

## Types Abstraits & Base de la POO - TD 2



### Types de données Abstraits Linéaires

#### I – Les classiques...

Les exercices suivants vont vous permettre de vous aguerrir dans l'implémentation de structures dynamiques. En effet, nous allons réaliser l'implémentation de différentes structures abstraites.

##### Exercice I.1

#### Deque pour une pile et une file

1. En vous basant sur la description faite en cours, créer un module `deque.h/deque.c` réalisant l'implémentation du TDA Deque. Vous respecterez les points suivants :
  - vous réaliserez une implémentation avec sentinelle
  - vous ajouterez une fonctionnalité permettant l'affichage complet de la Deque
  - pour les plus téméraire, vous réaliserez une implémentation générique du TDA Deque
2. Créer un programme principal de test (`test-deque.c`) utilisant votre module afin de tester les différentes fonctionnalités de ce dernier.

Votre module `deque.h/deque.c` maintenant au point, nous allons nous en servir pour créer 2 nouveaux modules.

3. Proposer un module `pile.h/pile.c` se basant au maximum sur le module `deque.h/deque.c`
4. Proposer un module `file.h/file.c` se basant au maximum sur le module `deque.h/deque.c`
5. Réaliser le même travail dans les langages C++ & Ada.

#### II – Un peu plus complet. . .

##### Exercice II.1

#### Liste chaînée

On souhaite maintenant créer un module `liste.h/liste.c` implémentant le TDA Liste.

1. Réaliser l'implémentation doublement chaînée du TDA Liste. Vous choisirez la version générique<sup>1</sup> ou contenant des entiers.
2. Ajouter une opération permettant d'afficher le contenu d'une liste.
3. Réaliser un programme principal `test-liste.c` mettant en œuvre chacune des fonctionnalités d'une liste.
4. Réaliser le même travail dans les langages C++ & Ada.



1. de préférence ... vous serez mieux préparés pour la suite !