

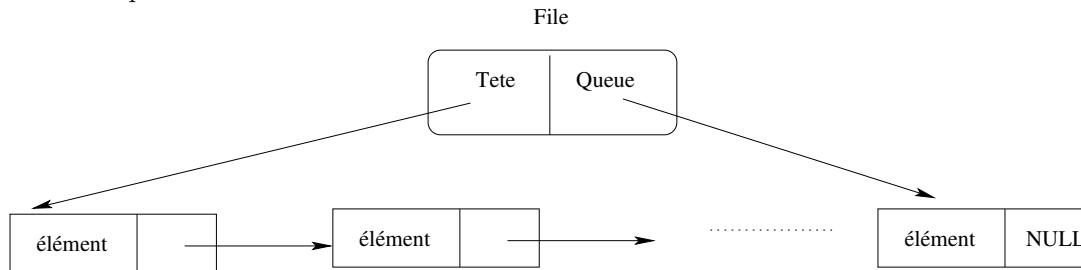
SRI 1ère année - file dynamique

1 Première partie : file dynamique d'éléments

Il s'agit d'implanter la structure de données MA_FILE (FIFO - first in, first out - le premier entré est le premier sorti) :

- insérer un élément consiste à ajouter un élément en queue de file,
- retirer un élément consiste à supprimer l'élément en tête de file.

Une file est dite dynamique quand on n'utilise que la place mémoire correspondant au nombre *réel* d'éléments mémorisés. Donc la place en mémoire occupée par la file va évoluer au cours de l'exécution du programme. Pour cela, une file dynamique sera représentée par une structure contenant 2 pointeurs, un vers la tête de la file et un vers la queue de la file.



Créer l'unité File Dynamique et le fichier de test associé qui permettent de :

1. Définir le type MA_FILE. Votre file doit pouvoir contenir des éléments de type quelconque définis à l'avance, pour cela, il faudra utiliser le type ELEMENT de l'unité **e**lément définie aux TPs précédents (avec un type ELEMENT correspondant au type entier dans le cadre des tests).
2. Définir la fonction INIT_FILE qui initialise la file (donnée en paramètre).
3. Définir une fonction AFFICHE_FILE qui permet d'afficher tous les éléments d'une file (donnée en paramètre).
4. Définir une fonction FILE_EST_VIDE qui permet de tester si la file (donnée en paramètre) est vide.
5. Définir une fonction ENFILER qui permet d'enfiler un élément (donné en paramètre) à une file (donnée en paramètre), cette fonction doit placer l'élément en queue de file. Voir remarque ci-après.
6. Définir une fonction DEFILER qui permet de défiler une file (donnée en paramètre), cette fonction doit renvoyer aussi l'élément qui était en tête de file. Voir remarque ci-après.
7. Définir une fonction SAISIR_FILE qui permet de saisir une file (i.e. qui demande à l'utilisateur d'entrer les éléments un par un et les insère dans la file).

Vous devez tester TOUTES vos fonctions en envisageant TOUS LES CAS possibles.

Remarque : Les procédures ENFILER et DEFILER doivent être efficaces, c'est-à-dire qu'elles doivent faire le moins d'opérations possibles.

2 Deuxième partie : application

Soit une caisse de cinéma. On veut écrire le programme en C qui calcule le nombre de tickets vendus et la valeur de la caisse en fin de journée sachant que :

- en début de journée, la caisse est vide,
- il y a plusieurs séances par jour mais il s'agit d'un nombre fixe,

- chaque séance correspond à une file de clients,
- chaque client correspond à un nombre de tickets et un prix total correspondant à l'achat de tous ces tickets.

3 Troisième partie (optionnelle) : file statique d'éléments

Il s'agit toujours d'implanter la structure de données MA_FILE (FIFO - first in, first out - le premier entré est le premier sorti), mais maintenant de manière statique.

Une file est dite statique quand la place qu'elle occupe en mémoire est toujours la même quel que soit le nombre réel d'éléments dans la file.

Pour cela, les éléments de la file seront mémorisés dans un tableau de taille fixée (TMAX) d'éléments.

Remarque : Les procédures ENFILER et DEFILER doivent toujours être efficaces, c'est-à-dire qu'elles doivent faire le moins d'opérations possibles. Cette remarque est particulièrement importante dans le cas statique !

Créer l'unité File Statique et le fichier de test associé qui permettent de :

1. Définir le type MA_FILE. Votre file doit pouvoir contenir des éléments de type quelconque définis à l'avance, pour cela, il faudra utiliser le type ELEMENT de l'unité **element** définie aux TPs précédents (avec un type ELEMENT correspondant au type entier dans le cadre des tests).
2. Définir la fonction INIT_FILE qui initialise la file (donnée en paramètre).
3. Définir une fonction AFFICHE_FILE qui permet d'afficher tous les éléments d'une file (donnée en paramètre).
4. Définir une fonction FILE_EST_VIDE qui permet de tester si la file (donnée en paramètre) est vide.
5. Définir une fonction FILE_EST_PLEINE qui permet de tester si la file est pleine.
6. Définir une fonction ENFILER qui permet d'enfiler un élément (donné en paramètre) à une file (donnée en paramètre), cette fonction doit placer l'élément en fin de file. Voir remarque ci-dessus.
7. Définir une fonction DEFILER qui permet de défiler une file (donnée en paramètre), cette fonction doit renvoyer aussi l'élément qui était en tête de file. Voir remarque ci-dessus.
8. Définir une fonction SAISIR_FILE qui permet de saisir une file (i.e. qui demande à l'utilisateur d'entrer les éléments un par un et les insère dans la file).

Vous devez tester TOUTES vos fonctions en envisageant TOUS LES CAS possibles.

4 Quatrième partie (optionnelle) : application

Reprendre le programme de calcul pour le cinéma écrit à la section 2 et le faire tourner avec des files statiques. Si vous avez bien "travaillé", vous ne devriez avoir QU'UNE SEULE modification à y faire.