

## PL1 - Travaux pratiques - Séance 4

### Sous-programmes, pointeurs et tableaux

Avant de commencer cette séance :

1. placez-vous dans votre répertoire **PL1** : `cd PL1`
2. récupérez les fichiers nécessaires à ce TP :  
`cp ~mounlaur/CCI_PL1/TP4.tar.gz .`
3. dé-compressez et dé-archiver ce fichier :  
`gunzip TP4.tar.gz ; tar -xvf TP4.tar`
4. placez-vous dans le répertoire **TP4** : `cd TP4`

#### Exercice 1 - Somme de 2 entiers [obligatoire]

Le fichier **somme.c** contient un programme C permettant de calculer la somme de 2 entiers lus au clavier.

**Q1.** Ouvrez ce programme (avec `code` ou `gedit`) et examinez-le. On constate que les variables à additionner (`a` et `b`) ainsi que leur somme `s` sont des **variables globales**. L'addition est effectuée par la fonction de nom `somme2entiers`.

Compilez ce programme en un exécutable de nom `somme` et exécutez-le.

**Q2.** On souhaite améliorer ce programme en supprimant l'utilisation de variables globales. Ainsi :

- `a`, `b` et `c` vont devenir des **variables locales** à la fonction principale `main` ;
- la fonction `somme2entiers` va prendre 2 **paramètres donnés** de type entier et renvoyer un **résultat** de type entier.

Effectuez ces modifications. Re-compilez votre programme et vérifiez qu'il fonctionne.

**Q3.** Même question mais en modifiant la fonction `somme2entiers` pour que le résultat soit fourni sous la forme d'un **paramètre résultat**.

#### Exercice 2 - Mystère [conseillé]

Le fichier **mystere.c** contient un programme C qui utilise des variables de type pointeur d'entiers.

**Q1.** Ouvrez ce programme (avec `code` ou `gedit`) et examinez-le. En exécutant "mentalement" ce programme essayez de trouver les valeurs des variables `a`, `b` et `c` aux points d'observation (1), (2) et (3) (en commentaires dans le programme).

**Q2.** Vérifiez votre réponse en modifiant le programme pour qu'il imprime les valeurs de ces variables à ces trois points d'observation. Compilez et exécutez ...

### Exercice 3 - recherche dans une séquence [obligatoire]

Le fichier **recherche.c** contient un programme C qui génère aléatoirement une séquence de 10 entiers dans un tableau et l'affiche.

**Q1.** Ouvrez ce programme (avec **code** ou **gedit**) et examinez-le. Compilez ce programme en un exécutable de nom **recherche** et exécutez-le.

**Q2** Ajoutez dans ce fichier une fonction **recherche** qui reçoit en paramètre donné un tableau **T** et sa longueur, un entier **x**, et qui renvoie un booléen indiquant si **x** appartient à **T** :

recherche : la fonction (un tableau **T** d'entiers, un entier **L**, un entier **x**)  $\rightarrow$  un booléen  
{recherche(**T**, **L**, **x**) vaut vrai ssi **x** est un élément de **T**}

**Q3.** Complétez la fonction **main** pour que, après avoir généré et affiché le tableau **Tab**, le programme

1. lise un entier au clavier ;
2. appelle la fonction **recherche**(**Tab**, 10, **x**) ;
3. affiche le résultat à l'écran

Compilez votre programme et faites quelques tests pour vérifier qu'il est correct.

**Remarque :** Pour programmer en C la fonction **recherche** il sera nécessaire de lui ajouter la taille du tableau en paramètre.

### Exercice 4 - modification d'une séquence [optionnel]

Le fichier **insertion.c** contient un programme C qui génère aléatoirement une séquence de 10 entiers positifs, la mémorise dans un type **Sequence** et l'affiche.

**Q1.** Ouvrez ce programme (avec **code** ou **gedit**) et examinez-le. Compilez ce programme en un exécutable de nom **insertion** et exécutez-le.

**Q2** Ajoutez dans ce fichier une fonction **ajout\_tete** qui reçoit en paramètre donné une **Sequence** **S**, un entier **x**, et qui insère **x** en tête de la séquence **S** :

ajout\_tete : la fonction (une **Sequence** **S**, un entier **x**)  $\rightarrow$  une **Sequence** **S**  
{ajout\_tete(**S**, **x**) renvoie la séquence **x** o **S**}

**Q3.** Complétez la fonction **main** pour que, après avoir généré et affiché la séquence **S**, le programme :

1. lise un entier au clavier ;
2. appelle la fonction **ajout\_tete**(**S**, **x**) ;
3. affiche le résultat à l'écran

Compilez votre programme et faites quelques tests pour vérifier qu'il est correct.