
TD2 - Les piles et les files

Inès de Courchelle, Elisabeth Ranisavljevic, Romain Dujol



2024-2025

**Objectifs :**

- Étudier des structures de données
- Connaître les fonctionnalités d'une pile et d'une file
- 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 piles et les files

Exercice 1 : Quiz

1. Que signifie l'abréviation FIFO ? Est-elle associée à la pile ou à la file ?

2. Que signifie l'abréviation LIFO ? Est-elle associée à la pile ou à la file ?

3. On considère la pile suivante :

```
1 haut de la pile -> 2
2                   3
3                   4
4                   5
5                   7
6                   8
7                   -----
```

Quel sera le contenu de la pile P après l'exécution des instructions suivantes ?

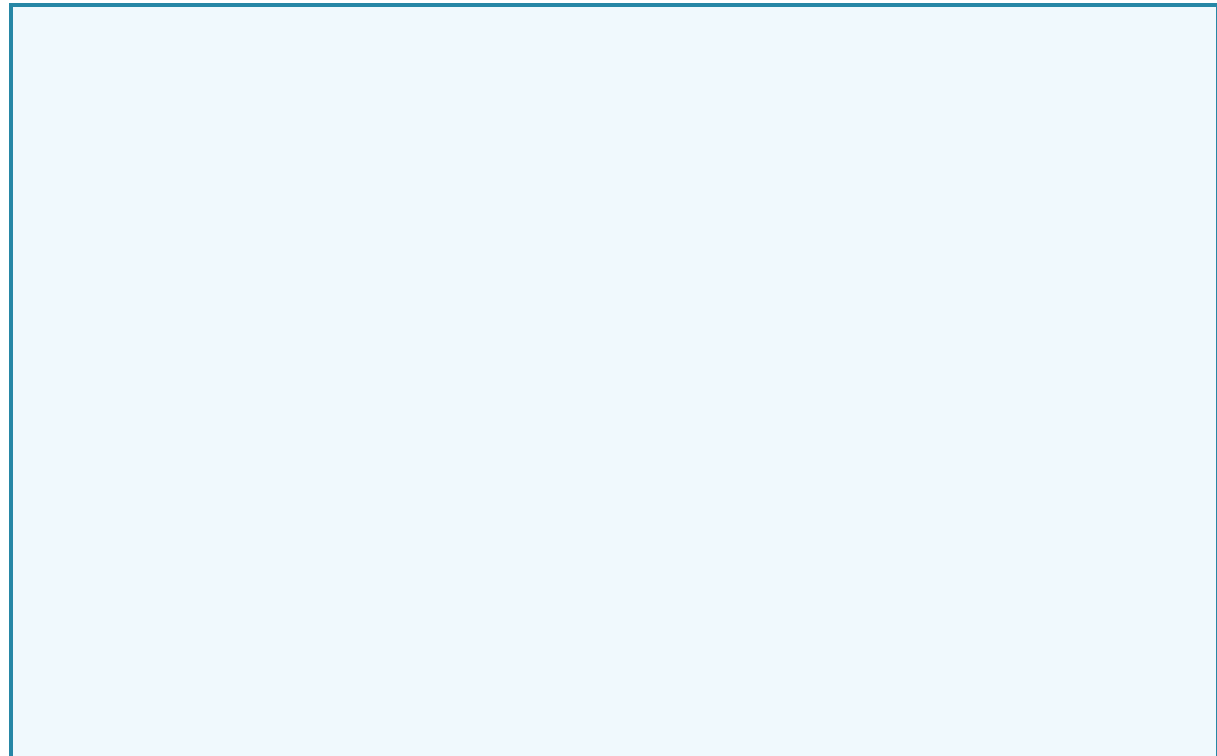
```
1 a <- depiler(p)
2 a <- depiler(p)
3 p <- empiler(p,5)
4 p <- empiler(p,15)
5 p <- empiler(p,24)
6 p <- empiler(p,42)
7 a <- depiler(p)
```

