

OBJECTIFS : L'objectif de ce TP est de créer le code d'une classe Liste qui permet d'instancier des objets de type *listes chainées*.

DOCUMENT A RENDRE : Ce travail est évalué. Le code python réalisé sera rendu sur le site *nsibranly.fr* en utilisant le code : **listes** .

1. Démarrage :

Le code de départ est le suivant. On y définit 2 classes avec leur constructeur.

Dans la partie programme principal, on crée un objet liste de la classe *Liste()*. Cet objet est nommé *l* .

On crée *manuellement* 4 objets de la classe *Cellule()* et on crée également *manuellement* un *chainage* entre ces 4 objets.

On a ainsi créé une liste chaînée qui comprend les valeurs 36 , 15 , 0 et 5 .

```
class Cellule :
    def __init__(self, val = None) :
        self.val = val
        self.suivant = None

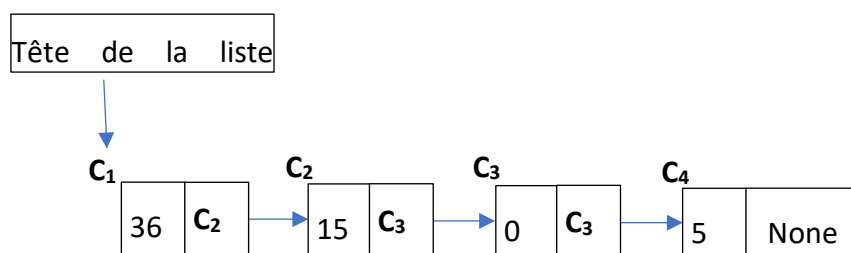
class Liste :
    def __init__(self) :
        self.tete = None

# Programme principal
# création manuelle de la liste l = [36,15,0,5]
l = Liste()
c1 = Cellule(36)
l.tete = c1

c2 = Cellule(15)
c1.suivant = c2

c3 = Cellule(0)
c2.suivant = c3

c4 = Cellule(5)
c3.suivant = c4
```



2. Travail à faire :

On vous demande de créer, dans l'ordre, les méthodes suivantes de la classe Liste :

- *taille_liste()* qui renvoie le nombre d'éléments de la liste chaînée
- *affichage()* qui écrit dans le shell la valeur des éléments de la liste chaînée
- *recherche_nb_fois_valeur(valeur)* qui renvoie le nombre d'éléments de la liste chaînée pour lesquels la valeur est égale à celle mise en argument

- `insérer_en_tete(valeur)` qui insère en tête de la liste chaînée, la valeur mise en argument
- `insérer_en_queue(valeur)` qui insère en queue de la liste chaînée, la valeur mise en argument
- `insérer_position(position,valeur)` qui insère la valeur mise en argument à la position donnée en argument

Ainsi avec le programme principal suivant :

```
# Programme principal
# création manuelle de la liste l = [36,15,0,5]
l = Liste()
c1 = Cellule(36)
l.tete = c1

c2 = Cellule(15)
c1.suivant = c2

c3 = Cellule(0)
c2.suivant = c3

c4 = Cellule(5)
c3.suivant = c4

# tests des méthodes
print(f"Cette liste compte {l.taille_liste()} éléments")
l.affichage()
print(f"La valeur 5 a été trouvée {l.recherche_nb_fois_valeur(5)} fois")
print(f"La valeur 7 a été trouvée {l.recherche_nb_fois_valeur(7)} fois")
l.insérer_en_tete(45)
l.affichage()
l.insérer_en_queue(47)
l.affichage()
l.insérer_position(3,36)
l.affichage()
```

le résultat de
l'exécution sera :

```
>>> (executing file "liste_chainees_cours.py")
Cette liste compte 4 éléments
Valeurs de la liste : 36 15 0 5
La valeur 5 a été trouvée 1 fois
La valeur 7 a été trouvée 0 fois
Valeurs de la liste : 45 36 15 0 5
Valeurs de la liste : 45 36 15 0 5 47
Valeurs de la liste : 45 36 15 36 0 5 47
```

Note importante : Ecrire le script de chaque méthode, l'une après l'autre. Effectuer toujours de multiples tests, ne pas hésiter à écrire avec des `print()` le contenu des variables dans le shell et à raisonner sur le papier avec un crayon.