

Proposition de correction

Exercice 1

Q1.a

5
4
3
2
1

Partez !

Q1.b

elle s'appelle elle-même

Q2.a

```
def ajouter(s, liste) :  
    res = []  
    for m in liste :  
        res.append(s + m)  
    return res
```

Q2.b

["ba", "bb", "bc"]

Q2.c

["a"]

Q3.a

""

Q3.b

["a", "b"]

Q3.c

["aa", "ab", "ba", "bb"]

Exercice 2

Q1.a

C

Q1.b

DBEBEF

Q2

```
def chiffrer (mot : str, alpha : dict) -> str:
    chaine = ""
    for m in mot:
        if m in alpha:
            chaine += alpha[m]
    return chaine
```

Q3.a

dico = {"A" : "C", "B" : "A", "C" : "D", "D" : "B", "E" : "G", "F" : "E", "G" : "F"}

Q3.b

```
def dico_dechiffrement(dico):
    nouveau = {}
    for lettre in dico :
        code = dico[lettre]
        nouveau [code] = lettre
    return nouveau
```

Q3.c

```
def dechiffre_mot(mot, dico):
    return chiffrer(mot, dico_dechiffrement(dico))
```

Q4

```
def dico_chiffrement(alphabet : str) -> dict:
    chiffrement = {}
    if type(alphabet) is str:
        alphabet = list(set(alphabet)) # pas de doublon
        melange = alphabet[:] # shallow copy
        random.shuffle(melange)
        for i in range(len(alphabet)):
            chiffrement[alphabet[i]] = melange[i]
```

return chiffrage

Exercice 3

Q1.a

Code_evaluation : valeur unique

Q1.b

INSERT INTO Evaluations

VALUES('EXKVLX886', 'Term7', 'Peltier', '13/10/2021', 1453)

Q2.a

11

Q2.b

Term7, 13/10/2021

Term5, 07/10/2021

Q2.c

SELECT DISTINCT Nom_evaluation

FROM Evaluations

WHERE Code_competences = 452

ORDER BY Nom_evaluation

Q3.a

le couple doit être unique

Q3.b

SELECT DISTINCT resultat.Num_eleve

FROM resultats, Evaluations

WHERE Evaluations.Code_competences = 532

AND resultat.Code_evaluation = Evaluations.Code_evaluation

ORDER BY resultat.Num_eleve

Q4.a

Attribut	Type
----------	------

id_eleve	INT
nom_eleve	CHAR
prenom_eleve	CHAR
classe	CHAR

Q4.b

id_eleve : identifiant unique

Exercice 4

Q1.a

```
class Carte:
    def __init__(self, val, coul):
        self.valeur = val
        self.couleur = coul
```

Q1.b

c7 = Carte(7, "coeur")

Q2

```
def initialiser() :
    jeu = []
    for c in ["coeur", "carreau", "trefle", "pique"] : # couleur carte
        for v in range(2, 15) : # valeur carte
            carte_cree = Carte(v, c)
            jeu.append(carte_cree)
    return jeu
```

Q3

file: les cartes sont défilées par le haut et enfilées par le bas

Q4

```
def comparer(carte1, carte2):
    if carte1.valeur == carte2.valeur:
        return 0
    elif carte1.valeur > carte2.valeur:
        return 1
    else:
        return -1
```

Exercice 5

Q1.a

0001 1111

Q1.b

- 1111 1111.1111 1111.1111 1111.0000 0000
- 255.255.255.0

Q1.c

254

Q2.a

R6

c'est la passerelle depuis R4 pour un réseau en 192.168.10.0/24

Q2.b

R4 → R6 → R5

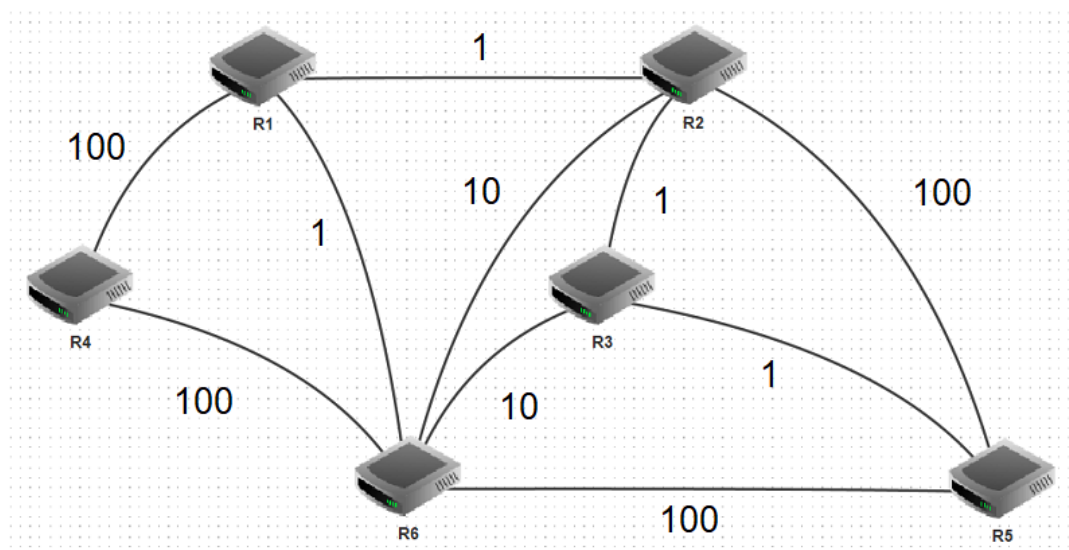
Q3.a

R4 → R1 → R2 → R5

Q3.b

R4	192.168.10.0	144.50.65.1	144.50.65.4
----	--------------	-------------	-------------

Q4.a



R4 → R1 → R2 → R3 → R5 (coût = 103)

Q4.b

R4	192.168.10.0	144.50.65.1	144.50.65.4
R2	192.168.10.0	85.40.65.3	85.40.65.2