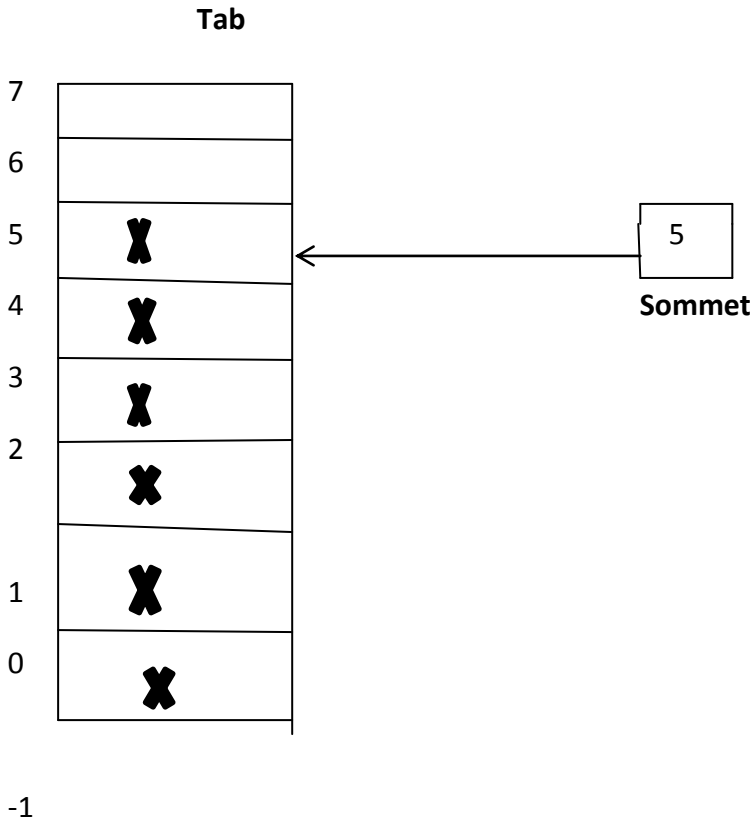
La pile sous forme d'un tableau nommé Tab et son indice appelé Sommet.

Cette pile est dite P.

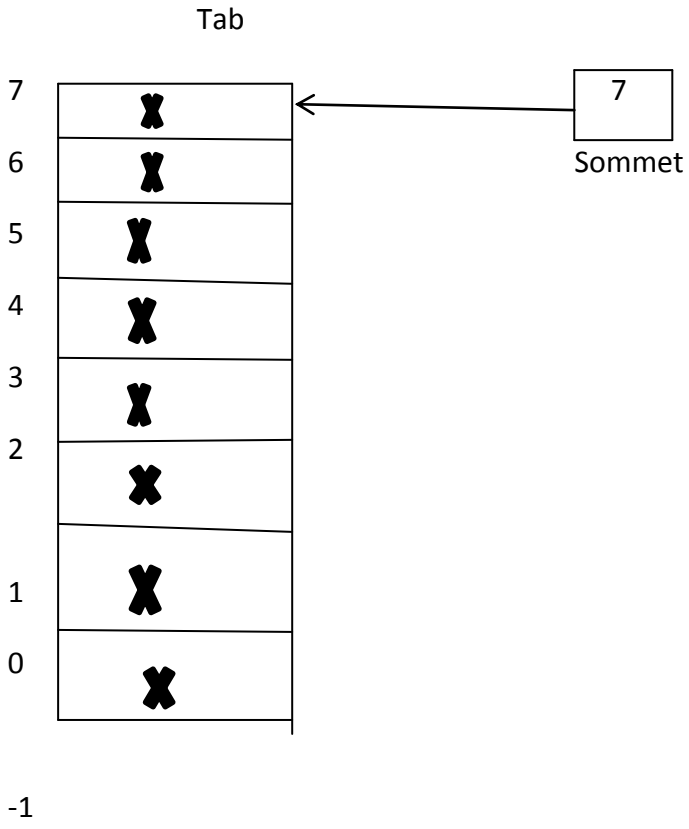


Cette pile P contient six (06) éléments car sommet = 5.

