Fonctions récursives : Snake (correction)

```
from maillon import Maillon
class Snake :
    '''Classe Snake _ Gestion de la position de Snake
        Position initiale de la tete (50,30)
    '''

    def __init__(self) :
        ''' Constructeur de la classe Snake
        ___position : Liste chainee, tye Maillon
        ___orrientation : str ('bas', 'haut', 'gauche', 'droite'
        '''
        self.__position = Maillon((50,30),None)
        self.__orientation = 'bas'

    def __repr__(self) :
        ''' redefinition de la representation de l'objet Snake '''
        return '('+self.__orientation[0]+')' + str(self.__position)

    def modifier_orientation(self, o) :
        ''' Mutateur de l'attribut __position
        o : str
        '''
        if o not in ['bas', 'haut', 'gauche', 'droite'] :
            raise NameError('orientation invalide')
        else : self.__orientation = o

    def lire_orientation(self) :
        '''Getter de l'attribut __orientation'''
        return self.__orientation

    def lire_positions(self) :
        '''Getter de l'attribut __position'''
        return self.__position

    def lire_tete(self) :
        '''Renvoie la valeur du maillon de tête (tuple)'''
        return self.__position.valeur

    def ajout_tete(self) :
        '''Ajoute un maillon de tête en fonction de l'orientation'''

        if self.__orientation == 'bas' : val =
(self.__position.valeur[0],self.__position.valeur[1]+1)
        elif self.__orientation == 'haut' : val =
(self.__position.valeur[0],self.__position.valeur[1]-1)
        elif self.__orientation == 'droite' : val =
(self.__position.valeur[0]+1,self.__position.valeur[1])
        elif self.__orientation == 'gauche' : val = (self.__position.valeur[0]-
1,self.__position.valeur[1])
        else : raise NameError('orientation invalide')
        m = Maillon(val,self.__position)
```

```
        self.__position = m


    def __retirer_dernier(self,m) :
        '''Détache le dernier maillon de liste chainee'''
        if self.taille()==1 :
            raise IndexError('Chaine trop courte')
        elif m.suivant.suivant == None :
            m.suivant = None
        else : self.__retirer_dernier(m.suivant)

    def couper_queue(self) :
        '''Coupe la queue du serpent'''
        self.__retirer_dernier(self.__position)

    def __longueur(self, m) :
        '''Renvoie le nombre de maillon dans la liste chainee m'''
        if m.suivant==None : return 1
        else : return 1+self.__longueur(m.suivant)

    def taille(self) :
        '''Renvoie la longueur de Snake'''
        return self.__longueur(self.__position)

    def __in(self,val,m) :
        '''Teste si la valeur val est presente dans la liste m'''
        if m.valeur == val : return True
        elif m.suivant == None : return False
        else : return self.__in(val,m.suivant)

    def est_mort(self) :
        ''' Teste si la tete touche le corps du serpent'''
        tete = self.__position.valeur
        return self.__in(tete,self.__position.suivant)
```