

Nom : \_\_\_\_\_

Prénom : \_\_\_\_\_

Numéro étudiant :

--	--	--	--	--	--	--	--

# Les files

7 novembre 2022

## A lire absolument :

1. L'objectif n'est pas d'apprendre la correction par cœur, mais de comprendre les mécanismes mis en œuvre. Cela vous permettra de vous adapter face à un problème nouveau.
2. En particulier, vous devez être capable de refaire l'intégralité du sujet, seul, sans aucune aide ni support.
3. Votre travail sera corrigé automatiquement par l'outil de correction automatique CAT. Cela implique que vous devez respecter scrupuleusement les consignes de chaque exercice. Faites très attention aux messages qu'il vous est demandé d'afficher. Un espace en trop, un saut de ligne en moins et l'exercice risque d'échouer.
4. L'enseignant voit votre activité sur le site, ainsi que l'historique de vos dépôts. Pensez à déposer votre travail régulièrement afin qu'il puisse vous apporter des conseils personnalisés.
5. Si l'enseignant vous demande de rendre votre travail sur papier, vous devez répondre directement sur le sujet en respectant absolument la zone prévue à cet effet. Tout ce qui se trouve en dehors de la zone sera ignoré.
6. Si le sujet contient un QCM, vous devez colorier les cases avec un stylo bleu ou noir. Les autres couleurs seront ignorées.
7. Chaque feuille est identifiée de manière unique. Vous pouvez donc rendre votre sujet avec les feuilles mélangées, mais il est préférable de les trier car cela vous permet de vérifier que vous n'en avez pas oublié une.
8. Si vous faites face à un problème, un bug, une erreur ou que vous souhaitez participer à l'amélioration de la plateforme, envoyez un mail à l'adresse suivante : [support-cat@liste.parisnanterre.fr](mailto:support-cat@liste.parisnanterre.fr)

Ne rien écrire dans cette zone



## Définition d'une file

Nous allons définir une structure 'struct file' permettant de manipuler une file. Pour cela, nous allons utiliser un tableau qui sera alloué dynamiquement. Nous aurons donc besoin d'un champs 'tableau' (un pointeur d'entier) et d'un champs 'taille\_tableau' qui contiendra le nombre de cases du tableau. Pour éviter de devoir décaler l'ensemble des valeurs à chaque opération de défilement, nous allons gérer ce tableau de manière circulaire. Cela signifie que nous allons avoir besoin de deux champs 'debut' et 'fin'. Le champs 'debut' indique l'indice de la case où se trouve le premier élément de la file. Le champs 'fin' indique l'indice de la première case libre après les éléments présents dans la file. Renommez ensuite cette structure en 'file'.



Ne rien écrire dans cette zone



## Initialisation d'une file

Écrire une fonction qui alloue dynamiquement de la mémoire pour une file. Vous devrez ensuite initialiser les différents champs de la file. En particulier, vous devrez initialiser le tableau de la file en allouant dynamiquement de la mémoire pour un tableau de 'taille' entiers. Le champs 'debut' doit être initialisé à 0, tout comme le champs 'fin'.

```
1 file * initialisation_file(int taille);
```

Ne rien écrire dans cette zone



## La file est vide ?

Écrire une fonction qui retourne 1 si la file est vide, 0 sinon.

```
1 int file_vide(file f);
```

Ne rien écrire dans cette zone



## La file est pleine ?

Écrire une fonction qui retourne 1 si la file est pleine, 0 sinon. Pour cela, vous devez utiliser les champs 'fin' et 'debut'.

```
1 int file_pleine(file f);
```

Ne rien écrire dans cette zone



# Enfiler

Écrire une fonction qui ajoute un élément dans la file, puis retourne 1. Si la file est pleine, retourner -1. N'oubliez pas d'utiliser le champs 'fin' qui indique l'indice de la prochaine case vide. Vous devrez ensuite incrémenter ce champs de 1. Si la valeur déborde du tableau, faites la revenir à 0.

```
1 int enfiler(file * f, int element);
```

Ne rien écrire dans cette zone



# Defiler

Écrire une fonction qui enlève un élément dans la file, puis retourne cet élément. Si la file est vide, retourner -1. N'oubliez pas d'utiliser le champs 'debut' qui indique l'indice de la première valeur présente dans la file. Vous devrez ensuite incrémenter ce champs de 1. Si la valeur déborde du tableau, faites la revenir à 0.

```
1 int defiler(file * f);
```

Ne rien écrire dans cette zone



## Affichage des valeurs d'une file

Écrire une fonction qui affiche tous les éléments présents dans une file. Pour cela, vous commencerez à l'indice 'debut', puis terminerez à l'indice 'fin'. Attention, le tableau étant circulaire, il se peut que 'debut' se trouve après 'fin' dans le tableau. L'affichage doit être semblable à cet exemple : |6|3|4|2|3|8|

```
1 void affichage_file(file f);
```

Ne rien écrire dans cette zone





# Libération de la mémoire de la file

Écrire une fonction qui libère la mémoire de la file. Attention, pensez également à libérer la mémoire du tableau alloué dynamiquement.

```
1 void liberation_file(file * f);
```

Ne rien écrire dans cette zone

