

TP 4 : POINTEURS ET PORTÉES DES VARIABLES

TABLE DES MATIÈRES

But	1
Exercice 1 : État de la mémoire	2
Exercice 2 : Écriture et utilisation d'une fonction ayant comme argument des pointeurs	2
Exercice 3 : tri à bulle d'un tableau d'entier	3

BUT

Vous devez maîtriser à la fin de cette séance les notions suivantes :

- la notion de portée des variables ;
- l'état de la mémoire au cours d'un programme ;
- les pointeurs.

EXERCICE 1 : ÉTAT DE LA MÉMOIRE

Qu'imprime le programme ci-dessous ? Justifiez vos réponses en donnant l'évolution de l'état de la mémoire

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

void f2(int *x, int *y);
void f3(int *x, int *y, int *z);

void f2(int *x, int *y) {
    int a;
    a = *x; *x = *y; *y = a;
}

void f3(int *x, int *y, int *z) {
    f2(x, y);
    f2(x, z);
}

int main(void) {
    int a, b, c;

    a = 1; b = 2; c = 3;

    f3(&a, &b, &c);

    printf("Main. a = %d, b = %d, c = %d\n", a, b, c);

    return 0;
}
```

EXERCICE 2 : ÉCRITURE ET UTILISATION D'UNE FONCTION AYANT COMME ARGUMENT DES POINTEURS

Le but de l'exercice est de calculer l'aire et le volumes de la sphère. Nous allons rappeler les formules de l'aire et du volume de la sphère :

$$\text{Aire de la sphère} = 4\pi R^2$$

$$\text{Volume de la sphère} = \frac{4}{3}\pi R^3$$

où R désigne le rayon de la sphère.

- (1) Écrire une fonction `int aire_vol_sphere(double rayon, double *Aire, double *Volume)`; Cette fonction retourne 0 si les arguments de la fonction sont corrects (rayon de la sphère positif), et 1 sinon.
- (2) Écrire une fonction `main` qui déclare les variables nécessaires et qui appelle la fonction `aire_vol_sphere`.

EXERCICE 3 : TRI À BULLE D'UN TABLEAU D'ENTIER

Le but de faire une fonction permettant de trier le contenu d'un tableau d'entier. Pour cela on utilisera le tri à bulle dont le principe est de propager, au cours d'une itération, le plus grand élément vers la fin du tableau par permutations successives. Ainsi, les plus grands éléments « remontent » comme des bulles. La figure illustre une itération d'un tel tri.

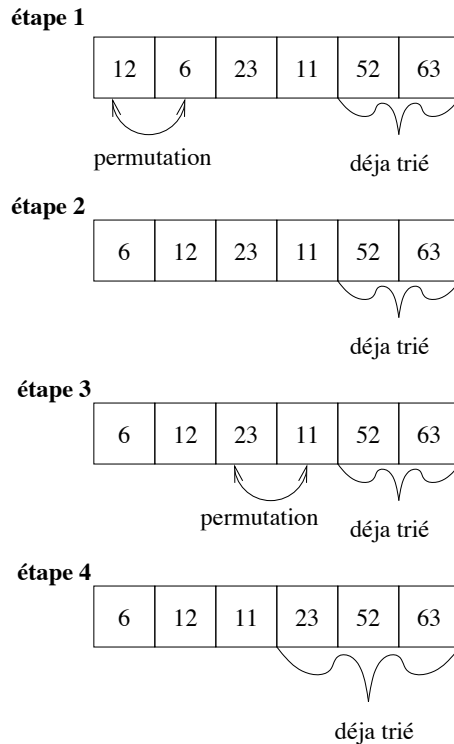


FIGURE 1. Exemple d'une itération de tri

- (1) Écrire une fonction `int init_tab(int *tab, int n, int elem_min, int elem_max)` qui initialise un tableau de manière aléatoire avec des entiers compris entre `elem_min` et `elem_max`. Cette fonction renvoie 0 si l'initialisation est possible et 1 sinon.
- (2) Déterminer le prototype et écrire le code de d'une fonction `affiche_tableau` qui affiche le contenu d'un tableau d'entier.
- (3) Déterminer le prototype de la fonction `tri_bulle` qui trie un tableau d'entier par ordre croissant et écrire son code.
- (4) Écrire un `main` qui initialise un tableau d'entier dont la taille est demandé à l'utilisateur, l'affiche, le trie puis affiche le tableau trié.