

Exercice 13

OBJECTIFS : TABLEAUX, FONCTIONS ET POINTEURS

A l'issue de la réalisation de cet exercice, les étudiants doivent être capables de manipuler des tableaux en relation avec des fonctions et des pointeurs.

PREPARATION DU PROJET

L'exercice 2 sert de mode d'emploi pour la création du projet.

- Lancez le "Microsoft Visual Studio 2010
- Créez un nouveau projet :
 - Projets Visual C++
 - Application console Win32 (Dans la fenêtre modèle)
 - Sous emplacement: introduisez votre répertoire d'exercices
 - Décocher "Créer le répertoire pour la solution"
 - Sous Nom du projet : **Ex13**
- Paramètre de l'application :
 - Application Console
 - Projet Vide
- **Copie du fichier Ex13.c :**
Il faut copier le fichier Ex13.C de K:\ES\Maitres-Eleves\SLO\Modules\SL124_LOGA\Exercices\Ex13 dans le répertoire Ex13. (Usage de l'explorateur Windows)
- Ajoutez au projet le fichier Ex13.c :
 - Depuis le Visual Studio 2010, sélectionnez "**Fichiers Sources**", avec un clic-droit obtenez le menu pour Ajouter un élément existant.
- Test de compilation : vous devez obtenir :
Génération : 1 a réussi, 0 a échoué, 0 a été ignore

DONNEES DU PROBLEME

Le fichier fournis est un canevas vide, supportant la structure des deux tests ainsi que les demandes à l'utilisateur.

ACTION DU TEST A

Il faut déclarer dans le programme principal un tableau de 10 double, nommé **TableA**.

En utilisant une boucle, il faut initialiser le tableau avec des valeurs variant de 10.00 à 10.09. L'initialisation est à placer au début du main.

Pour afficher l'adresse et la valeur de chaque élément, il faut réaliser la fonction **ShowItems** qui reçoit 2 valeurs :

- Le nombre d'éléments du tableau
- L'adresse du début du tableau.

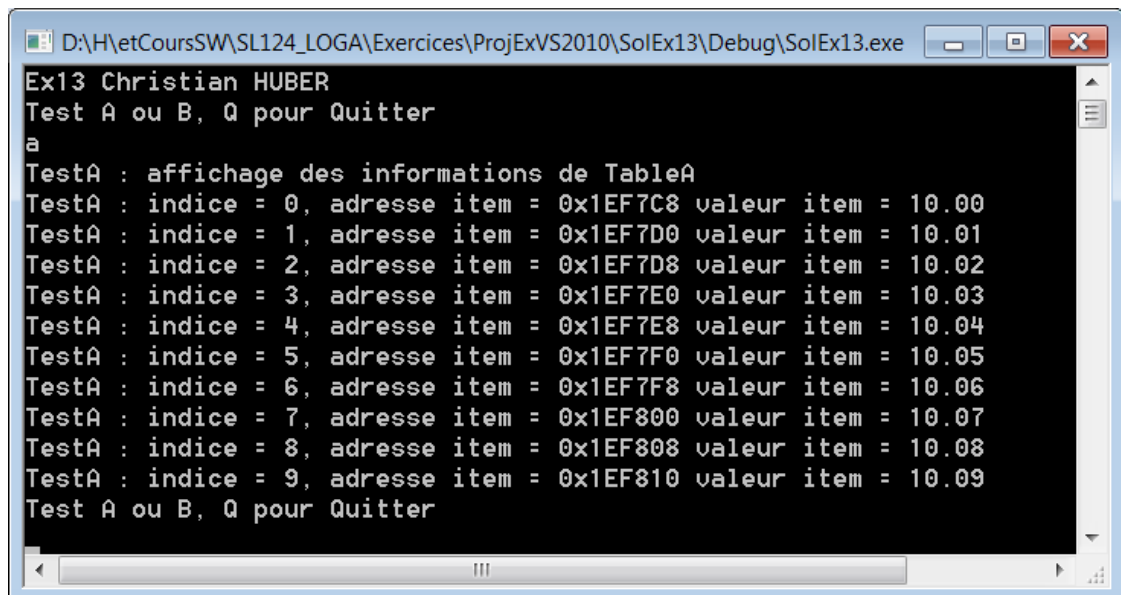
La fonction est responsable de l'affichage du résultat, dont chaque ligne a la forme suivante :

TestA : indice = xxx, adresse item = 0xaaaaaa valeur item = vv.vv

Remarque: adresse item à afficher en hexadécimal.

