
TD 3 - Les listes chaînées

Inès de Courchelle, Elisabeth Ranisavljevic



2023-2024

**Objectifs :**

- Étudier des structures de données liées
- Connaître les particularités des chaînes
- Appréhender des notions de gestion de données

Durée 1h30**Format** papier**Attention** Tous les exercices ne seront pas corrigés en cours. On vous en laisse pour vos révisions !

Les listes chaînées

Exercice 1 : Quiz

1. VRAI ou FAUX. Une liste chaînée simple a son dernier élément qui pointe vers NULL

2. Donnez une définition d'une liste doublement chaînée simple. Vous pouvez illustrer votre réponse.

3. VRAI ou FAUX. Une liste doublement chaînée peut être circulaire.

4. Donnez des exemples de listes chaînées que l'on rencontre dans son quotidien.

Exercice 2 : Manipulations basiques

1. Donnez la structure d'une liste chaînée simple contenant uniquement des entiers

2. Donnez l'algorithme d'une fonction permettant de créer un maillon d'une liste chaînée simple

Entête de la fonction

```
1 FONCTION creerMaillon(donnee : entier) : maillon
```

3. Donnez l'algorithme permettant d'ajouter un élément à ma chaine

Entête de la fonction

```
1 FONCTION ajouterTete(maListe : maillon, donnee : entier): maillon
```

4. Donnez l'algorithme d'une fonction qui initialise une liste avec un nombre d'éléments dont la donnée est aléatoire

Entête de la fonction

```
1 FONCTION initListe(nbElements : entier, liste : maillon) : maillon
```

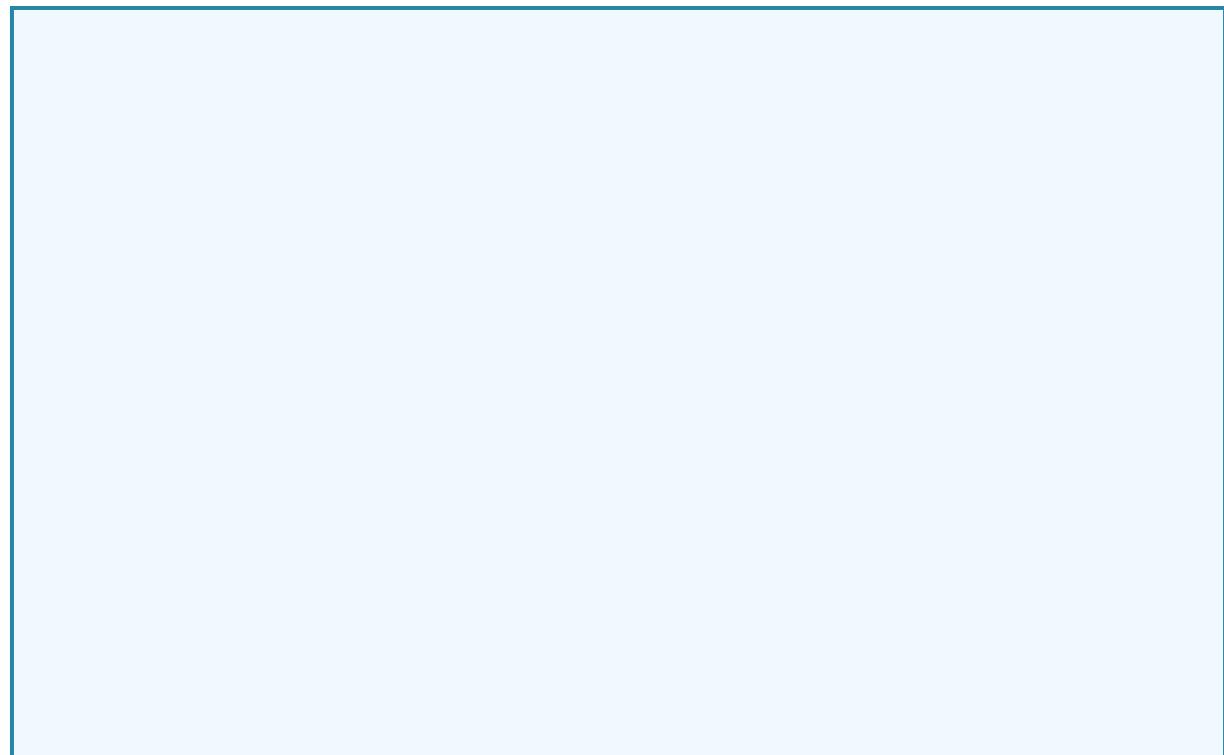
5. Donnez l'algorithme d'une fonction qui recherche la première occurrence d'une valeur dans une liste chaînée simple. La fonction renvoie la position dans la liste, ou -1 si la valeur n'est pas trouvée.

