

Entraînement : listes chaînées

Les solutions peuvent être rédigées en **pseudo-code, python, C, C++ ou Java**. La syntaxe du langage n'a pas d'importance tant que celle-ci reste **cohérente** et **compréhensible**. (Dans les exemples, les solutions sont données en pseudo-code).

Grille d'évaluation

A (20)	L'algorithme répond au problème posé de façon claire et exhaustive.
B (16)	Le principe général de l'algorithme est le bon. Cependant, il y a une ou deux erreurs / oublis sur les cas particuliers ou les conditions d'arrêts et vérification de pointeurs nuls. Il peut y avoir des petites erreurs dans la manipulation de la liste (erreur de nom, confusion cellule / valeur)
C (11)	Le principe général de l'algorithme est le bon mais il y a de nombreuses erreurs ou oublis de cas particuliers.
D (8)	Le principe général de l'algorithme ne permet pas de répondre au problème, cependant les opérations de manipulation sur la liste chaînée sont écrites correctement.
E (1)	L'algorithme est faux ou inexistant et la manipulation de la liste n'est pas correcte.

Exercice 1.

Une **file** est une structure de données "First In, First out" : on fait sortir les éléments dans l'ordre dans lequel ils sont arrivés. Elle accepte deux opérations **enfile** qui rajoute un élément et **defile** qui supprime l'élément le plus ancien.

Exemple, si l'on part d'une file vide **F** :

```
enfile(F,2)
enfile(F,1)
enfile(F,3)
enfile(F,4)
Affiche defile(F)
Affiche defile(F)
Affiche defile(F)
Affiche defile(F)
```

affiche 2134.

Et

```
enfile(F,2)
enfile(F,1)
Affiche defile(F)
enfile(F,3)
enfile(F,4)
Affiche defile(F)
enfile(F,5)
Affiche defile(F)
Affiche defile(F)
Affiche defile(F)
```

affiche 21345.

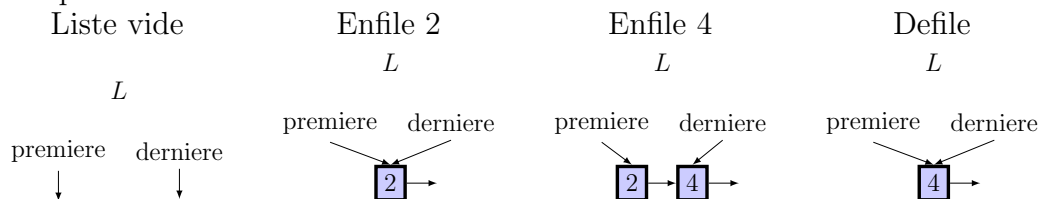
On propose d'implanter une File en utilisant une structure de liste chaînée. Cependant, contrairement à d'habitude, on va maintenir **deux pointeurs** : un sur la tête de liste et un sur le dernier élément de la liste.

```
Structure Cellule :
  Entier valeur
  Cellule suivante

Structure File :
  Cellule premiere
  Cellule derniere
```

On enfile en fin de liste et on défile en début de liste.

Exemple illustré.



Implantez les fonctions **Enfile**(File F, Entier a) (pas de valeur de retour) et **Defile**(File F) (renvoie un entier). On supposera que l'appel **Cellule(a)** permet de créer une cellule de valeur *a*.

Solution

```
Enfile
Input : File F, Entier a
Procédé :
  d <- F.derniere
  n <- Cellule(a)
  Si d != None :
    d.suivante <- n
    F.derniere <- n
  Sinon :
    F.premiere <- n
    F.derniere <- n

Defile
Input : File F
Output : un entier
Procédé :
  p <- F.premiere
  Si p = None :
    Erreur
  Sinon
    v <- p.valeur
    F.premiere <- p.suivante
    Si F.premiere = None :
      F.derniere <- None
  Retourner v
```