

*Travaux Dirigés de Structures de Données
[TD n°2 : Types Abstraits de Données : Piles & Files]*

Objectifs : - Concrétisation des piles
- Nouvelles représentations de files

Exercice 1

On s'intéresse au **TAD Pile**, défini dans le cours. Les primitives du TAD Pile sont supposées déjà définies.

- Ecrire un algorithme (*ou une fonction C*), **taille_pile**, qui calcule le nombre d'éléments dans une pile donnée.
- Ecrire un algorithme (*ou une fonction C*), **copie_pile**, qui retourne une copie d'une pile donnée.

Exercice2

Utiliser une pile pour évaluer une expression arithmétique postfixée codée sur un tableau de caractères, en supposant pour simplifier que: tous les opérateurs sont binaires et limités à +, -, * et / . On utilise uniquement des nombres entiers naturels

Indication : On utilise la pile pour stocker les opérandes. Dès qu'on lit une opération, on récupère les deux opérandes sur le dessus de la pile, on effectue l'opération et on met le résultat sur la pile. A la fin de la lecture de l'expression, il ne doit rester dans la pile qu'une seule valeur: le résultat du calcul.

Note : on se donne la primitive « estopérande » pour tester si un caractère est opérande ou pas de même la primitive « effectueropération » pour effectuer une opération arithmétique entre deux opérandes.

Exercice 3

Ecrire une primitive (ou algorithme) qui retourne une file inversée.

Note : La représentation du TAD file par tableau circulaire est conseillée.

Exercice 4

Nous avons également vu en cours une implémentation d'une file par un tableau circulaire. Dans cet exercice, nous allons implémenter une file de taille N à l'aide de deux piles de taille N. L'idée est la suivante : Le sommet de la première pile correspond à l'avant de la file, tandis que le sommet de la seconde pile correspond à l'arrière de la file. Lorsque la première pile est vide, la seconde peut être retournée sur la première. Ce retournement est utilisé pour retirer l'élément en tête de file lorsque la pile correspondant à l'avant est vide.

- Ecrire toutes les primitives standards concernant une file, à savoir
Création d'une file ; tester si une file vide ou pleine ; retourner la tête d'une file.
- Ecrire la primitive enfiler (F) correspondante à cette nouvelle conception de la file
- De même écrire la primitive défiler (F).

Note : on pourra utiliser toutes les primitives du cours aux pile et files