

## TRAVAUX PRATIQUES – Semaine n°2

### Thèmes

- Conteneur généralisé
- Réutilisation du conteneur et adaptation de programme

### Exercice 1. Définition et documentation de fonctions

Les sources de programme du TD2 se trouvent sur COMMUN (SDA/Sem02/TP).

Rappel : Codage des fonctionnalités (`initialiser`, `ecrire`, `lire`, `afficher`) d'un conteneur généralisé à tout type d'item. Le type du conteneur est le suivant :

```
struct ConteneurTS {  
    enum {CAPACITE = 15};  
    Item tab[CAPACITE];  
    unsigned int nbItems;  
};
```

Ajoutez un projet/solution SDA-Sem02-TP auquel vous ajouterez un projet Exo1. Ajoutez au projet un fichier (d'extension `.cpp`) où vous recopierez les sources sur COMMUN.

1.1. Que fait le programme ? Compilez, exécutez et testez le programme.

1.2. Complétez les fonctions associées au conteneur de type `ConteneurTS` par la définition documentée de la fonction `extrema` suivante :

La fonction `extrema`, indique les extrema du conteneur d'items par les deux positions dans le conteneur : l'une correspondant à la plus petite valeur des éléments du conteneur, l'autre à la plus grande valeur. Plusieurs éléments du conteneur peuvent avoir la même valeur, la position d'un extremum correspondra à l'index le plus petit dans le conteneur où a été détecté l'extremum. Donner la précondition de cette fonction.

Vérifiez par assertion la fonction `extrema`. Vous vérifierez entre autres que les index retournés pour le minimum et le maximum sont bien les résultats attendus.

Indication pour le codage de `extrema` : on utilisera la fonction `comparer` à deux arguments (`arg1` et `arg2`) de type `Item` qui renvoie

- (1) un nombre négatif si `arg1 < arg2`,
- (2) 0 si `arg1 = arg2`
- (3) un nombre strictement positif si `arg1 > arg2`.

Avant de l'utiliser, prototypez la fonction `comparer`.

### Exercice 2. Première application du conteneur généralisé

Utilisez un conteneur de type `ConteneurTS` pour afficher la plus petite et la plus grande taille des étudiants de votre TP.

Spécification : la taille considérée (e.g. 1.70) est donnée en mètre. Vous collecterez au tableau la taille des étudiants du TP pour l'initialisation du conteneur.

Ajoutez un autre projet à votre solution (Exo2) et ajoutez un fichier source (`.cpp`). Recopiez le source de Exo1 dans le nouveau fichier source.

Quelles sont les adaptations à faire pour développer l'application demandée ?

### Exercice 3. Deuxième application du conteneur généralisé

Soit le type `Date` suivant :

```
/* Type Date */  
struct Date {  
    unsigned int jour; // le jour  
    unsigned int mois; // le mois  
    unsigned int année; //l'année  
};
```

En réutilisant le source du type `ConteneurTS` et toutes ses fonctionnalités (cf. exercice 1) vous adapterez le source à l'application qui consiste en l'affichage de la date de naissance du/des plus jeune(s) et du/des plus vieil (vieux) étudiant(s) du TP pour laquelle vous collecterez au tableau la date de naissance des étudiants du TD.

Quelles sont les adaptations spécifiques à faire pour développer l'application demandée ?

Codez l'application.