4. On considère la file suivante :

```
1  sens de la file -> 2 | 3 | 4 | 5 | 7 | 8 ->
```

Quel sera le contenu de la file f après l'exécution de l'algorithme suivant ?

```
1  f <- enfiler(f,12)
2  a <- defiler(f)
3  f <- enfiler(f,5)
4  f <- enfiler(f,15)
5  a <- defiler(f)
```



5. On considère l'architecture de fichier suivante : Où doit-on définir la structure en C ?

- a. maFile.h
- b. maFile.c
- c. main.c
- d. aucunes de ces réponses

Exercice 2 : Les piles

```
1 Fonction creerPileVide() : Pile
2 Fonction creerElementPile(e : ElementP) : ElementP
3 Fonction estPileVide(p : Pile) : Entier
4 Procédure empiler(p : Pile, e : ElementP)
5 Fonction depiler(p : Pile) : ElementP
6 Procédure viderPile(p : Pile)
7 Procédure afficherPile(p : Pile)
```

Écrire l'algorithme d'une fonction `fusionner` qui fusionne le contenu de 2 piles passées en paramètres.

exemple

```
1 p1 -> 2    p2 -> 4
2          3          5
3          4
4  -----
5
6 p3 -> 2
7          3
8          4
9          4
10         5
11  -----
```

a. La version itérative

b. La version récursive



1. Quand est ce que je m'arrête ? *Quand les deux piles sont vides*
2. Qu'est ce que je fais si la pile N°1 n'est pas vide ? *Je l'a dépile et j'empile son premier élément dans la pile*
3. Qu'est ce que je fais si la pile N°2 n'est pas vide ? *Je l'a dépile et j'empile son premier élément dans la pile*
4. Attention je n'oublie pas qu'il faut que je change le sens de la pile (comme avant)



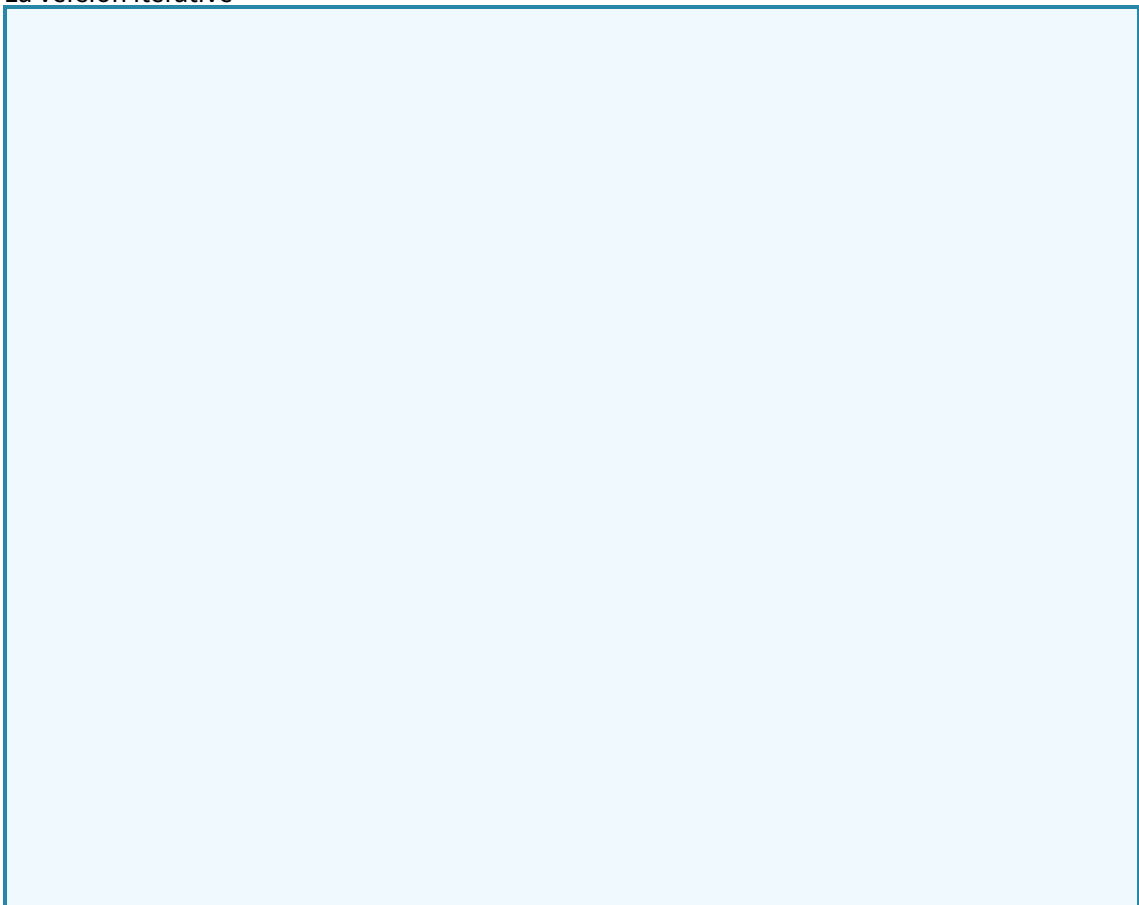
Exercice 3 : Les files

```
1 Fonction creerFileVide() : File
2 Fonction creerElementFile(e : ElementF) : ElementF
3 Fonction estFileVide(f : File) : boolean
4 Procédure enfiler(f : File, e : ElementF)
5 Fonction defiler(f : File) : ElementF
6 Procédure viderfile(f : File)
7 Procédure afficherFile(f : File)
```

