et que l'id de l'agence existe bien.

## Exercice 2

1. adresse possible : 192.168.1.2
2. adresse réseau L2 : 192.168.18.0 ; nombre d'adresses possibles :  $256 - 3 = 253$

3.

```
G = {
  'R1' : ['R2', 'R3', 'R4', 'R6'],
  'R2' : ['R1', 'R3', 'R5'],
  'R3' : ['R1', 'R2', 'R5', 'R6'],
  'R4' : ['R1', 'R6'],
  'R5' : ['R2', 'R3', 'R6', 'R7'],
  'R6' : ['R1', 'R3', 'R4', 'R5', 'R7', 'R8'],
  'R7' : ['R5', 'R6', 'R8', 'R9'],
  'R8' : ['R6', 'R7', 'R9'],
  'R9' : ['R7', 'R8']
}
```

4.

Destination	Suivant	Nombre de sauts
R2	R2	1
R3	R3	1
R4	R4	1
R6	R6	1
R5	R2	2
R8	R6	2
R7	R6	2
R9	R6	3

5.

P1 -> S1 -> R1 -> R6 -> R8 -> R9 -> S2 -> P3 ; 3 sauts

6.

```
M = [[0,1,1,1,0,1,0,0,0],
      [1,0,1,0,1,0,0,0,0],
      [1,1,0,0,1,1,0,0,0],
      [1,0,0,0,0,1,0,0,0],
      [0,1,1,0,0,1,1,0,0],
      [1,0,1,1,1,0,1,1,0],
      [0,0,0,0,1,1,0,1,1],
      [0,0,0,0,0,1,1,0,1],
      [0,0,0,0,0,0,1,1,0]]
```

7.

```
def degre(MATRICE):  
    d = []  
    for l in MATRICE :  
        cpt = 0  
        for v in l:  
            cpt = cpt + v  
        d.append(cpt)  
    return d
```

8.

[4, 3, 4, 2, 4, 6, 4, 3, 2]

9.

Le graphe possède 2 sommets de degré impair, il admet donc une chaîne eulérienne, le robot va donc pouvoir parcourir l'ensemble du réseau en empruntant chaque fibre optique une et une seule fois.

10.

P1 -> S1 -> R1 -> R2 -> R5 -> R6 -> R7 -> R9 -> S2 -> P3 ; coût du trajet : 6

### Exercice 3

1.

Étage	Écriture Maya	Valeur du chiffre de l'étage	Valeur de conversion
3			
2		$2 \times 5 + 1 = 11$	$11 \times 20^1 = 220$
1		$3 \times 5 = 15$	$15 \times 20^0 = 15$

2.

On a bien en additionnant les étages de la question 1 :  $15 + 220 + 3200 = 3435$

3.

```
M = Maya()  
M.ajouter([0,0,3])  
M.ajouter([0,1,2])  
M.ajouter([0,3,1])
```

4.

```
def nbEtages(self) :  
    return len(self.nombre)
```

5.

```
def valeurChiffre(L):  
    return L[1] + 5 * L[2]
```

```
6.
def MayaToDec(self):
    coeff = 20**(self.nbEtages() - 1)
    ch_Dec = 0
    while not self.estVide() :
        ch_Maya = self.retirer()
        ch_Dec = ch_Dec + (valeurChiffre(ch_Maya)) * coeff
        coeff = coeff // 20
    return ch_Dec
```

```
7.
def decompChiffre(n):
    if n == 0:
        return [1, 0, 0]
    return [0, n % 5, n // 5]
```

```
8.
def DecToMaya(n):
    M = Maya()
    for e in DecToVige(n):
        M.ajouter(decompChiffre(e))
    return M
```

```
9.
def multiplie(self):
    M = Maya()
    M.nombre = [[1,0,0]] + self.nombre
    return M
```

```
10.
([0, 4, 2], 0) et ([1, 0, 0], 1)
```

Il y a des problèmes au niveau du code :

- ligne 4, 5, 6 et 7 : mauvaise indentation

- ligne 13 :  $p + t == 0$  et pas  $p + t = 0$

- ligne 13 :  $(m1[0] == 1 \text{ and } m2[0] == 1)$  à la place de  $(m1[0] == 0 \text{ and } m2[0] == 1)$

```
11.
def somme(self,maya2):
    if self.nbEtages() == maya2.nbEtages():
        r = 0
        res=Maya()
        for i in range(self.nbEtages()):
            s,r=mystere(self.nombre[i],maya2.nombre[i],r)
            res.ajouter(s)
        if r==1:
            res.ajouter([0,1,0])
    return res
```