

```
1 class Maillon:
2     """
3     Crée un maillon de la liste chaînée
4     """
5     def __init__(self, val: int, suiv: object)->None:
6         self.valeur = val
7         self.suivant = suiv
8
9 class Liste:
10     """
11     Crée une liste chaînée
12     """
13     def __init__(self):
14         self.tete: object = None
15
16     def est_vide(self)->bool:
17         return self.tete is None
18
19     def ajoute(self, val: int)->None:
20         self.tete = Maillon(val, self.tete)
21
22     def __len__(self)->int:
23         maillon = self.tete
24         taille = 0
25         while not(maillon == None):
26             maillon = maillon.suivant
27             taille += 1
28         return taille
29
30     def taille_rec(self, maillon: object)->int:
31         """
32         méthode interne pour calculer la taille de la chaîne
33         """
34         if maillon is None:
35             return 0
36         else:
37             return 1 + self.taille_rec(maillon.suivant)
38
39     def taille(self)->int:
40         """
41         appel principal de la méthode récursive pour mesurer
42         la taille de la chaîne
43         """
44         return self.taille_rec(self.tete)
45
46     def __getitem__(self, n: int)->object:
47         """
48         renvoie l'élément de rang n. Les indices commencent à 0.
49         """
50         maillon = self.tete
51         i = 0
```

```
52     while i < n and maillon is not None:
53         maillon = maillon.suivant
54         i += 1
55
56     if maillon is None or n < 0:
57         raise IndexError("indice invalide")
58
59     return maillon.valeur
60
61     def get_element_rec(self, n: int, maillon: object)->int:
62         """
63         méthode interne pour renvoyer le n-ième élément.
64         """
65         if maillon is None or n < 0:
66             raise IndexError("indice invalide")
67         if n == 0:
68             return maillon.valeur
69         else:
70             return self.get_element_rec(n-1, maillon.suivant)
71
72     def get_element(self, n: int)->int:
73         """
74         appel principal de la méthode récursive pour renvoyer le n
75         -ième élément
76         """
77         return self.get_element_rec(n, self.tete)
78
79 lst = Liste()
80 lst.ajoute(8)
81 lst.ajoute(5)
82 lst.ajoute(3)
83 lst.ajoute(9)
84 print(len(lst))
85 print(lst.taille())
86 print(lst[3])
87 print(lst.get_element(3))
```