Écrire l'algorithme d'une fonction Fusionner qui fusionne le contenu de 2 files passées en paramètres.

Exemple

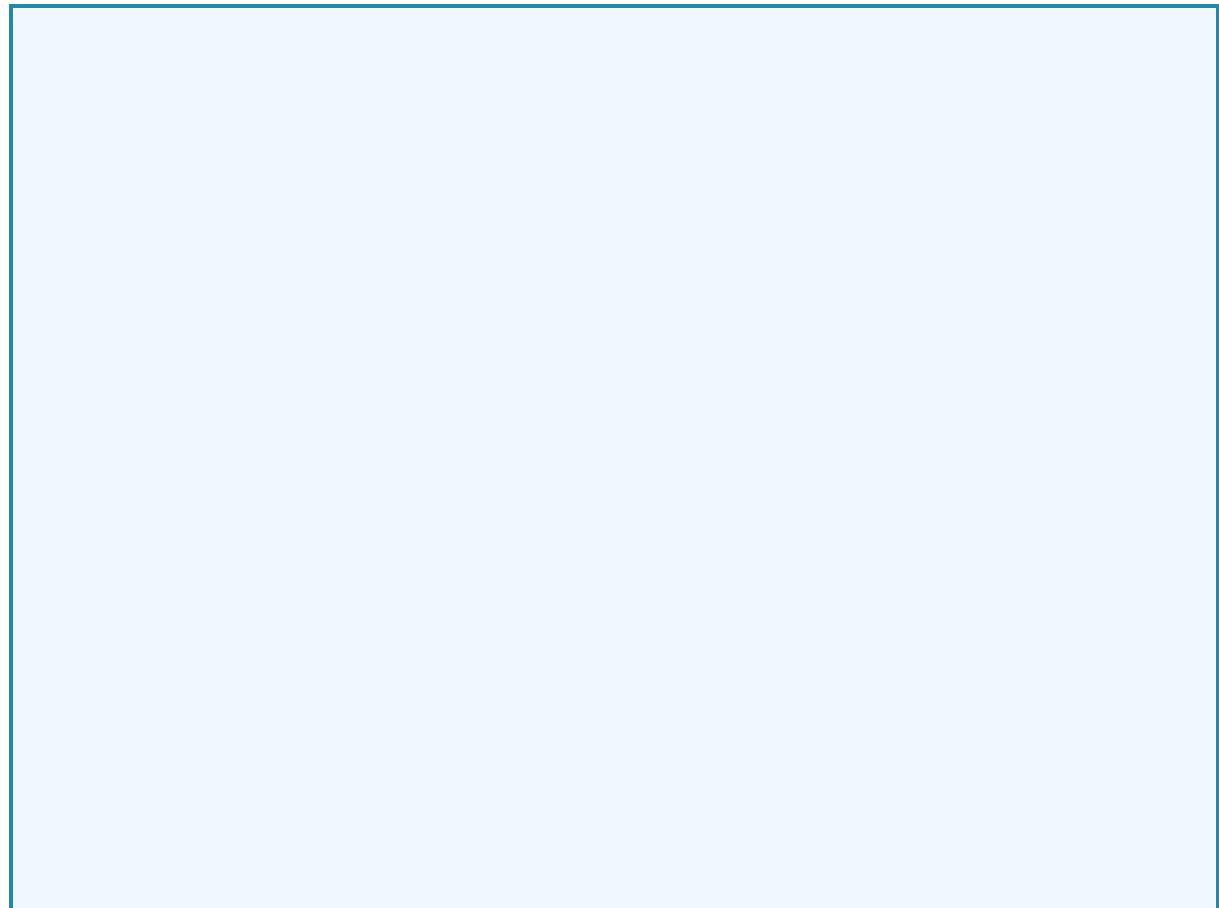
```
1 f1 -> 2 | 3 | 4 ->
2 f2 -> 6 | 7 | 9 ->
3
4 RESULTAT
5 -----
6 f3 -> 6 | 7 | 9 | 2 | 3 | 4 ->
```

