

Commencez par récupérer le squelette de code sur le gitlab du cours <https://gitlabinfo.iutmontp.univ-montp2.fr/r3.02-dev-efficace>.

## 1 Exercices où l'on ne modifie pas this

### Exercice 1. Longueur

#### Question 1.1.

Ecrire une méthode `int longueur()` qui calcule la longueur de `this`.

### Exercice 2. somme

#### Question 2.1.

Ecrire une méthode `int somme()` qui calcule la somme des éléments de `this`.

### Exercice 3. Croissant

#### Question 3.1.

Ecrire une méthode `boolean croissant()` qui retourne vrai ssi les entiers de `this` sont triés par ordre croissant.

### Exercice 4. Get

#### Question 4.1.

Ecrire une méthode `int get(int i)` qui pour tout  $i$ ,  $0 \leq i < this.longueur()$ , renvoie l'entier en position  $i$  de `this`.

## 2 Ecriture de méthodes void

### Exercice 5. AjoutFin

#### Question 5.1.

Ecrire une méthode `void ajoutFin(int x)` qui modifie `this` en ajoutant  $x$  à la fin.

### Exercice 6. Concaténation

#### Question 6.1.

Ecrire une méthode `void concat(Liste l)`. Pré-requis :  $l$  n'a aucun maillon en commun avec `this`. Action (décrite informellement) : raccorde la fin de `this` avec le début de  $l$ .

#### Question 6.2.

Que voit on affiché dans le programme suivant ?

- `Liste L1 = (1); Liste L2 = (2,3);`
- `L1.concat(L2);`
- `System.out.println(L1);`
- `L2.val = 50;`
- `System.out.println(L1);`
- `L2.suiv.val = 51;`
- `System.out.println(L1);`

### 3 Ecriture de méthode avec des variantes de spécifications

#### Exercice 7. Suppression d'occurences

##### Question 7.1.

Ecrire une méthode `Liste supprOccs(int x)` qui retourne une liste indépendante de `this` en supprimant toutes les occurences de `x` dans `this`.

##### Question 7.2.

Ecrire une méthode `Liste supprOccsV2(int x)` qui retourne une liste en supprimant toutes les occurences de `x` dans `l`, et ne crée pas de nouveau maillon.

### Enigme bonus

#### Exercice 8. Retournement de l'enfer

##### Question 8.1.

Ecrire une méthode `Liste retourne()` qui renvoie `this` retournée .. sans utiliser de boucle, ni de méthode auxiliaire (en particulier un `ajoutFin(..)`, qui rendrait la tâche facile!)