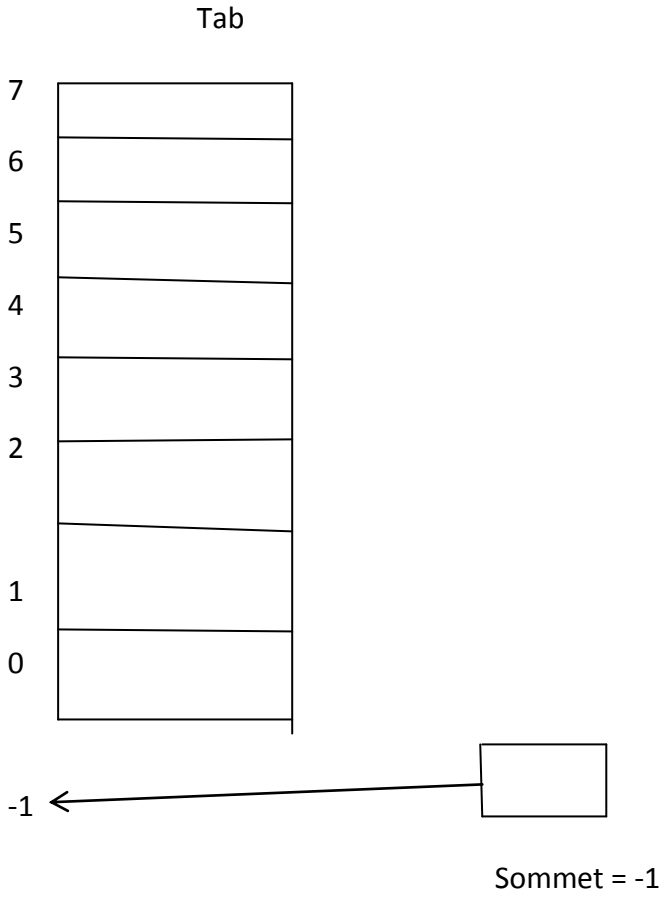
Et elle peut contenir au maximum huit (08) éléments (Maxpile = 8)

```
/* Implémentation*/
#define Maxpile 8      /*On suppose que la pile contient 8 éléments de 0 à 7*/
Typedef  struct  pile  pile ;
Struct   pile    {   int  sommet;
                     Item   Tab[Maxpile];                /* item est un type à préciser*/
                 };
Pile  P;
```

La pile qui suit est pleine : Sommet = Maxpile -1. C'est-à-dire les cases de 0 à 7 sont occupées.



Le schéma qui suit montre une pile vide : Sommet = -1. C'est-à-dire toutes les cases sont inoccupées.



Par conséquent c'est très simple la pile P est :

- Pleine quand Sommet = Maxpile – 1 et
- Vide quand Sommet = -1.

La pile sous forme d'une structure chaînée P.

Deux cas de figure se présentent ou bien, on utilise des nœuds dont le champ adresse est celle du nœud suivant (Figure 1) ou bien ce même champ contient l'adresse du nœud précédant (Figure 2). On utilisera le premier cas et les résultats seront les mêmes ; c'est une question de vue d'angle ni plus ni moins.

