

Objectifs .:

1 Classe Maillon :

Une liste chaînée (ou liste liée) est une structure de données composées d'une séquence d'éléments de liste. Chaque enregistrement d'une liste chaînée est appelé **maillon**. Le champ de chaque maillon qui contient l'adresse du maillon suivant ou précédent est généralement appelé pointeur.

Chaque élément (ou maillon) de la liste est composé :

- d'un contenu utile (de n'importe quel type),
- d'un pointeur pointant vers l'élément suivant de la séquence.

Le dernier élément de la liste possède un pointeur vide :

```
class Maillon:
    """Maillon d'une liste chaine"""
    def __init__(self,valeur=None,suivant=None):
        self.valeur=valeur
        self.suivant=suivant

    def __str__(self):
        if self.suivant==None:
            return f"{self.valeur} -> None"
        else:
            return f"{self.valeur} -> {str(self.suivant)}"
```

Une adresse contient le pays, la region, la ville, la rue, le numéro.

Q1. Implémenter l'adresse sous la forme d'une liste chaînée.

2 Classe Liste_chaine

```
class Liste_chaine :  
  
    def __init__ ( self ):  
        self.sommet = None  
  
    def __str__(self):  
        if self.sommet is None:  
            return "Pas d'element dans la liste"  
        else:  
            return str(self.sommet)  
  
    def ajout_debut ( self , element ):  
        m = Maillon ( element , self . sommet )  
        self . sommet = m  
  
    def suppression_tete ( self ):  
        e = self . sommet . element  
        self . sommet = self . sommet . suivant  
        return e
```

Implémenter les méthodes suivantes :

- `est_vide(L)` : renvoie vrai si la liste est vide,
- `taille(L)` : renvoie le nombre d'éléments de la liste,
- `get_dernier_maillon(L)` : renvoie le dernier élément de la liste,
- `get_maillon_indice(L, i)` : renvoie le maillon d'indice `i`,
- `ajouter_debut(L, d)` : ajoute un élément `d` au début de la liste,
- `ajouter_fin(L, d)` : ajoute un élément `d` à la fin de la liste,
- `insérer_apres(L, i, M)` : insère un maillon `M` à l'indice `i`,
- `supprimer_apres(L, M)` : supprime le maillon suivant le maillon `M`,