a. La version itérative

b. La version récursive



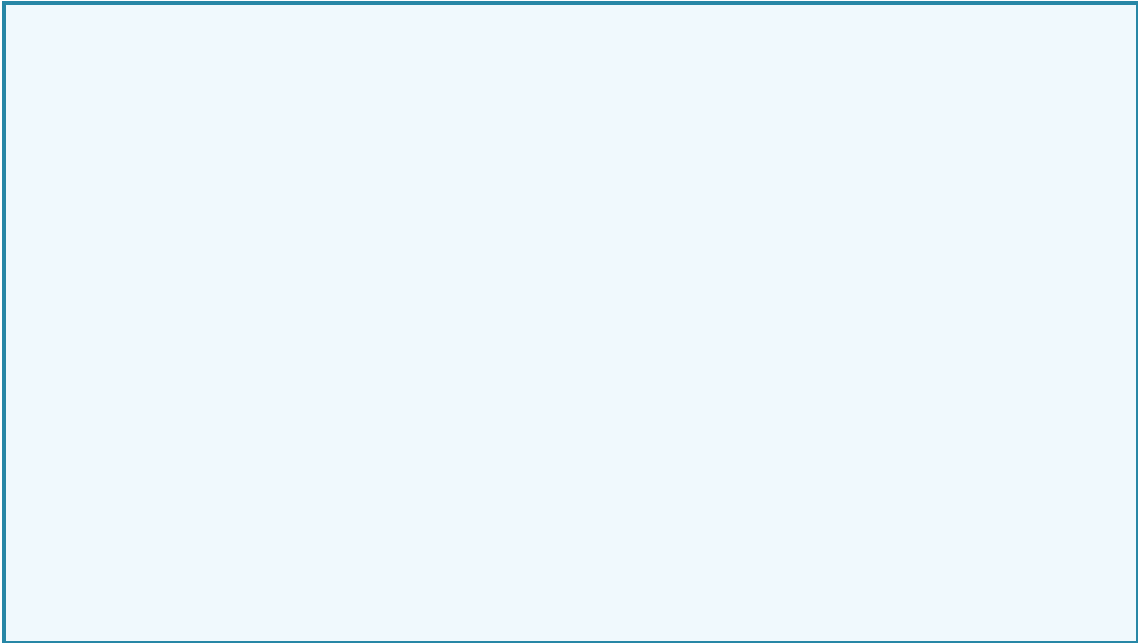
1. Quand est ce que je m'arrête ? *Quand les deux files sont vides*
2. Qu'est ce que je fais si la file N°1 n'est pas vide ? *Je l'a défile et j'enfile son premier élément dans la file*
3. Qu'est ce que je fais si la file N°2 n'est pas vide ? *Je l'a défile et j'enfile son premier élément dans la file*