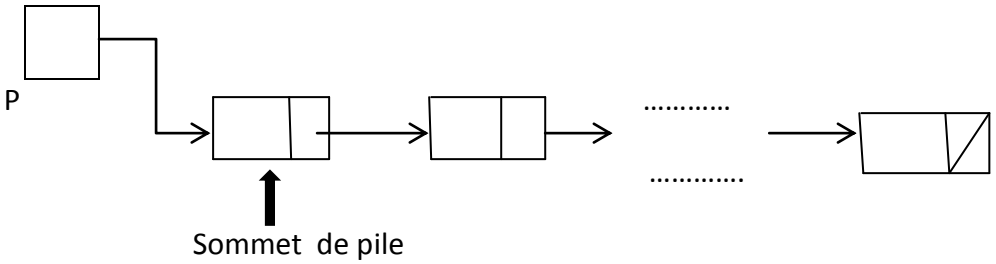


Figure 1

Ou bien

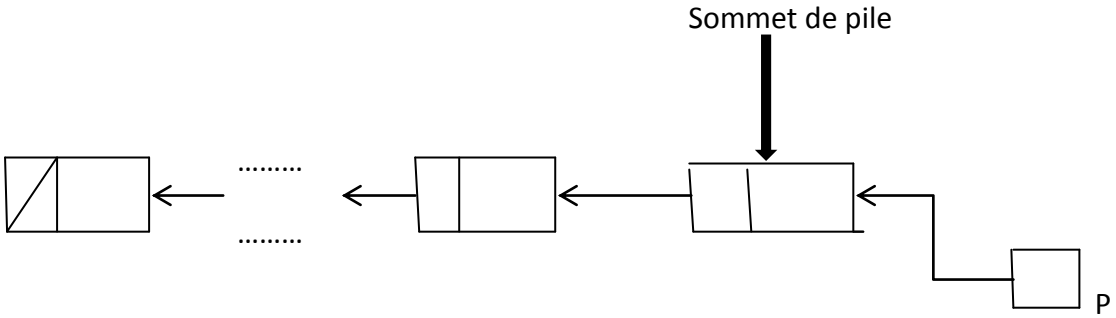
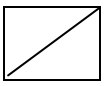


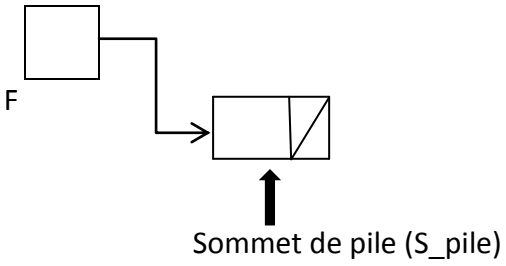
Figure 2

Si la pile est vide, on aura :



P = NULL

Si la file contient un seul nœud, on aura :



/* Implémentation*/

```
#include <stdlib.h>

Typedef struct Pile {
                    Item info ;          /*item un type à définir*/
                    Struct Pile *suiv ;  /*Autres littératures utilisent prec au lieu */
                   } Pile ;              /*de suiv tout dépend de l'angle de vue !*/

Pile *P = NULL ;
```

NB : prec comme précédant, tout comme suiv pour suivant.

Tous ces schémas vont nous aider à écrire les opérations de base en langage C:

- Implémentation de la Pile P.
- Son initialisation.
- Quand est-ce qu'elle est vide.
- Quand est-ce qu'elle est pleine.
- Empiler une information (Push).
- Dépiler une information (Pop).

Que ce soit la pile est statique ou dynamique.

File d'attente sous forme d'un tableau circulaire

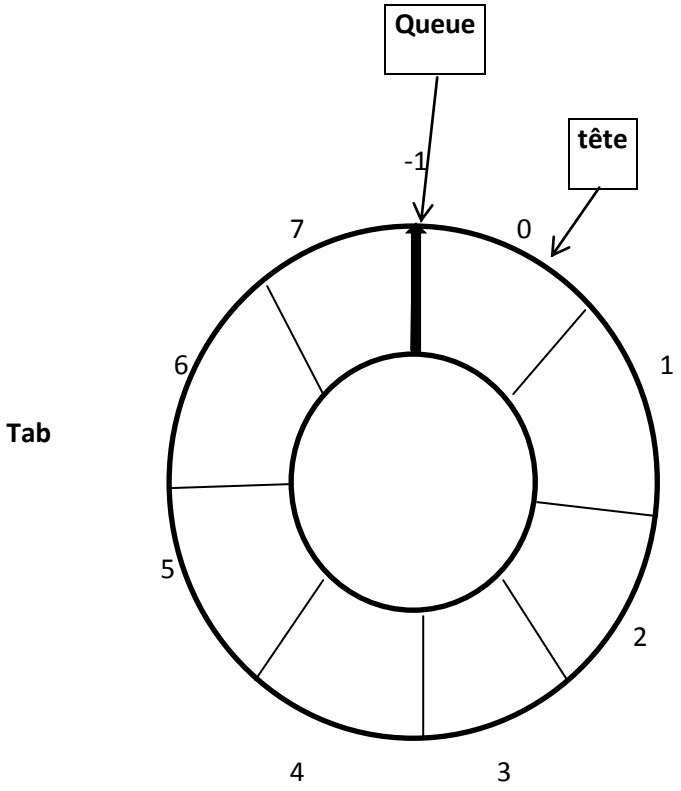


Figure 1

Le tableau est appelé Tab et les deux indices Tête et Queue ; et la file d'attente qui est constituée du tableau et ses deux indices sera appelée F avec Maxfile = 8.

```
/*Implémentation d'une file d'attente F statique*/

#define Maxfile 8              /* on suppose qu'il y ait  8 éléments de 0 à 7*/

Typedef   struct  fila  fila ;     /*fila pour ne pas confondre avec le mot clé FILE Fichier*/

 Struct  fila     {                /* quoique FILE est écrit en majuscule */

                    Int tete ;

                    Int queue ;

                    Item  Tab[Maxfile] ; /* item type à définir*/

                  } ;

fila  F ;
```

