Terminale NSI **Structures de Données – Exercices sur les listes**

On donne, ci-dessous, la réalisation d'une **Liste** chaînée sous forme de classe :

```
class Cellule:
    """une cellule d'une liste chaînée"""
    def __init__(self, v, s):
        self.valeur = v
        self.suivante = s

class Liste:
    """une liste chaînée"""
    def __init__(self):
        self.tete = None

    def est_vide(self):
        return self.tete is None

    def ajoute(self, x):
        self.tete = Cellule(x, self.tete)

    def __len__(self):
        return longueur(self.tete)

    def __getitem__(self, n):   # Accéder au nième élément
        return nieme_element(n, self.tete)

    def reverse(self):
        self.tete = renverser(self.tete)

    def __add__(self, lst):
        r = Liste()
        r.tete = concatener(self.tete, lst.tete)
        return r
```

Exercice 1

1) Sous forme récursive, écrire les fonctions
   a) Longueur(liste)
   b) Nieme_elemment(n, liste) (Attention à lever une exception **IndexError)**
   c) Concatener(liste1, liste2)

   Qui permettent d'exécuter la classe **Liste**

2) En utilisant une boucle **while**, écrire la fonction **renverser(liste)**

Exercice 2

Ecrire une fonction **affiche_liste(liste)** qui affiche, en utilisant la fonction **print**, tous les éléments de la liste **liste**.

a) L'écrire comme une fonction récursive.
b) L'écrire avec une boucle **while**
c) Ajouter une méthode **affiche** à la classe **Liste**

Exercice 3

Ecrire une fonction **listeN(n)** qui reçoit en argument un entier **n**, supposé positif ou nul, et renvoie la liste des entiers 1, 2, …, n dans cet ordre.
Si **n = 0**, la liste renvoyée est vide.