FACULTÉ DES SCIENCES ET TECHNOLOGIES | Université de Lille

L3 EEA/IE – Informatique pour l'EEA
TP 2 – Manipulations de tableaux

1 Exercice 1

Soit un vecteur V de taille N et constitué d'éléments de nature entière :

```c
#include <stdio.h>
#include <stdlib.h>
#define N 10

int main()
{
    int V[N];

    /* a completer */

    return 0;
}
```

Il vous est demandé de trier ce vecteur par la méthode de tri à bulles vue en TD. Il en existe plusieurs variantes, mais l'idée commune consiste à échanger 2 éléments consécutifs s'ils ne sont pas ordonnés. L'opération est répétée jusqu'à ce que tous les éléments soient rangés dans l'ordre souhaité. Ainsi, dans le cas d'un tri par ordre croissant parcourant le vecteur du début vers la fin, les éléments les plus grands seront déplacés vers la fin de la suite, case après case, à la manière des bulles d'air remontant à la surface d'un liquide. Après un premier parcours, l'élément le plus grand se retrouvera donc à la fin du tableau ; après un second parcours, les deux plus grands sont 'rangés', etc. Plusieurs passes seront donc nécessaires pour trier entièrement le vecteur. Une certaine latitude vous est laissée ici pour optimiser cette opération de tri.

On demande donc :

- une fonction d'initialisation du vecteur `InitV` avec des valeurs arbitraires entières positives,
- une fonction d'affichage du vecteur `AffichV`, qui présente toutes les valeurs sur une seule ligne,
- une fonction de tri `TriV`, triant entièrement le vecteur et faisant appel à chaque passe à la fonction précédente pour contrôle,
- le programme principal `main` faisant appel successivement aux trois fonctions décrites.

Toutes les variables doivent être locales à des fonctions. Donner la traduction en langage C de ces fonctions et du programme principal.

2 Exercice 2

On dispose de 2 tableaux représentés par les variables `TAB1` et `TAB2`. Dans chacun de ces 2 tableaux, les valeurs sont triées par ordre croissant. L'objectif ici est de coder en langage C un programme (fonctions et programme principal) permettant de :

- trouver les valeurs communes des 2 tableaux (`TAB1` et `TAB2`) et de recopier ces éléments communs dans un troisième tableau (représenté par la variable `TabDoublon`) également trié par ordre croissant. Cette recherche doit se faire impérativement en parcourant les 2 tableaux « au fur et à mesure » de l'analyse. Les valeurs communes ne seront recopiées qu'une seule fois dans `TabDoublon`.
- supprimer les valeurs communes dans les tableaux `TAB1` et `TAB2` et « tasser » les valeurs restantes dans ces tableaux.

Exemple :

- `TAB1` = {1, 3, 6, 9},
- `TAB2` = {3, 8, 9},
- `TabDoublon` = {3, 9}.

Tableaux `TAB1` et `TAB2` après recherche des doublons :

- `TAB1` = {1, 6},
- `TAB2` = {8} .

3 Exercice 3

On dispose de 2 tableaux représentés par les variables `TAB1` et `TAB2`. Dans chacun de ces 2 tableaux, les valeurs sont triées par ordre croissant. L'objectif ici est de coder en langage C un programme (fonctions et programme principal) permettant de fusionner les 2 tableaux (`TAB1` et `TAB2`) en un troisième (représenté par la variable `TAB3`) également trié par ordre croissant. L'opération de fusionnement consiste à obtenir un tableau résultat (`TAB3`) comprenant tous les éléments des 2 tableaux initiaux, sans modifier ces derniers. La réalisation du fusionnement doit se faire impérativement en triant au fur et à mesure les éléments issus de chacun des tableaux `TAB1` et `TAB2`.

Remarque : Si les tableaux `TAB1` et `TAB2` contiennent une même valeur, celle-ci ne sera recopiée qu'une fois dans le tableau `TAB3`.

Exemple :

- `TAB1` = {1, 3, 6, 9},
- `TAB2` = {3, 8, 9},

Tableau `TAB3` après fusion : `TAB3` = {1, 3, 6, 8, 9}.

4 Rendu

Le compte rendu ne concerne que le code C **soigneusement commenté** et sera à rendre à votre encadrant à la fin de la séance. Le fichier devra s'appeler TP2_NOM-binome1_NOM-binome2.c (les trinômes ne sont pas acceptés). La compilation ne devra produire aucune erreur ni aucun avertissement. **Tout plagiat d'un code qui a été produit par un autre binôme sera sévèrement sanctionné.**