

Listes Chainées

L'objectif de ce tp est d'implémenter les listes chaînées en Python, sans recourir au type list.

Classe Maillon : Les objets de la classe Maillon ont deux attributs :

- val, la valeur du terme
- suiv, qui pointe vers le terme suivant.

1. Compléter le constructeur de la classe Maillon de telle sorte que :

```
class Maillon:
    def __init__(self, valeur=None, suivant=None):
        pass
```

2. Quel est le maillon créé par Maillon("D") ?

3. a) Compléter la classe Maillon avec la méthode __str__() qui renvoie:

- une chaîne de caractère du type valeur-suivant_non_vide (utiliser str(self.suivant) pour l'affichage de suivant non-vide)
- simplement valeur si le maillon n'a pas de suivant.

- b) Qu'obtient-on avec le code suivant :

```
L=Maillon(3, Maillon(2, Maillon(1)))
print(L)
```

- c) le 1^{er} élément de la liste est :

Classe Liste_Chainee

4. Compléter le code de la classe Liste_Chainee (rien à modifier dans __str__())

```
class Liste_Chainee:
    def __init__(self, premier=None): #premier: 1ere valeur de la liste
        self.prem = Maillon(premier)

    def ajoute(self, valeur): #ajout d'un élément en fin de liste
        m = . . . . . #creation d'un maillon
        if . . . . . is None: #si la liste est vide
            self.prem = . .
        else:
            p = . . . . . #p est initialise au 1er maillon
            while . . . . . is not None : #recherche du dernier élément
                p = . . . . .
            . . . . . = m # m est le dernier élément

    def __str__(self):
        return str(self.prem)
```

En déduire la construction la liste des entiers de 1 à 10 avec une liste chaînée

5. Compléter la classe Liste_Chainee avec les méthodes suivantes :

- a) tete() qui renvoie la tête de la liste (la tête est une valeur).

- b) queue() qui renvoie la liste après retrait du premier élément.

- c) longueur() (__len__(self)) qui renvoie le nombre d'éléments de la liste

- d) insere1(valeur) qui insère valeur en tête de liste.

- e) terme_indice(i) qui renvoie la valeur de l'élément d'indice i (sachant que l'indexation commence à 0) et None s'il n'y en a pas.

- f) insere(i,v) qui insère un élément d'indice i dans la liste ayant pour valeur v si c'est possible et sinon ne fait rien.

6. Bonus : En utilisant la classe Liste_Chainee, écrire les méthodes

- a) __add__(self, autreliste) qui ajoute autreliste à la liste

- b) nb_occurences(elem) qui compte le nombre d'occurrences de elem dans la liste

- c) supprime(i) qui supprime l'élément d'indice i de la liste