

# STRUCTURES DE DONNEES

## LISTES, PILES et FILES

### 1. Généralités

### 2. Notion d'interface

### 3. Les listes

### 4. Les piles

### 5. Les files

#### 5.1 Qu'est-ce qu'une file ?

Une file est une structure de données dans laquelle on accède aux éléments suivant la règle du « **premier arrivé, premier sorti** ».

On ne peut donc accéder qu'à l'objet situé au début de la file.

On qualifie ce comportement par l'expression « premier entré, premier sorti » **FIFO** en anglais : **F**irst **I**n, **F**irst **O**ut.

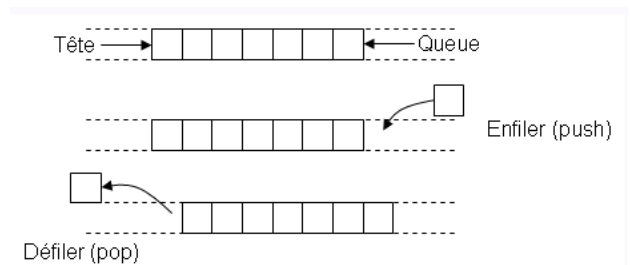
Le dernier élément entré dans la file est la « **queue** » de la file, la « **tête** » de la file est l'élément le plus ancien de celle-ci ?

Il est facile de faire le parallèle avec la vie courante. Quand vous allez prendre un billet de cinéma, vous faites la queue au guichet !

#### 5.2 Deux opérations élémentaires sur les files

ENFILER(F,x) qui correspond à l'insertion de l'élément x à la queue de la file F, si la file n'est pas pleine.

DEFILER(F) qui retire l'élément de tête de la file F et la retourne si la file n'est pas vide.



### 5.3 Autres exemples d'opérations sur les files

**Voici d'autres opérations possibles sur une file. En vous aidant de ce qui a été fait ci-dessus, expliquez ce que font ces opérations.**

- CREER\_FILE\_VIDE()
- EST\_VIDE(F)
- ENFILER(F,e)
- EST\_PLEINE(F)
- DEFILER(F)

### 5.4 Exemple d'application

Soit la suite d'instructions suivantes :

```
F=CREER_FILE_VIDE()
ENFILER(F, 21)
ENFILER(F, 22)
ENFILER(F, 23)
N =DEFILER(F)
ENFILER(F, 24)
ENFILER(F, 25)
N =DEFILER(F)
```

**Ecrire le résultat de la file obtenue à chacune de ces instructions.**

Instructions	Résultat
F=CREER_FILE_VIDE()	
ENFILER(F, 21)	
ENFILER(F, 22)	
ENFILER(F, 23)	F = (21, 22, 23)
N =DEFILER(F)	
ENFILER(F, 24)	
ENFILER(F, 25)	
N =DEFILER(F)	

**Que contient la file créée à la fin de ces instructions ?**

**Quelle valeur a pris la variable N ?**

## 5.5 Représentation d'une file avec un tableau

### 5.5.1 Exemple

On réalise une file capable de contenir  $n$  éléments en utilisant un tableau constitué de  $(n+3)$  éléments.

- La première case du tableau (indice 0) contient l'indice de la tête de la file.
- La deuxième case du tableau (indice 1) contient l'indice de la queue de la file.
- La troisième case du tableau (indice 2) contient le nombre d'éléments présents dans la file.
- Les cases suivantes du tableau (indices 3 à  $n+2$ ), contiennent les éléments de la file ou sont vides.

`F = CREER_FILE_VIDE()`

Tête	Queue	Taille	file				
none	3	0					
0	1	2	3	4	5	6	7

`ENFILER(F, 8)`

Tête	Queue	Taille	file				
3	4	1	8				
0	1	2	3	4	5	6	7

`ENFILER(F,3)`

Tête	Queue	Taille	file				
3	5	2	8	3			
0	1	2	3	4	5	6	7

`ENFILER(F, 5)`

Tête	Queue	Taille	file				
3	6	3	8	3	5		
0	1	2	3	4	5	6	7

`DEFILER(F)`

Tête	Queue	Taille	file				
4	6	2	8	3	5		
0	1	2	3	4	5	6	7

Tout comme les piles, bien que l'élément « 8 » apparaisse encore dans le tableau, il n'est plus dans la file.

L'indice de tête est donc « 4 », et la taille de la file est bien « 2 ».

### **5.5.2 Pseudo-code**

Ecrire les algorithmes correspondant aux fonctions ENFILER(F, x) et DEFILER(F).

### **5.5.3 Implémentation en python**

Implémenter les deux fonctions ci-dessus en langage python

Puis écrire le programme principal qui permet de réaliser l'exemple du paragraphe 5.5.1