En plus le schéma du dessus (figure 1) représente une file d'attente vide :

Tête = 0 et Queue = -1

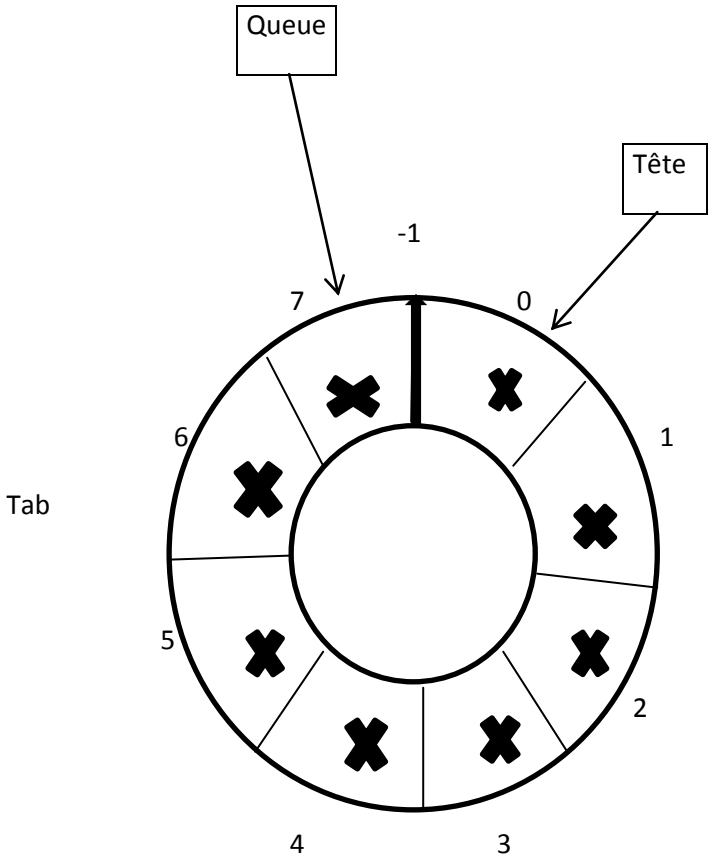


Figure 2

Ce schéma (figure 2) représente une file d'attente pleine : Tête = 0 et Queue = Maxpile – 1 étant donné que Maxpile=8 (C'est-à-dire 8 cases de 0 à 7).

