2 Quelques opérations sur les listes

Implém nt r l s class s List t El m nt. Ajout r dans chaqu class un construct ur par défaut t d s construct urs qui pr nn nt d s paramètr s.

La class List possèd d s opérations élém ntair s qu vous av z vu dans TD : t st r si la list st vid , ajout r un élém nt au début ou à la fin, r tourn r l nombr d'élém nts, supprim r un élém nt, ... On y ajout ra qu qu ls autr s opérations suivant s.

1. Écrir la méthod toString() qui r nvoi la chaîn d caractèr cont nant l cont nu d la list .
2. Écrir dans la class List un méthod itérativ `sommeI` qui r tourn la somm d tous l s élém nt d la list .

   Écrir un méthod récursiv `sommeR` qui r tourn la somm d tous l s élém nt d la list .

   Si n st l nombr d'élém nts d la list , qu ll st la compl xité d `sommeI` t d `sommeR` ?
3. Écrir un méthod `supprimer(int k)` qui nlèv l $k^{\text{ième}}$ élém nt d la list .

   Fait s att ntion n supprimant l pr mi r ou l d rni r élém nt,qu s pass t'il si la list n'a qu'un s ul élém nt ?
4. Écrir un méthod `jouter(int k,int x)` qui ajout l'élém nt x à la $k^{\text{ième}}$ position d la list .
5. Écrir un méthod `sup(int x)` qui, étant donné un nti r, r nvoi un nouv ll list qui n conti nt qu l s élém nts strict m nts supéri urs à x.
6. Écrir un méthod `nettoie(int x)` qui supprim d la list tous l s élém nts égaux à x t r nvoi l nombr d'occur nc d x dans la list .
7. Soit `l'` st la list obt nu d `l` n supprimant tous l s élém nts l s plus grands, alors la list trié d `l` st xact m nt c ll trié d `l'` concat né av c l s élém nts l s plus grands à la fin.

   Écrir un méthod récursiv qui tri un list n appuyant sur c tt r marqu . Vous pouv z écrir un méthod s auxiliair qui calcul la val ur la plus grand dans un list . Qu ll st la compl xité d c tt méthod ?
8. Si on souhait avoir un list dont chaqu élém nt n' st pas qu'un nti r mais conti nt d s informations d'un étudiant : son nom t sa not . Comm nt p ut on adapt r la class `Element` pour avoir un t ll list ? Adapt r aussi la méthod d tri pour tri r la list dans l'ordr croissant d not s d s étudiants.

3 Listes doublement chaînées

On souhait maint nant cré r d s list s doubl m nt chainé s, (p rm ttant d s déplac r dans d ux dir ctions). Adapt z l s class s List t Elém nt afin d'obt nir d s list s doubl m nt chainé s.

1. Écrir un méthod `supprimerDebut()` qui nlèv l'élém nt début d la list .
2. Écrir un méthod `supprimerFin()` qui nlèv l d rni r élém nt d la list .
3. Écrir un méthod `jouterDebut(int )` qui ajout un élém nt au début d la list .
4. Écrir un méthod `jouterFin(int )` qui ajout un élém nt à la fin d la list .

5. Écrir un méthod qui inv rs la list (l pr mi r élém nt d vi nt l d rni r t vic - v rsa...).
6. Écrir un méthod `conc ten tion(List l)` qui concatèn la list `l` à la fin da la list courant .
7. Écrir un méthod flipFlap() qui inv rs d ux élém nts consécutifs (l pr mi r t l s cond, l l troisièm t l quatrièm ,...).
8. Écrir un méthod fibonacci(int n) qui r nvoi la list d s n pr mi rs élém nts d la suit d Fibonacci (astuc : s s rvir d s d ux élém nts précéd nts). Pour rapp l $F_n = F_{n-2} + F_{n-1}$ si $n > 1$, $F_0 = 0$ t $F_1 = 1$.