2021 centre étrangers sujet 2

Exercice 1

1. [0.5 point]

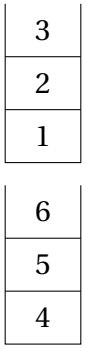
La liste à la fin du mélange sera donc :

```
['10', 'A', '9', 'R', '8', 'D', '7', 'V']
```

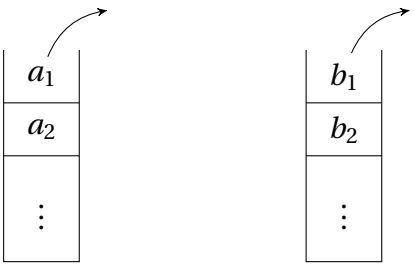
2. [0.5 point]

```
for i in range(N):
    p_temp.empiler(L[i])
return p_temp
```

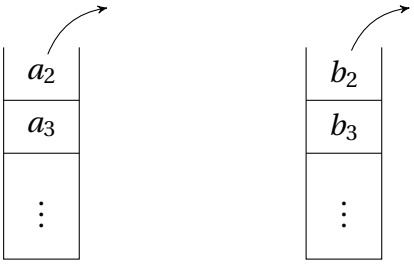
3. [0.5 point]



4.a. [1 point] Tant que les deux piles ne sont pas vides, on dépile le premier élément de chaque pile et on les ajoute à la liste.
liste de départ : `[]`



liste : $[a_1, b_1]$



liste : $[a_1, b_1, a_2, b_2]$
Etc.

4.b. [1 point]

```
def fusion(p1, p2):
    L = []
    while not p1.est_vide():
        L.append(p1.depiler())
        L.append(p2.depiler())
    return L
```

5. [0.5 point]

```
affichage_pile(p_temp)
```

Exercice 2

1. [0.5 point]

```
def mur(laby, i, j):
    return laby[i][j] == 1
```

2.a. [0.75 point] Elle calcule une distance entre les deux cases. La distance vaut 1 si et seulement si les deux cases sont adjacentes. Dans ce cas la fonction renvoie `True`.

2.b. [1 point]

```
def adjacentes(L):
    for in in range(len(L)-1):
        if not voisine(L[i], L[i+1]):
            return False
    return True
```

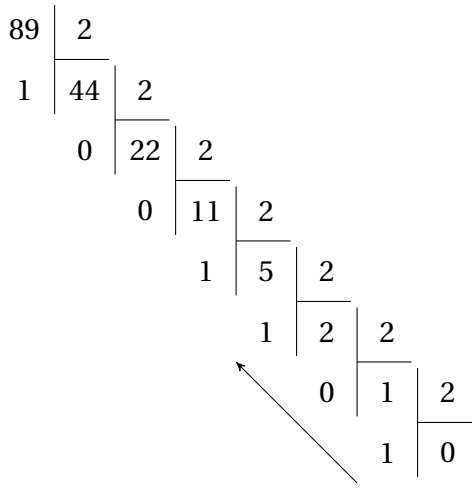
3. [0.75 point] On sort de la boucle si `i >= len(case)` or i est incrémenté de 1 dans la boucle. La boucle se termine donc nécessairement.

4. [1 point]

```
def echappe(cases, laby):
    n = len(laby)
    return teste(cases, laby) and cases[0] == (0, 0) and cases[-1] == (n-1, n-1)
```

Exercice 3

1. [0.5 point] On effectue des divisions successives par 2.



Donc, en ajoutant un zéro devant : $89_{10} = 01011001_2$

2. [0.5 point]

$$\begin{array}{rr} & 11001110 \\ \oplus & 01101011 \\ \hline & 10100101 \end{array}$$

3. [1 point]

```
def xor_crypt(message, cle):
    mess_crypt = []
    for i in range(len(message)):
        nb_crypt = xor(ord(message[i]), ord(cle[i]))
        mess_crypt.append(nb_crypt)
    return mess_crypt
```

4. [1 point]

```
def generer_cle(mot, n):
    cle = ""
    k = len(mot)
    for i in range(n):
        cle = cle + mot[i % k]
    return cle
```

5. [1 point]

E_1	E_2	$E_1 \oplus E_2$	$(E_1 \oplus E_2) \oplus E_2$
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

On remarque qu'on retrouve E_1. Pour décrypter un message il faut donc faire un nouveau XOR avec la même clé.

Exercice 4

1.a. [0.5 point] Non car la clé primaire doit être unique. Or deux licenciés peuvent avoir le même nom.

1.b. [0.5 point] `id_licencie`

2.a. [0.5 point] La liste des noms et prénoms des joueurs de l'équipe des -12ans.

2.b. [0.5 point] Elle renvoie toutes les informations des joueurs de l'équipe des -12ans.

2.c. [0.5 point]

```
SELECT date FROM matchs
WHERE equipe = 'Vétérans'
AND lieu = 'Domicile';
```

3. [0.5 point]

```
INSERT INTO licencies(id_licencie, prenom, nom, annee_naissance, equipe)
VALUES (287, 'Jean', 'Lavenu', 2001, 'Hommes 2');
```

4. [0.5 point]

```
UPDATE lincencies SET equipe = 'Vétérans'
WHERE nom = 'Cuvillier' AND prenom = 'Joseph'
```

5. [0.5 point]

```
WHERE adversaire = 'LSC' AND date = '2021-06-19';
```

Exercice 5

1.a. [0.5 point] (0, 0, 255)

1.b. [0.5 point] 16711680

1.c. [0.5 point] La première ligne récupère la couleur de la LED 0 sous forme d'un tuple RGB et la stocke dans la variable `coul`. La deuxième ligne affiche l'entier `num_color` correspondant à la couleur précédente.

2.a. [0.75 point]

Ble	Ble	Ble	Ble	Ble	Bla	Bla	Bla	Bla	Bla	Rou	Rou	Rou	Rou	Rou

2.b. [0.75 point]

Ver	Jau	Jau	Ver	Jau	Jau	Ver	Jau	Jau	Ver	Jau	Jau	Ver	Jau	Jau

3.a. [0.5 point] Initialise l'objet représentant un bandeau de `Pixel_count` LEDs.

3.b. [0.5 point] Met en bleu les LED 6 et 7 et affiche la modification.