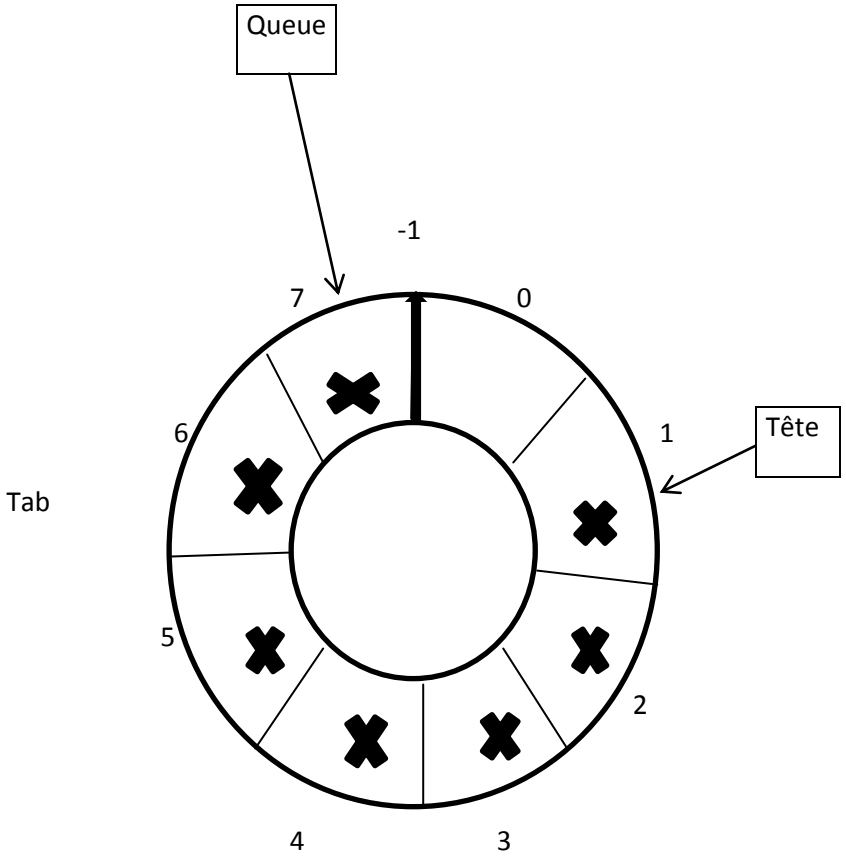


Figure 3

Ce schéma représente une file d'attente qui n'est pas pleine car la case N° 0 est non occupée donc on peut en rajouter un. Mais comment ?

Tout simplement Queue doit être égale à 0, donc quand elle n'est pas pleine et

Queue= Maxfile-1, donc Queue passe à 0. C'est le modulo en mathématique.

C'est-à-dire : Queue ← (Queue + 1) % Maxfile ;

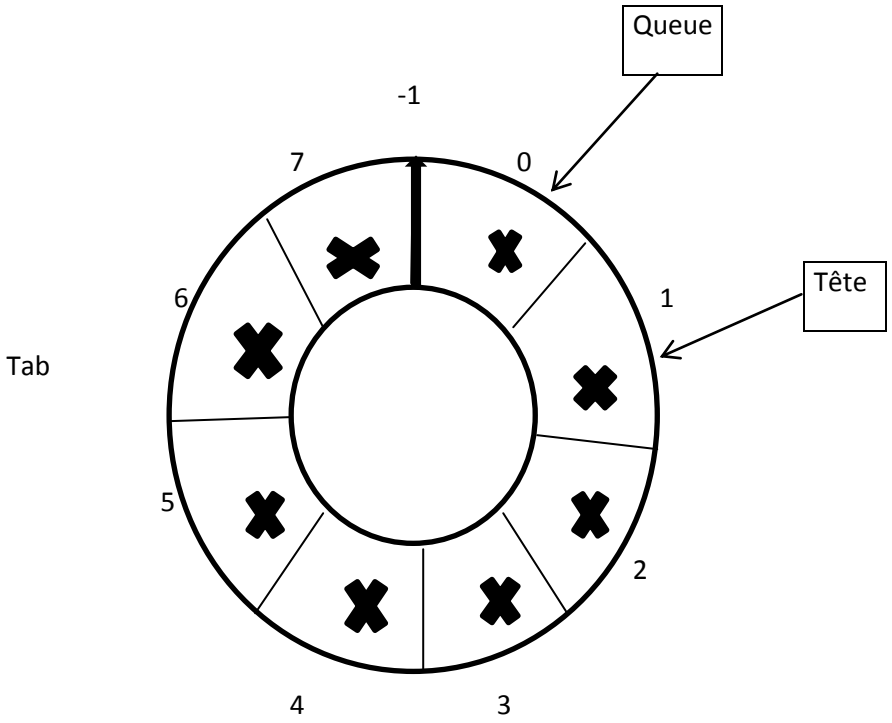


Figure 4

Cet autre schéma représente une file d'attente pleine mais c'est un autre cas : Tête = Queue + 1. Mais attention ne pas confondre avec le cas où Queue = -1 et Tête = 0 car ce cas aussi on aura :

Tête = Queue+1 et c'est le cas quand la file est vide !

Conclusion :

- Quand (Tête = 0 et Queue = -1) on dit que la file est vide.
- Et elle est pleine quand ((Tête = 0 et Queue = Maxfile-1) ou (tête = queue+1 et Queue≠-1).