La fonction **ShowItems** doit afficher en utilisant un pointeur que l'on incrémente pour passer d'un élément du tableau au suivant (pas d'utilisation d'indice).

Exemple de résultat :



```
D:\H\etCoursSW\SL124_LOGA\Exercices\ProjExVS2010\SolEx13\Debug\SolEx13.exe
Ex13 Christian HUBER
Test A ou B, Q pour Quitter
a
TestA : affichage des informations de TableA
TestA : indice = 0, adresse item = 0x1EF7C8 valeur item = 10.00
TestA : indice = 1, adresse item = 0x1EF7D0 valeur item = 10.01
TestA : indice = 2, adresse item = 0x1EF7D8 valeur item = 10.02
TestA : indice = 3, adresse item = 0x1EF7E0 valeur item = 10.03
TestA : indice = 4, adresse item = 0x1EF7E8 valeur item = 10.04
TestA : indice = 5, adresse item = 0x1EF7F0 valeur item = 10.05
TestA : indice = 6, adresse item = 0x1EF7F8 valeur item = 10.06
TestA : indice = 7, adresse item = 0x1EF800 valeur item = 10.07
TestA : indice = 8, adresse item = 0x1EF808 valeur item = 10.08
TestA : indice = 9, adresse item = 0x1EF810 valeur item = 10.09
Test A ou B, Q pour Quitter
```

ACTION DU TEST B

Il faut déclarer dans le programme principal un tableau a 2 dimensions de 16 lignes et de 4 colonnes, chaque élément est du type char, la matrice est nommée MatriceB.

Initialisez la matrice lors de la déclaration de telle manière à ce que l'on obtienne la table suivante :

No Ligne	[3] [2] [1] [0]	No Ligne	[3] [2] [1] [0]
0	0 0 0 0	8	1 0 0 0
1	0 0 0 1	9	1 0 0 1
2	0 0 1 0	10	1 0 1 0
3	0 0 1 1	11	1 0 1 1
4	0 1 0 0	12	1 1 0 0
5	0 1 0 1	13	1 1 0 1
6	0 1 1 0	13	1 1 1 0
7	0 1 1 1	15	1 1 1 1

Remarque: utilisation des valeurs numérique 0 et 1. Attention à la position des colonnes lors de l'initialisation !

Le numéro de la ligne, introduite par l'utilisateur est stocké dans la variable ValB. Il est nécessaire de limiter ValB à 15. Il faut aussi réaliser la saisie et la déclaration de ValB du type int.

En utilisant directement la matrice, il faut afficher les 4 valeurs de la ligne sélectionnée de la manière suivante :

TestB : No ligne = n [3] = v [2] = v [1] = v [0] = v

Exemple de résultat à obtenir :

```

D:\H\etCoursSW\SL124_LOGA\Exercices\ProjExVS2010\SolEx13\Debug...
Ex13 Christian HUBER
Test A ou B, Q pour Quitter
b
TestB: entrez le numero de ligne (max 15)
1
TestB : No Ligne = 1 [3]=0 [2]=0 [1]=0 [0]=1
Test A ou B, Q pour Quitter
b
TestB: entrez le numero de ligne (max 15)
8
TestB : No Ligne = 8 [3]=1 [2]=0 [1]=0 [0]=0
Test A ou B, Q pour Quitter
b
TestB: entrez le numero de ligne (max 15)
14
TestB : No Ligne = 14 [3]=1 [2]=1 [1]=1 [0]=0
Test A ou B, Q pour Quitter
b
TestB: entrez le numero de ligne (max 15)
15
TestB : No Ligne = 15 [3]=1 [2]=1 [1]=1 [0]=1
Test A ou B, Q pour Quitter

```