

Bases de la programmation impérative

Ensimag 1A

1. Renversement de liste

On se propose de réaliser un renversement de liste. Une liste contenant des valeurs (1,2,3) deviendrait ainsi une liste contenant les valeurs (3,2,1).

On souhaite réaliser cette opération *en une seule passe*, sans création de cellules supplémentaires.

2. Sentinelles

Les listes vues dans les TD précédents ne sont pas toujours faciles à programmer. La quasi-totalité des fonctions développées nécessite de faire attention à des cas spéciaux. Par exemple, lors de l'ajout en queue d'un élément on écrit `liste.queue.suivant = nouvelle_cellule`, ce qui ne marche que pour une liste non vide.

On se propose de simplifier la programmation des listes à l'aide d'une astuce très utile : les sentinelles.

On ajoute un élément fictif en tête de liste (dans une cellule). Avec cet ajout, nous avons la garantie que la liste ne sera jamais vide.

On modifie donc le constructeur de la classe liste en conséquence :

```
1 class Liste:
2     """
3     listes simplement chainees, avec sentinelle.
4     """
5     def __init__(self, valeurs=()):
6         self.tete = Cellule(None) # sentinelle
7         self.queue = self.tete
8         for valeur in valeurs:
9             self.ajouter_en_queue(valeur)
```

2.1. Implémentez `ajouter_en_queue(self, valeur)`.

2.2. Implémentez `ajouter_en_tete(self, valeur)`. Attention, on ajoute en tête mais juste après la sentinelle.

2.3. Implémentez un itérateur `valeurs(self)` itérant sur toutes les valeurs réelles. Utilisez-le pour réaliser `__str__`.

2.4. Supprimez la première cellule contenant une valeur donnée. Implémentez `supprimer(self, valeur)`.