DS 1 Rattrapage (correction)

Exercice 1

Q1

6, 7, 8, 9, 10

Partie A

Q2

```
def effectif(val : int, lst : list) -> int:
  """
  @param val -- une valeur
  @param lst -- une liste
  @return le nombre d'apparitions de val dans lst.
  """
  total = 0
  for elt in lst:
    if elt == val:
      total += 1
  return total
```

Q3

9

Q4

```
def majo_abs1(lst : list) -> int:
  """
  @param lst -- une liste
  @return élément absolument majoritaire s'il existe, None sinon
  """
  majoritaire = len(lst) // 2
  for elt in lst:
    if effectif(elt, lst) > majoritaire:
      return elt
  return None
```

Q5

9 + 1 = 10

Partie B

Q6

```
def eff_dico(lst):
  dico_sortie = {}
```

```

for elt in lst :
    if elt in dico_sortie:
        dico_sortie[elt] += 1
    else:
        dico_sortie[elt] = 1
return dico_sortie

```

Q7

```

def majo_abs2(lst : list) -> int:
    """
    @param lst -- une liste lst
    @return élément absolument majoritaire s'il existe, None sinon
    """
    majoritaire = len(lst) // 2
    effectif = eff_dico(lst)
    for clef, valeur in effectif.items():
        if valeur > majoritaire:
            return clef
    return None

```

partie C

Q8

Lst[0]

Q9

soit $n \in \mathbb{N}^*$, si $\exists m \in \mathbb{N}^* / m > n/2$

alors $\exists m' \in \mathbb{N}^* / m' = m/2 > n/4$

soit $m' > 1/2 n/2$

Q10

```

fonction majoritaire(liste : tableau d'entiers) : entier
début
    n := taille(liste)
    si ( n = 1 ) alors
        renvoyer liste[0]      { cas de base }
    sinon
        liste_gauche := liste[0..n/2 - 1]
        liste_droite := liste[n/2..n - 1]
        nb_majoritaire := majoritaire(liste_gauche)
        si ( nb_majoritaire <> Nil ) alors
            si ( effectif(nb_majoritaire, liste) > n/2 ) alors
                renvoyer nb_majoritaire
        nb_majoritaire := majoritaire(liste_droite)
        si ( nb_majoritaire <> Nil ) alors
            si ( effectif(nb_majoritaire, liste) > n/2 ) alors
                renvoyer nb_majoritaire
fin

```

Q11

```
def majo_abs3(lst):
    n = len(lst)
    if n == 1:
        return lst[0] # cas de base
    else:
        lst_g = lst[:n//2]
        lst_d = lst[n//2:]
        maj_g = majo_abs3(lst_g)
        maj_d = majo_abs3(lst_d)
        if maj_g is not None:
            eff = effectif(maj_g, lst)
            if eff > n/2:
                return maj_g
        if maj_d is not None:
            eff = effectif(maj_d, lst)
            if eff > n/2:
                return maj_d
```

Exercice 2

Partie A

Q1

non, car un même titre peut être repris dans différents albums

Q2

Welcome too the Jungle Appetite for Destruction

Q3

```
SELECT titre
FROM Chanson
WHERE album = 'Showbiz'
ORDER BY titre
```

Q4

```
INSERT INTO Chanson
VALUES(10, 'Megalomania', 'Hullabaloo', 'Muse')
```

Q5

```
UPDATE Chanson
SET titre = 'Welcome to the Jungle'
WHERE id = 7
```

Partie B

Q6

- Élimination des Redondances
- Amélioration de la Cohérence et de l'Intégrité des Données
- Facilité de Mise à Jour
- Optimisation des Performances

Q7

clef étrangère qui met en relation la table Chanson avec la table Album

Q8

- Chanson(id, titre, #id_album)
- Album(id, titre, année, #id_groupe)
- Groupe(id, nom)

Q9

```
SELECT Album.titre
FROM Album, Chanson
WHERE Chanson.titre = 'Showbiz'
AND Album.id = Chanson.id_album
ORDER BY Album.titre
```

Q10

```
SELECT Chanson.titre, Album.titre
FROM Album, Chanson, Groupe
WHERE Groupe.nom = 'Muse'
AND Album.id_groupe = Groupe.id
AND Chanson.id_album = Album.id
ORDER BY Chanson.titre, Album.titre
```

Q11

Donne le nombre d'Albums réalisés par le groupe Muse