Université d'Artois — Faculté des Sciences Jean Perrin

Algorithmique et Programmation 3

TP5 — Listes chaînées (version itérative)

Travail à réaliser

- Reprenez les classes `Maillon` et `ListeChainee`.
- Testez les différentes méthodes vues en cours.
- Écrire la méthode `contient` permettant de tester si une valeur donnée est présente dans une liste chaînée :

```
>>> lc : 1 -> 2 -> 10 -> 5 -> 0
>>> lc.contient(7)
False
```

- Écrire la méthode `moyenne` retournant la moyenne des valeurs d'une liste chaînée :

```
>>> lc : 1 -> 2 -> 10 -> 5 -> 0 -> 7
>>> lc.moyenne()
5
```

- Écrire la méthode `indices_minimum` retournant les indices des maillons de valeur minimale :

```
>>> lc : 1 -> -2 -> 10 -> 5 -> -2
>>> lc.indices_minimum()
[1, 4]
```

- Écrire la méthode `kieme_depuis_fin` retournant la valeur du kième élément depuis la fin :

```
>>> lc : 1 -> -2 -> 10 -> 5 -> -2
>>> lc.kieme_depuis_fin(2)
5
```

- Écrire la méthode `permute_tete_queue` permettant de permuter la tête et la queue :

```
>>> lc : 1 -> -2 -> 10 -> 5 -> -2
>>> lc.permute_tete_queue()
>>> lc
-2 -> -2 -> 10 -> 5 -> 1
```

- Écrire la méthode `premiere_repetition` trouvant le premier élément qui se répète :

```
>>> lc : 1 -> -2 -> 10 -> 5 -> -2 -> 5 -> 12 -> 10
>>> lc.premiere_repetition()
-2
```

– Écrire la méthode `permute_pairs` permettant de permuter chaque deux maillons consécutifs :

```
>>> lc : 1 -> -2 -> 10 -> 5 -> -2 -> 5 -> 12 -> 10
>>> lc.permute_pairs()
>>> lc
-2 -> 1 -> 5 -> 10 -> 5 -> -2 -> 10 -> 12
```

– Écrire une méthode permettant de diviser une liste chaînée en deux listes, en attribuant les maillons à tour de rôle.