

## Algorithmique 2 - L2, S4

### TD 1 - Piles et Files

Pour chacune de vos fonctions vous préciserez la complexité en temps.

#### Exercice 1.1 *Entraînement*

Illustrer le résultat de chacune des opérations suivantes.

1. depiler (depiler (depiler (empiler (empiler (depiler(empiler (empiler (creerPile (),1),2)), 3),4)))).
2. defiler (defiler (defiler (enfiler (enfiler (defiler(enfiler (enfiler (creerFile (),2),3)), 1),4)))).

#### Exercice 1.2 *Implémentation du type Pile*

Proposer une implémentation du type `Pile_Entier` à l'aide d'un tableau d'entiers. Il faudra gérer les cas où le tableau est plein.

#### Exercice 1.3 *Implémentation de 2 piles adjacentes*

1. Expliquer comment implémenter deux piles d'entiers dans un seul tableau de manière qu'aucune pile ne déborde (il faudra aussi gérer le cas où le nombre d'éléments des 2 piles excède la taille du tableau). Vous donnerez le pseudo-code de chacune des opérations. Pour chacune des opérations on donnera le numéro de la pile où on effectue l'opération.
2. En utilisant les fonctions de la question précédente, donnez le pseudo-code qui permet de faire passer le sommet de la pile 1 sur la pile 2.

#### Exercice 1.4 *Implémentation du type File*

1. Expliquez le problème pour implémenter une file avec un tableau lorsque l'on ne considère pas ce dernier comme circulaire.
2. Proposer une implémentation du type `File_Entier` à l'aide d'un tableau d'entiers. Il faudra gérer les cas où le tableau est plein.

#### Exercice 1.5 *Utilisation Piles et Files*

1. Montrer comment implémenter une pile à l'aide de deux files.
2. Montrer comment implémenter une file à l'aide de deux piles.

#### Exercice 1.6 *2 files d'attente*

On veut simuler des files d'attentes d'un supermarché à l'aide d'un seul tableau, et pour simplifier on va considérer que l'on ne dispose que de deux caisses (lorsqu'il n'y a pas moyen de faire une insertion on affichera un message pour dire qu'il faut une troisième caisse). On rappelle qu'une insertion se fera toujours dans la file la plus courte, et si de même taille dans la première.

1. Comme pour le cas des piles, donnez le pseudo-code qui permet de gérer les 2 files.
2. Ecrire le pseudo-code qui permet d'insérer un consommateur dans la file d'attente la plus courte.

**Exercice 1.7\*** *File à Double Entrée*

Une file à double entrée est une file où on autorise l'insertion et la suppression à chaque bout.

1. Expliquez comment ce TDA est plus général que les TDAs pile et file.
2. Proposer une implémentation du type `FileDoubleEntree_Entier` à l'aide d'un tableau.
3. Une telle file peut-être restreinte en spécifiant le bout où on peut faire des insertions ou des suppressions. Pour chacune des variantes, dire si on peut l'utiliser pour simuler les TDAs pile et file.