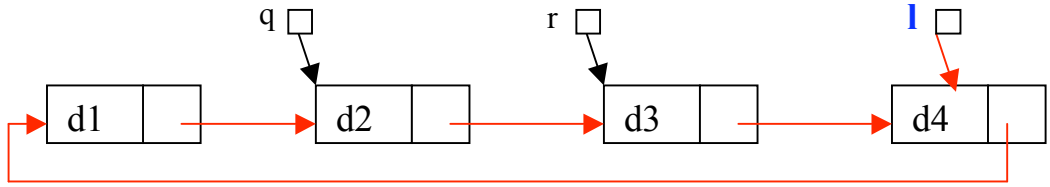
E.D. Algorithmes et Structures de Données n° 2

Thème : Les Listes

Exercice II.1 Manipulation d'une liste chaînée circulaire

Soit **l** la liste chaînée **circulaire** suivante:



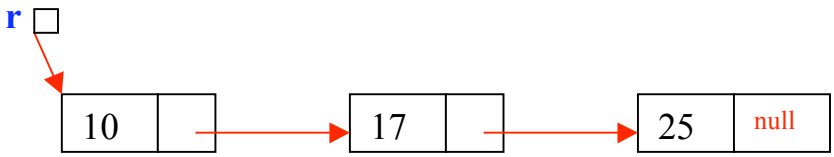
l est le pointeur qui désigne la liste. Il pointe sur le dernier élément ajouté à la liste et, si la liste n'est pas vide, l.suivant désigne le premier élément ajouté à la liste.

Considérons deux pointeurs de travail supplémentaires q et r, indiqués dans le dessin ci-dessus. Dire à quoi sont égaux, successivement:

r.valeur; q.suivant; q.suivant.valeur; r.suivant.suivant.valeur;

Exercice II.2

Soit r la liste chaînée **ordonnée** suivante:



On suppose que pour toute place p de la liste sauf la dernière, on a p.valeur<p.suivant .valeur et que la liste n'est pas vide.

On considère la méthode suivante :

```
Liste qui_fait_quoi (Elt x)
Liste p=this ;
booleen arret = false;
début
  si (p.valeur > x)
  alors retourner new Liste (x,this)
  sinon si (p.valeur == x) alors retourner this;
        sinon
        tant que (arret==false) et (p.suivant ≠ null)  faire
```

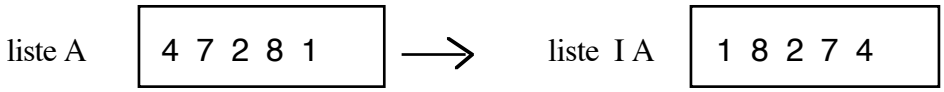
```
                si    (p.suivant.valeur ≥ x )
                alors  arret = true;
                sinon  p = p.suivant;
                finsi;
             fait;
          si (p.suivant==null ou
          (p.suivant ≠ null et p.suivant.valeur ≠ x)
          alors Liste q = new Liste(x,p.suivant) ;
                p.suivant=q ;
          finsi;
          retourner this ;
         finsi;
fin
```

Question 1 Que fait cette méthode ?

Question 2 Soit la liste obtenue à partir de la liste r donnée ci-dessus et APRES exécution de `r=r.qui_fait_quoi(20)`. Dire à quoi sont égaux dans la nouvelle liste, respectivement **r.suivant; r.suivant.suivant.valeur.**

Exercice II.3 Inversion d'une liste chaînée - Algorithme itératif

On se propose de réaliser l'inversion d'une liste chaînée. On veut utiliser la classe Liste et ses méthodes vues en cours.



Question 1 Écrire une nouvelle méthode `renverser` qui inverse la liste courante. On supposera que cette liste n'est pas vide.

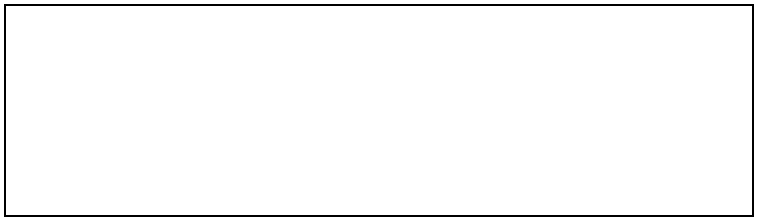
Question 2 Calculer la complexité de cette méthode.

Exercice II.4 Inversion récursive d'une liste chaînée

On veut réaliser l'inversion d'une liste chaînée par une méthode récursive, en utilisant les listes chaînées vues en cours.

Question 1 Définir une fonction externe `Elt en_tête` qui renvoie la valeur du premier élément de la liste.

Question 2 Ecrire la méthode récursive `Liste inverser` en n'utilisant que les méthodes des listes chaînées.



Question 3 Calculer la complexité de cette procédure.

Exercice II.5 Inversion d'une liste contiguë

Question 1 Définir un principe efficace d'inversion d'une liste représentée par un tableau**.**

Question 2 Écrire la méthode inverser et calculer sa complexité.