5.4 Opérations avec les listes chaînées

NSI Tle - JB Duthoit

5.4.1 Longueur de la liste

```
def longueur(l):
    if lst is None:
        return 0
    else:
        return 1 + longueur(l.suivante)
```

Complexité : la complexité du calcul de la longueur est directement proportionnelle à la longueur elle-même, puisqu'on réalise un nombre constant d'opérations pour chaque cellule de la liste.
Ainsi, pour une liste l de mille cellules, longueur(l) va effectuer mille tests, mille appels récursifs et mille additions.

5.4.2 Renvoyer le n-ème élément d'une liste

```
def nieme_element(n, l):
    if l is None:
        raise IndexError("indice invalide")
    if n == 0:
        return l.valeur
    else:
        return nieme_element(n-1,l.suivante)
```

5.4.3 Concaténer deux listes

```
def concatener(l1, l2):
"""concatène les listes l1 et l2,
sous la forme d'une nouvelle liste"""
i f l1 i s None:
return l2
else:
return Cellule(l1.valeur, concatener(l1.suivante,l2))
```