Exercice 4 : Beaucoup de correction

Nous considérons un professeur enseignant plusieurs matières. Pour chaque matière, il y a une pile de copie à corriger. Chaque copie possède un nom, un prénom et une note.

1. Modéliser le problème de manière schématique (pas d'informatique !).



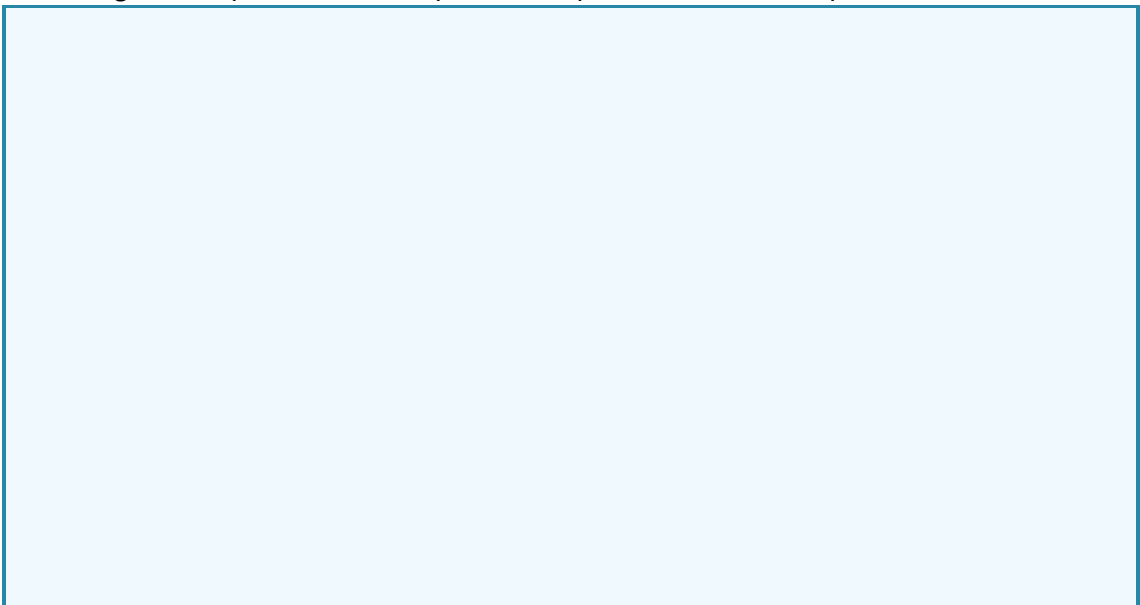
2. Implémenter les structures nécessaires.



3. Écrire l'algorithme permettant de créer une copie.



4. Écrire l'algorithme permettant d'empiler une copie dans un tas de copie.



5. Écrire l'algorithme permettant d'afficher un tas de copie.

6. Écrire l'algorithme permettant de compter les notes en dessous de 10. L'algo devra être écrit en récursif.



1. Identifier le critère d'arrêt. Quand est-ce que j'arrête de faire un appel récursif ?
J'arrête lorsqu'il n'y a plus de copie dans le tas.
2. Compter le nombre de copie supérieur à 10 dans un accumulateur