Entête de la fonction

```
1 FONCTION chercherElt(donnee : entier, liste : maillon, pos : entier) :  
    entier
```



1. Identifier les cas d'arrêts :
 - a. Lorsque je suis à la fin de la liste
 - b. Lorsque j'ai trouvé la bonne position
2. L'appel récursif :
 - a. j'augmente la position de +1
 - b. je passe au maillon suivant



Exercice 3 : Opérations avancées

1. Donnez l'algorithme permettant de supprimer le premier maillon dont la position est passée en paramètre.

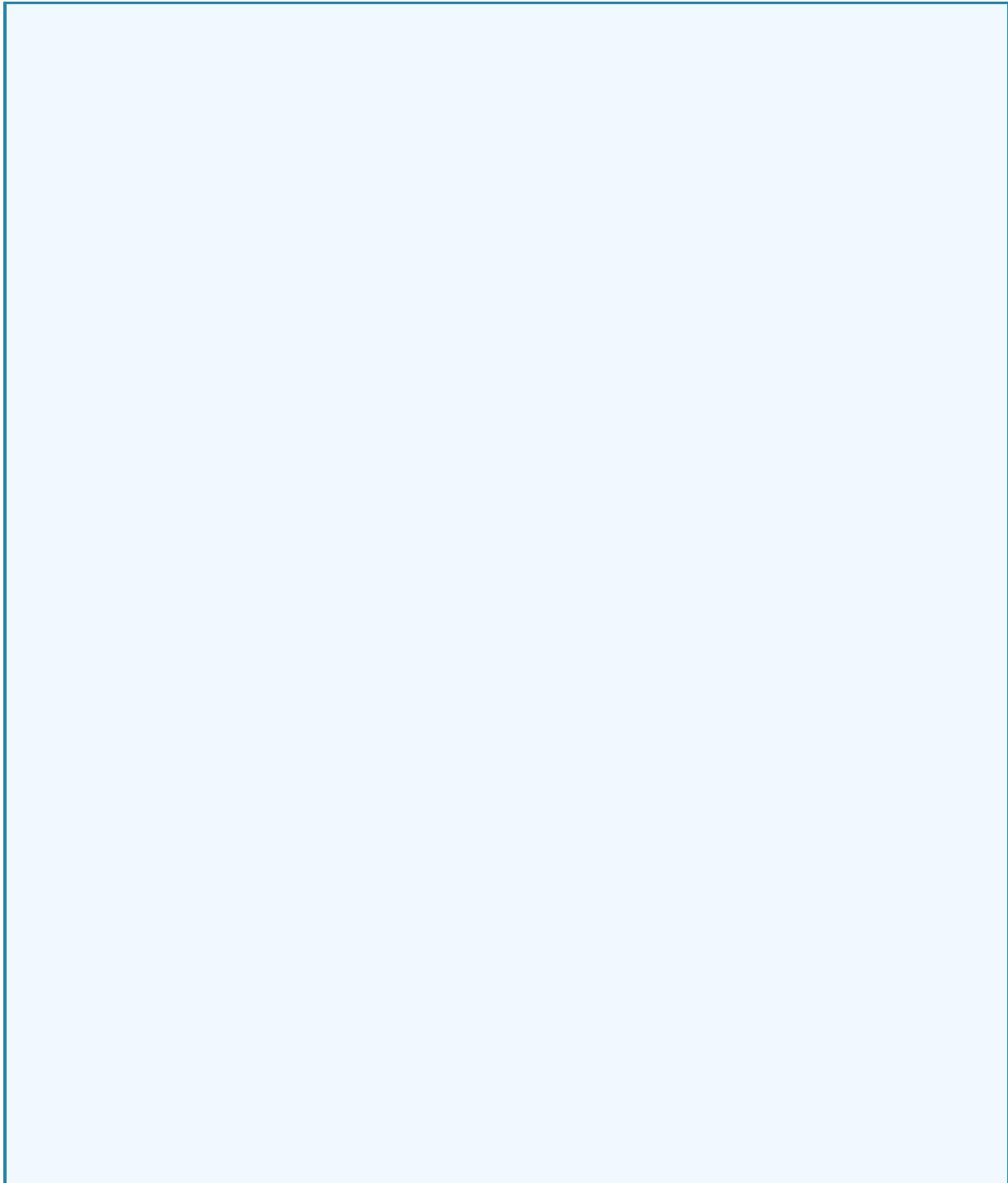
Entête de la fonction

```
1 FONCTION supprPos(maListe : maillon, pos : entier): maListe
```

