ET_Récursivité_Algorithmique_Corrigé

Exercice 1 : Récursivité

1.
   a. Une fonction est dite récursive si cette fonction s'appelle elle-même
   b. Cette fonction s'arrête quand n est égal à -1. En effet, quand n = -1, n >= 0 devient False, on ne "rentre plus dans le if", les appels récursifs cessent.

2.
```
def fact(n) :
    if n == 0:
        return 1
    else :
        return n*fact(n-1)
```

3.
   a. Après l'exécution de `res = somme_entiers_rec(3)` dans la console, on obtient l'affichage suivant :
      3
      2
      1
   b.
      La valeur affectée à la variable res est 6 (3+2+1 = 6)

4.
```
def somme_entiers(n) :
    somme = 0
    while n > 0:
        somme = somme + n
        n = n - 1
    return somme
```

Exercice 2 : POO et algorithmique

Partie 1

1. nom, tab_voisines, tab_couleurs_disponibles et couleur_attribuee sont des attributs de la classe Region.
2. nom_region est de type string
3. `ge = Region("Grand Est")`
4.
```
def renvoie_premiere_couleur_disponible(self):
    return self.tab_couleurs_disponibles[0]
```
5.
```
def renvoie_nb_voisines(self) :
    return len(self.tab_voisines)
```
6.
```
def est_coloriee(self):
    return self.couleur_attribuee != None
```
7.
```
def retire_couleur(self, couleur):
    if couleur in self.tab_couleurs_disponibles:
        self.tab_couleurs_disponibles.remove(couleur)
```
8.
```
def est_voisine(self, region):
    for r in self.tab_voisines:
        if r == region:
            return True
    return False
```

Partie 2

9.
```
def renvoie_tab_regions_non_coloriees(self):
    tab_r =[]
    for r in self.tab_regions:
        if not r.est_coloriee():
            tab_r.append(r)
    return tab_r
```
10.
   a. La méthode renvoie None dans le cas où toutes les régions sont coloriées

   b. La région renvoyée est non coloriée et possède le plus grand nombre de voisins

11.
```
def colorie(self):
    r_max = self.renvoie_max()
    while r_max != None:
        couleur = r_max.renvoie_premiere_couleur_disponible()
        r_max.couleur_attribuee = couleur
        for r in r_max.tab_voisines :
            r.retire_couleur(couleur)
        r_max = self.renvoie_max()
```