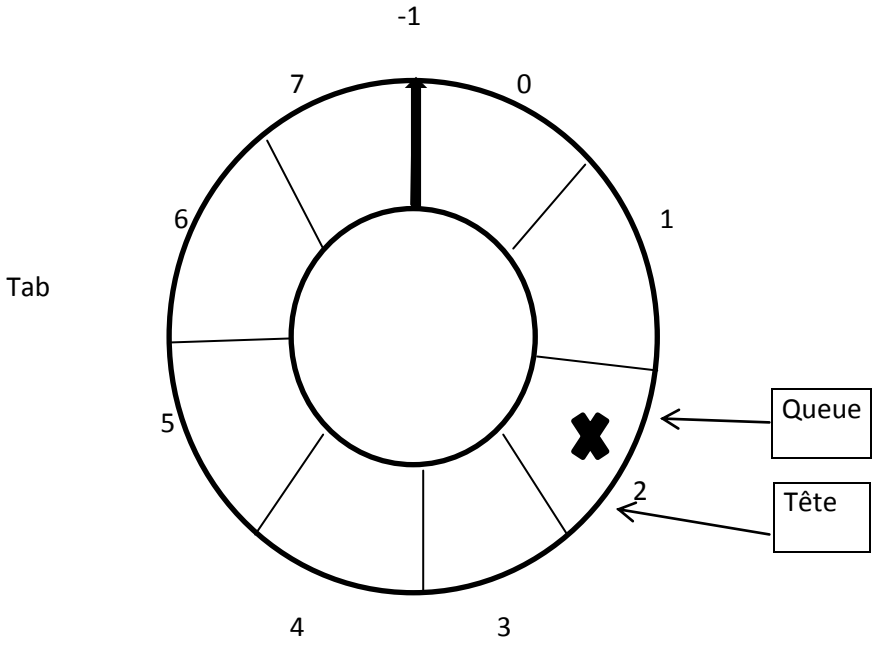
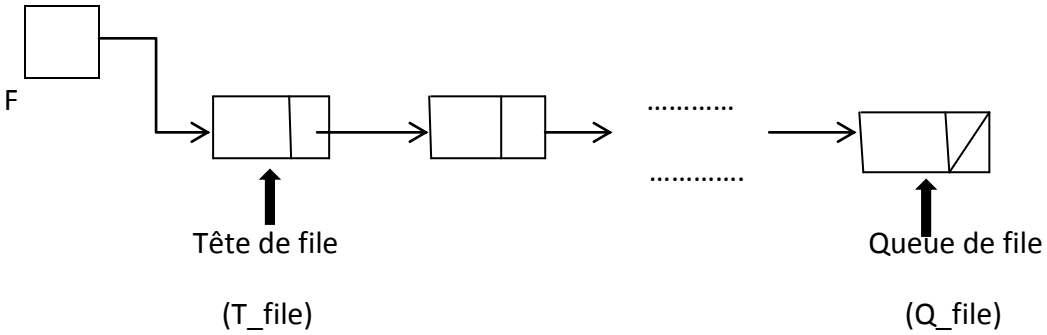


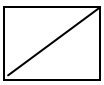
Figure 5

Cet autre schéma nous montre que si Tête = Queue et quelle que soit le N° de la case, comme dans ce cas égale à 2 : On constate que la file contient un seul élément, donc si on le retire elle devient vide ! C'est-à-dire il faut réinitialiser la file d'attente donc Tête = 0 et Queue = -1.

File d'attente sous forme d'une structure chaînée.

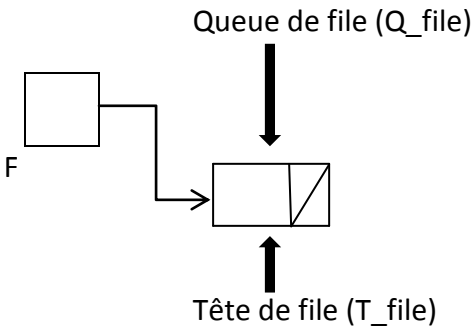


Si la file est vide, on aura :



F = NULL

Si la file contient un seul nœud, on aura :



C'est-à-dire que Tête = Queue.

```
/*Implémentation d'une file d'attente F */

        #include <stdlib.h>

        Typedef         struct   fila     {

                                        Item  info;   /* item un type à préciser*/

                                        Struct  fila   *suiv ;

                                        } fila;

        Fila   *F = NULL ;
```

Tous ces schémas vont nous aider à écrire les opérations de base en langage C :

- Implémentation de la fille d'attente F.
- Son initialisation.
- Quand est-ce qu'elle est vide.
- Quand est-ce qu'elle est pleine.
- Enfiler une information (Enqueue).
- Défiler une information (Dequeue).

Que ce soit la file d'attente est statique ou dynamique.

Après cette étude sur des schémas concernant les piles et les files d'attentes que ce soit statique sous forme de tableaux, ou dynamique sous forme de listes chaînées, on va essayer d'écrire quelques opérations de base les concernant en langage C.

On trouve cela sur le fascicule intitulé « LangageCpartieVI ».