Nom : 25 novembre 2020

Interro — Structures abstraites 1

— 30'

Convention : Dans ce contrôle, on suppose qu'une liste chainée est un objet ListeC.

Le premier maillon de la liste L est obtenu avec l'instruction **L.tete**.

Chaque maillon M est représenté par un objet dont la valeur est **M.valeur** et le suivant **M.suivant**. Le maillon vide sera appelé None.

```
class ListeC:
    def __init__(self):
        # liste vide
        self.tete = None
```

Exercice 1 *(1 point)*

Rappelez la différence entre type abstrait (=interface) et implémentation.

Exercice 2 *(3 point)*

On considère une liste chainée L représentée ci-dessous :

a. Donnez l'instruction permettant d'afficher la valeur "astérix".

b. Donnez l'instruction permettant d'afficher le maillon dont la valeur est "idéfix".

c. Dans cet exemple particulier, proposez un code ou une instruction permettant de supprimer le dernier Maillon.

Exercice 3 *(3 points)*

Nous disposons du programme ci-contre :

a. Représenter graphiquement l'état de la liste chainée L après les lignes 1 à 3.

b. Représenter graphiquement l'état de la liste chainée L1 après les lignes 5 à 10.

c. Ligne 12 : représenter graphiquement le lien entre les listes chainées L et L1. Quelle opération est réalisée ?

d. Écrire l'état des listes chainées L et L1 après la ligne 14. Avez-vous déjà rencontré ce problème en Python ?

```
1  L = ListeC()
2  L.tete =  Maillon()
3  L.tete.valeur = 7
4
5  M1, M2, M3 =  Maillon(), Maillon(), Maillon()
6  M1.valeur, M2.valeur, M3.valeur = 11, 13, 17
7  M1.suivant, M2.suivant = M2, M3
8
9  L1 = ListeC()
10 L1.tete = M1
11
12 L.tete.suivant = L1.tete
13
14 L1.tete.valeur = 88
```

Tournez SVP

Exercice 4 *(3 points)*

Au sein de la classe ListeC, on propose la méthode **mystere** ci-dessous :

```
1  def mystere(self, valeur):
2      if self.tete is None:
3          self.tete = Maillon()
4          self.tete.valeur = valeur
5      else:
6          m = self.tete
7          while m.suivant is not None:
8              m = m.suivant
9          m.suivant = Maillon()
10         m.suivant.valeur=valeur
```

a. On dispose du programme principal ci-dessous **:**

```
L = ListeC()
L.mystere('NSI')
print(L)
```

En expliquant rapidement votre raisonnement, expliquez quel va être l'état de la liste chainée après exécution du programme principal.

b. En déduire ce que font les lignes 2 à 4 de la fonction mystere.

c. En expliquant votre raisonnement, expliquez ce que font les lignes 6 à 10 de la fonction mystere. Votre démonstration pourra utiliser un dessin.

Exercice 4 *(2 points)*

Compléter les lignes 2 et 5 du programme ci-dessous de sorte que la fonction `produitDesChiffres(n)` calcule le produit des chiffres composant l'entier positif n.
Par exemple, `produitDesChiffres(243)` doit renvoyer 2 × 4 × 3 = 24.

```
def produitDesChiffres(n):
    if ......:
        return n
    else:
        return n%10 * ..................
```