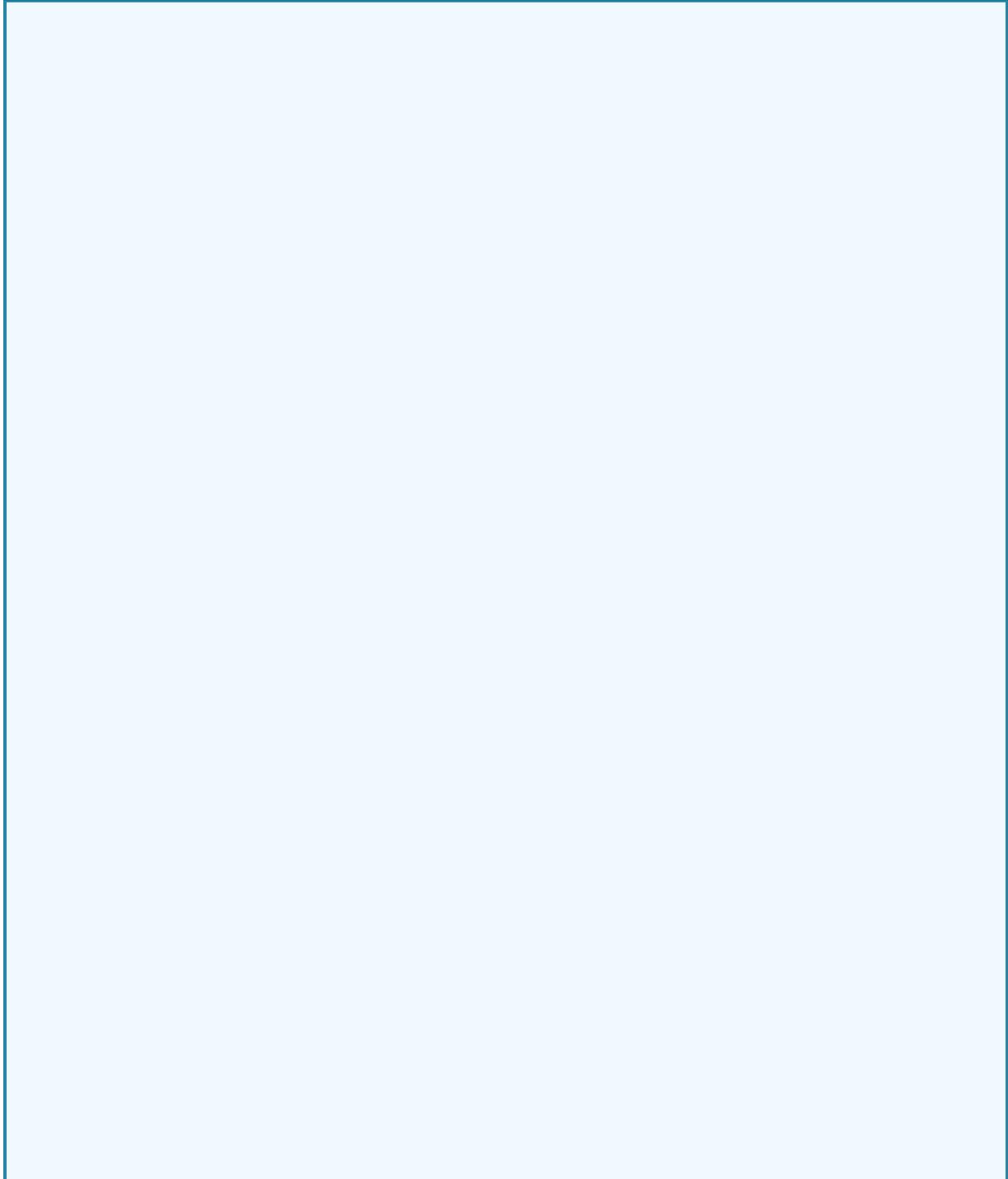


1. Identifier le cas d'arrêt : lorsque je suis à la fin de la liste
2. L'appel récursif :
 - a. je peux utiliser des méthodes que j'ai déjà codé
 - b. je peux passer de maillon en maillon et supprimer la tete de liste

2. Donnez l'algorithme permettant de supprimer tous les maillons dont la valeur est égale à une valeur donnée.



3. Donnez l'algorithme permettant de rattacher la liste à elle-même. Par exemple, {10, 15, 20} devient {10, 15, 20, 10, 15, 20}.



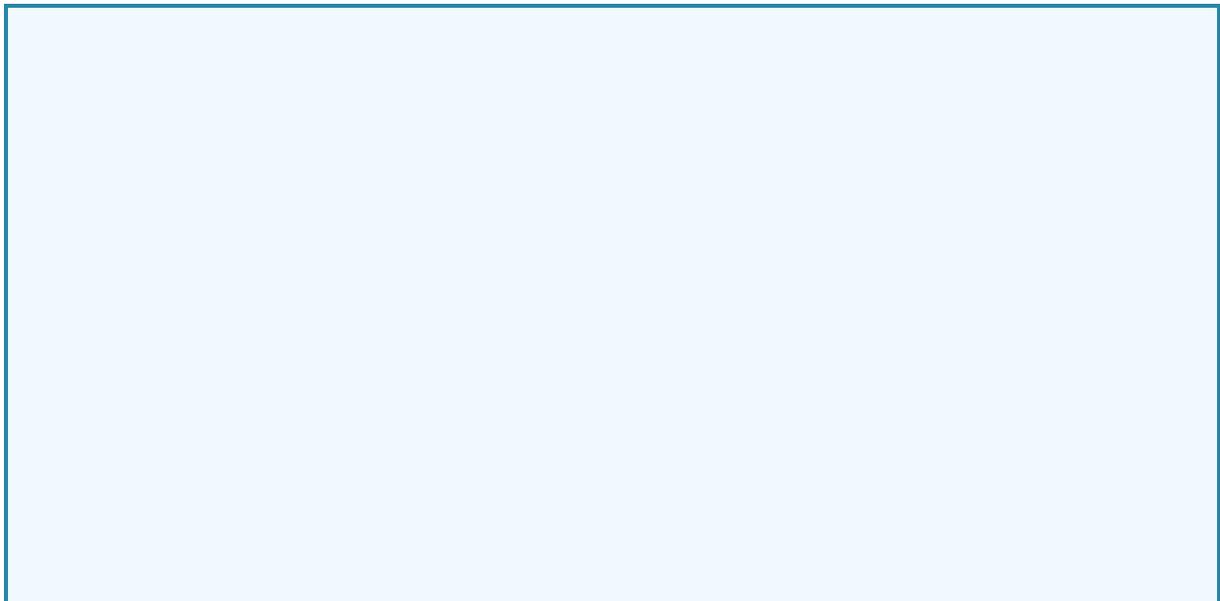
4. Donnez l'algorithme permettant d'inverser la liste. Par exemple, {1, 4, 3, 8} devient {8, 3, 4, 1}.



5. Rappelez l'algorithme permettant de rechercher un elt dans une liste. Si l'élément n'est pas dans la liste alors la fonction renvoie -1 sinon elle renvoie la position de l'élément dans la liste.

A large, empty rectangular box with a light blue background and a thin blue border, intended for the student to write the algorithm for searching an element in a linked list.

6. Donnez l'algorithme permettant de supprimer les doublons. Par exemple, {42, 24, 2, 2, 3, 24, 8} devient {42, 3, 8}

A large, empty rectangular box with a light blue background and a thin blue border, intended for the student to write the algorithm for removing duplicates from a linked list.