

TP d'algorithmique des données n° 2

Tableaux et Structures

Ce TP doit être fini avant la prochaine séance de TP, éventuellement sur votre temps de travail personnel.

Connectez-vous sous linux. Placez-vous dans le répertoire **TPalgo** à l'aide de la commande **cd**. Créez un répertoire **TP2** avec la commande **mkdir** et placez-vous dans ce répertoire.

Exercice 1 Fonctions

En C les fonctions se déclarent comme les méthodes statiques en JAVA, sans y mettre les mots *public static*. Par exemple la fonction suivante calcule la somme de deux entiers :

```
int somme(int i, int j){
    return i+j;
}
```

En C, toutes les variables sont passées par valeur, c'est-à-dire que leur valeur n'est modifiée que localement lors de l'exécution de la fonction, comme pour les types primitifs en JAVA. Le passage par référence (comme pour les objets en JAVA) nécessite l'utilisation de pointeurs.

Ouvrez le fichier **fonction.c** sans l'exécuter.

1. Devinez ce que fait ce programme.
2. Vérifiez votre intuition en testant ce code.

Exercice 2 Tableau

En C, la déclaration d'une variable de type tableau se fait simplement en indiquant entre crochet la taille du tableau. Par exemple :

```
#include<stdio.h>
#include<stdlib.h>

int main(){
    int tab[3]
}
```

permet de déclarer un tableau de taille 3, appelé **tab** dans la fonction **main**.

L'utilisation des tableaux se fait ensuite comme en JAVA. Par exemple, le fichier **tableau.c** retourne la somme des éléments d'un tableau de taille 4 saisis au clavier.

1. Ecrivez un programme qui calcule le maximum d'un tableau à 6 éléments entrés au clavier.
2. Ecrivez un programme contenant :
 - une constante entière **n** qui vaut 10 : après les insertions de bibliothèques, écrivez **const int n=10;** dans votre fichier.

- une fonction saisie qui prend en argument un tableau de n entiers et qui remplit ce tableau de valeurs saisis au clavier
 - une fonction affichage qui prend en argument un tableau de n entiers et qui l'affiche.
 - une fonction tri qui prend en argument un tableau de n entiers et le tri par la méthode du tri bulle
- qui demande à l'utilisateur de saisir les valeurs du tableau, affiche le tableau saisi par l'utilisateur (non trié) puis le tableau trié par ordre croissant par la méthode du tri bulle.

Exercice 3 Structures

Une structure permet en C de créer de nouveaux types à partir de ceux existants. Par exemple, si l'on veut gérer des points du plan pour faire un programme faisant des calculs de géométrie, on a besoin de deux *float* codant leurs coordonnées. Pour cela, la syntaxe est la suivante (à utiliser au début du programme) :

```
struct spoint {
    float abs;
    float ord;
};
```

On a ainsi défini un nouveau type *struct spoint* contenant deux champs de type *float*. On accède aux champs d'une structure à l'aide d'un `.`. La fonction suivante :

```
struct spoint milieu(struct spoint A, struct spoint B){
    struct spoint M;
    M.abs=(A.abs+B.abs)/2;
    M.ord=(A.ord+B.ord)/2;
    return M;
}
```

calcule le milieu de 2 points.

Pour le code et éviter d'écrire toujours le mot clé **struct**, on peut créer un *alias* grâce au code suivant :

```
typedef struct spoint point;
```

On dispose maintenant d'un type **point** qui est exactement la même chose que **struct spoint**. Il est à noter que cela peut se faire-mais ne doit pas être fait, cela rendrait le code illisible- pour tout type. Par exemple :

```
typedef int toto;
```

permet d'utiliser le mot toto à la place de int.

1. Proposez un enregistrement **rectangle** qui a pour champs **longueur** et **largeur**.
2. Proposez une fonction qui prend en argument un **rectangle** et retourne son périmètre, testez la.
3. Proposez une fonction qui prend en argument un **rectangle** et retourne son aire, testez la.
4. Proposez une fonction qui prend en argument deux **rectangles** et indique la **longueur** et la **largeur** du **rectangle** qui a la plus grande surface, testez la.

5. Proposez un programme utilisant les fonctions précédentes dont le menu est le suivant :
- tapez 1 pour initialiser les valeurs du premier rectangle,
 - tapez 2 pour initialiser les valeurs du deuxième rectangle,
 - tapez 3 pour connaître le périmètre du premier rectangle,
 - tapez 4 pour connaître la surface du premier rectangle,
 - tapez 5 pour connaître la longueur et la largeur de celui des 2 rectangles qui a la plus grande surface,
 - tapez 0 sinon.

Pour le menu de ce programme, on pourra utiliser l'instruction **switch**.